

Article

Not peer-reviewed version

Decoding Brain Signals in a Neuromorphic Framework for a Personalized Adaptive Control of Human Prosthetics

[Georgi Rusev](#) , [Svetlozar Yordanov](#) , Simona Nedelcheva , [Alexander Banderov](#) , [Fabien Sauter-Starace](#) , [Petia Koprinkova-Hristova](#) * , [Nikola Kasabov](#) *

Posted Date: 11 February 2025

doi: 10.20944/preprints202502.0797.v1

Keywords: brain-machine interfaces; motor control decoder; spiking neural networks; neuromorphic systems; ECoG, personalized neuro-prosthetics



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Article

Decoding Brain Signals in a Neuromorphic Framework for a Personalized Adaptive Control of Human Prosthetics

Georgi Rusev¹, Svetlozar Yordanov¹, Simona Nedelcheva¹, Alexander Banderov¹, Fabien Sauter-Starace², Petia Koprinkova-Hristova^{1,*} and Nikola Kasabov^{1,*}

¹ Institute of Information and Communication Technologies, Bulgarian Academy of Sciences, Sofia, Bulgaria

² Université Grenoble Alpes, CEA Leti DTIS, Grenoble F-38000, France

* Correspondence: petia.koprinkova@iict.bas.bg, nkasabov@aut.ac.nz

Abstract: Current technological solutions for Brain-machine Interfaces (BMI) achieve reasonable accuracy, but most systems are large in size, power consuming and not auto-adaptive. This work addresses the question whether current neuromorphic technologies could resolve these problems? The paper proposes a novel neuromorphic framework of a BMI system for prosthetics control via decoding Electro Cortico-Graphic (ECoG) brain signals. It includes a three-dimensional spike timing neural network (3D-SNN) for brain signals features extraction and an on-line trainable recurrent reservoir structure (Echo state network (ESN)) for Motor Control Decoding (MCD). A software system, written in Python using NEST Simulator SNN library is described. It is able to adapt continuously in real time in supervised or unsupervised mode. The proposed approach was tested on several experimental data sets acquired from a tetraplegic person. First simulation results are encouraging, showing also the need for a further improvement via multiple hyper-parameters tuning. Its future implementation on a neuromorphic hardware platform that is smaller in size and significantly less power consuming is discussed too.

Keywords: brain-machine interfaces; motor control decoder; spiking neural networks; neuromorphic systems; ECoG; personalized neuro-prosthetics

1. Introduction

Progress in neuroscience, biomedical engineering, and artificial intelligence led to significant advancements in Brain-Computer Interface (BCI) technology and its applications in various domains.

Contemporary signal acquisition and processing techniques significantly improved the accuracy and reliability of BCIs [1,2]. In particular, artificial intelligence (AI) offered novel neural signal decoding techniques, including adaptive algorithms and deep learning models with improved classification accuracy.

Recent medical applications of BCI offer revolutionary benefits to patients with neurological conditions such as stroke, spinal cord injuries, and neurodegenerative disorders [3,4] leading to severe communication disabilities, allowing direct interaction with devices and prostheses.

The research and practical applications of BMIs for controlling human prosthetics have been developing fast in the last decade [5–9]. A pioneering work has been developed in [10,11], where brain implants are inserted in the skull to measure epidural ElectroCorticoGrams (ECoG) signals of the motor and sensory cortices in order to extract movement intentions. These signals are decoded in a motor control decoder (MCD) using machine learning and statistical signal processing techniques. The output of the decoder is sent as commands to control exoskeleton or implantable spinal cord stimulator. For a better calibration, in [12] an adaptive tensor-based recursive exponentially weighted Markov-switching multi-linear model (REW-MSLM) decoder is proposed. REW-MSLM uses a mixture of expert (ME) architecture, mixing or switching independent decoders (experts) according to the probability estimated by a 'gating' model. The performance of the REW-MSLM decoder allows the patients

to use the neuroprosthesis for several weeks without recalibration. In this case, an unsupervised recalibration/adaptation would be a great added value for home use.

However, there are still many challenges and unsolved problems that prevent the commercialization of advanced BCI systems, such as accurately interpreting brain signals, the need for individual calibration, and ensuring long-term reliable use. The NEMO-BMI project (<https://nemo-bmi.net>) aims at the development of on the one hand auto-adaptive spinal stimulation and brain signal decoders and on the other hand of miniaturized and low power hardware for brain signal decoding. MCDs and their hardware realizations using both classical machine learning and statistical signal processing methods along with neuromorphic methods are part of the project.

With the development of neuromorphic devices, that are small in size, consume significantly less power and are adaptable to new data, such as SpiNNaker [13] and Loihi [14], to mention only two of them, the challenge now is to develop neuromorphic computational models for BMI and to implement them on these devices. Spike timing neural networks (SNN) are the most used and implemented in the neuromorphic devices neural information processing techniques, where brain-inspired principles of machine learning, such as spike information representation and spike timing dependent plasticity (STDP) are used along with many other methods [15].

Our paper aims at the introduction of a neuromorphic framework using SNNs, which allows adaptation of the neuronal activity decoder in a real time during the neuroprosthesis self-directed use. The neuromorphic algorithms may improve BMI performance and can be implemented in a novel chip with low-power consumption compared to conventional BMI algorithm hardware implementations.

This paper adds numerous discussions on the challenges related to the development of adaptive BMI [16–23], by proposing a framework of a neuromorphic MCD along with the discussions on the applicability of various design methods. The addressed issues and the proposed framework are illustrated on a subset of data from case study [10,12]. These include ECoG data of a single person for idle vs several movement states either in serious game or in real 3D space. The used in this paper ECoG data is collected using 64 electrodes as described in [10,12]. Although the hand and leg movements are triggered by brain activity in numerous spatial clusters of neurons over time, here we consider only ECoG signals from the motor cortex, which makes the task of predicting the desired by a patient's movement challenging.

The pioneering designed and experimentation of MCD in CEA [10,12] has demonstrated a good accuracy of detecting imaginary movements of legs and hands. In this paper, we are aiming at achieving similar results, but with the use of neuromorphic algorithms that would allow conceptually for a small in size and drastically lower power consuming BMI in the future.

2. Materials and Methods

2.1. Neuromorphic Framework for Adaptive MCD

One class of computational SNN architectures are the brain-inspired SNN, where a SNN is spatially structured using a three-dimensional (3D) brain template, and connectivity trained on spatio-temporal brain data, such as EEG or ECoG using STDP or other brain-inspired learning methods [15]. In the NeuCube SNN architecture [24], the spatio-temporal brain patterns learned in such a 3D SNN are classified or used for regression in an output SNN module, such as the dynamic evolving SNN (deSNN) [25].

A new computational framework is proposed here, where a 3D-SNN is structured according to an individual brain template. The patterns learned in the SNN from the ECoG time series data are neurons' spiking frequencies during a given period of time. They are classified by a recurrent fast trainable neural network architecture called Echo State Network (ESN). A preliminary version of this framework was first tested on a benchmark EEG data, producing better classification results than an original system [26]. Here, this framework is further developed for MCD on ECoG signals and tested on the case study data from [10,12]. The proposed framework is shown in Figure 1. It consists of two basic modules:

- A 3D recurrent SNN architecture (3D-SNN), spatially structured and adaptable to an individual 3D brain template, for feature extraction from recorded ECoG brain signals. The SNN is incrementally trained in an unsupervised mode.
- A recurrent Echo State Network (ESN) for decoding of the brain trained in a supervised mode on-line via recursive least squares (RLS) or reinforcement learning (RL) algorithms.

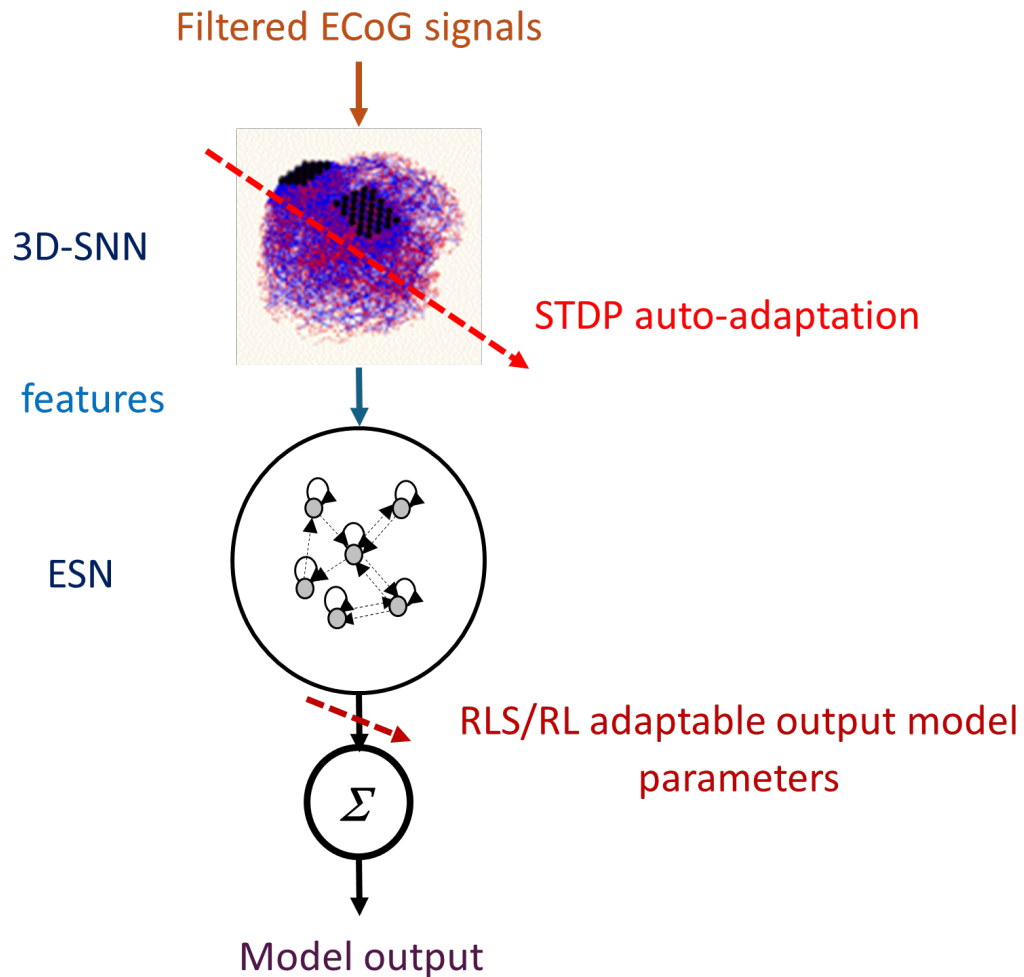


Figure 1. Functional diagram of the proposed MCD.

The 3D-SNN is structured according to a personalized brain template of a patient. It can consist of full 3D brain volume template or only of the 3D ECoG electrodes positions. SNN consists of leaky integrate and fire (LIF) neurons modeled by eq. 1:

$$dV_m/dt = -(V_m - E_L)/\tau_m + (I_{syn} + I_e)/C_m \quad (1)$$

where V_m is neural cell membrane potential; E_L - resting membrane potential; τ_m - membrane time constant; C_m - membrane capacity; I_e - constant input current; I_{syn} - total synaptic current coming from connected neurons. A spike is emitted when cell membrane potential V_m goes above the threshold value V_{th} and then the membrane potential returns to its resting state V_{rest} for refractory period t_{ref} .

Neurons were positioned in the 3D space according to the template. The overall connectivity within 3D-SNN is initially designed based on small world connectivity rule via STDP synapses. The neuronal activity of the 3D-SNN is triggered by filtered ECoG electrodes measurements called further temporal features. In current study these temporal features were extracted from raw ECoG data by Morlet transformation with various fundamental frequencies [17] or by shifting the row signal. They

are fed as generating currents of the neurons corresponding to ECoG electrode positions. The 3D-SNN connections adapt continuously to these signals via STDP rule [27] in unsupervised mode, which can be considered as a kind of auto-adaptition.

The features extracted from 3D-SNN are spiking frequencies of the selected neurons (in our case those corresponding to the ECoG electrode positions) for a given time interval (59 recording steps in our case study below) denoted as t_{ECoG} :

$$features(t - t_{ECoG} \div t) = n_{spikes} / t_{ECoG} \quad (2)$$

Next, these features are fed as input $in(t)$ to the ESN, that is a member of a novel and rapidly developing family of reservoir computing approaches [28], whose aim is the creation of a fast trainable recurrent neural network (RNN) architecture, able to approximate nonlinear time series. It incorporates a pool of neurons with sigmoid activation function (usually the hyperbolic tangent) that has randomly generated recurrent connection weights. The reservoir state $R(t)$ for the current time instant t depends both on its previous state $R(t - 1)$ and the current input $in(t)$ as follows:

$$R(t) = (1 - a)R(t - 1) + a \tanh(W^{in}in(t) + W^{res}R(t - 1)) \quad (3)$$

Here W^{in} is the matrix of input to reservoir connection weights that are randomly generated; W^{res} is the internal reservoir connection weight matrix that is sparse and also randomly generated, with spectral radius below 1; a is leaking rate parameter.

The ESN output $out(t)$ is calculated as a linear combination of the concatenation of the input and reservoir states $[R(t) in(t)]$ weighted by the output weight matrix W^{out} :

$$out(t) = W^{out}[(R(t) in(t))] \quad (4)$$

The only trainable parameters of ESN are the output weights W^{out} . Since the output is linear function, the least squares method is applied to train the ESN in a single iteration. For the aims of on-line training the recursive version of least squares (RLS) can be applied too.

The ESN of our MCD has different groups of outputs and corresponding to them parameters W^{out} that are presented below in the case of the different experimental data sets.

- State (idle, hand or leg movement) ($out(t) = state, W^{stateout}$): a vector with dimension corresponding to the number of possible states that was decoded via $argmax$ from the predicted model output.
- Trajectory (left and right hand positions in 3D space) ($out(t) = trajectory, W^{trajectory}$): a vector of x, y and z coordinates for each hand.
- Satisfaction ($out(t) = satisfaction, W^{satisfactionout}$): two-dimensional vector that was decoded via $argmax$ from the predicted model output.

A satisfaction corresponds to when an action performed by the BCI's effector matches the user's intention. A dissatisfaction is when it does not match. In [29,30], the authors used the "satisfaction" to derive motor labels in order to auto-adaptively update their motor control decoder. In presented here case, the satisfaction was used as reinforcement signal in order to correct the output of MCD in case of dissatisfaction.

The adaptation of the proposed MCD is achieved in two ways, as follows:

- Using STDP rule to continuously adapt the 3D-SNN connectivity in an unsupervised way, based on history of the input signal as well as the state of spiking neurons. This can be used for auto-adaptation of the model, when new data from the subject are continuously entered for unsupervised learning. This will ensure that the model will continuously adapt (auto-adapt) with the subject's movement improvement in time.
- Using reinforcement learning rule (RL) of the output parameters of the ESN responsible for motor action based on satisfaction prediction.

2.2. Software Implementation

The training algorithm is shown as Algorithm 1.

Algorithm 1 Pseudocode of on-line training algorithm

Initialization

Initialize ESN module parameters

Compose 3D-SNN module using neurons and ECoG positions

```

while new data do
2:   Read from files  $\leftarrow$  ECoG signals and desired state
      generating currents  $\leftarrow$  filtered ECoG signals
4:   for  $t_{\text{ECoG}}$  do
      Simulate 3D – SNN
6:   end for
      features  $\leftarrow$  3D – SNN
8:   ESN  $\leftarrow$  features
      predicted state  $\leftarrow$  ESN output
10:  train  $W^{\text{out}}$ 
end while
12: Keep model parameters

```

In training mode the only adjustable parameters of the model are the output connection weights W^{out} of the ESN module. They are tuned incrementally with every new input/output training data pair via recursive least squares (RLS) method. The 3D-SNN connection weights change according to STDP rule continuously. In case of dissatisfaction predicted (or expressed by the patient if possible) reinforcement learning rule [31] changes the output parameters of MCD (W^{out}) using “satisfaction” label as reinforcement signal as follows:

$$\begin{aligned}
 W^{\text{out}} &= W^{\text{out}} - \alpha(1 - \text{satisfaction})E \\
 E &= \delta E + (1 - \delta)[R_{\text{in}}]_{\text{out}}
 \end{aligned}
 \tag{5}$$

where α and δ are learning rate and eligibility trace E decay rate respectively; R , in and out are current state, and output of the ESN module.

A block scheme of the developed neuromorphic algorithm for the MCD is shown in Figure 2. Both modules are written in Python. The 3D-SNN is based on *NEST simulator* library, version 3.3 [32] while the rest of code exploits *numpy*, *SciPy* and other Python libraries for mathematical calculations.

The software works in pseudo-online mode receiving (in this version, simply reading from *edf* file) as input the next portion of 59 records from ECoG electrodes and generating as output motor actions as well as satisfaction label for current time step. Results are kept in *csv* files but could be easily exported in a pseudo-online mode one by one.

Preliminary studies have been conducted on the feasibility of implementing the proposed MCD framework and algorithms on a neuromorphic hardware platform.

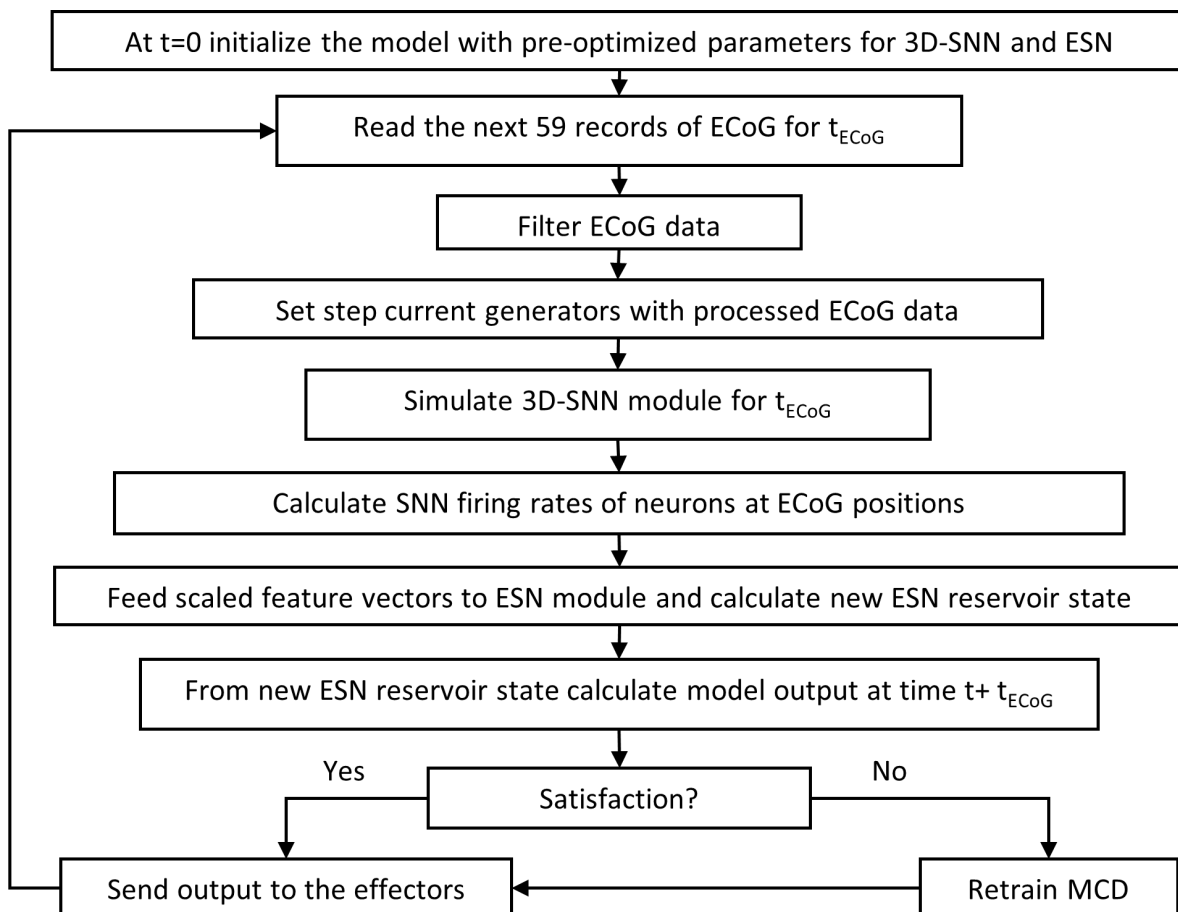


Figure 2. The overall algorithm of the MCD framework.

The estimation of model running time was done on a desktop configuration with following parameters:

- 12th Gen Intel® Core™ i7-12700
- Installed RAM 32 GB
- Base speed 2.10 GHz
- Cores 12
- Logical processors 20

The software runs under WSL in Windows 11. Estimated times for model initialization and during the training and testing phases are as follows:

- Initialization includes generation/reading of model parameters and setting-up of two recurrent structures (3D-SNN and ESN): 49 seconds approximately.
- Single RLS iteration (training on one input/output train data example): 2.5-3.5 seconds
- Single RL iteration (training on one input/output train data example): 0.55-0.65 seconds
- Single step model output calculation: 0.58-0.63 seconds

Our preliminary tests on neuromorphic hardware foreseen in this project – Intel’s Loihi – showed the following speed of 3D-SNN structure only: simulation for 10000 ms of 1000 sparsely connected LIF neurons execution time was 55.89 milliseconds (using the built-in Loihi2 profiler function). In comparison, our tests on a single core virtual machine on desktop configuration with Intel Core™ i3-4130 CPU, 3.4GHz, 8GB RAM needed approximately 1 sec to simulate for 1000 ms a 3D-SNN with 1471 neurons sparsely connected via STDP synapses using NEST simulator library. Thus, Loihi2 would offer approximately 200 (1000/5) times faster execution time and of course much lower power consumption.

At this point, we have a 3D-SNN implementation in LAVA (software recommended from Intel to prepare a code of implementation in Loihi). ESN module can be implemented also on the same neuromorphic hardware. As currently our model has about 1600 neurons totally (3D-SNN + ESN). We expect execution time on neuromorphic hardware for a test step (that uses ECoG recordings from approximately 100 milliseconds) to be orders of magnitude faster than on a desktop computer, e.g. approximately 0.6/200 seconds.

2.3. Experimental Data

The described here MCD experimental model, based on the proposed MCD framework, was trained on three data bases (DBs) collected from two different male patients with traumatic sensorimotor tetraplegia caused by a complete C4–C5 spinal cord injury as described in [10,12].

First patient underwent bilateral, while the second one – unilateral (on left hemisphere) implantation of chronic wireless ECoG WIMAGINE implants [33] composed of 64 planar electrodes with a 2.3 mm active diameter and 4–4.5 mm inter-electrodes distance (electrodes are in contact with the dura mater). The ECoG signals are low and high pass filtered in a bandwidth from 0.5 Hz to 300 Hz using analog bandpass filter and, after digitalization, using a digital low pass FIR filter with a cutoff frequency of 292.8 Hz, for digitalization the sampling frequency is 585.6 Hz, the resolution chosen at 10 bits in the Analog Digital Converter and eventually radio-transmitted in the MICS band (402-405 MHz) to a custom base station connected to a computer [33]. During the experimental sessions, 32 electrodes for each implant were selected in a checkerboard-like pattern because of limited data rates of the Microsemi Zarlink radio link (250kb/s in 2-FSK mode).

First DB, called the "Runner", was described in [29,30]. In this experiment, the first test subject was seated in front of a computer screen where a human avatar was represented from a third person perspective. The avatar could either stand still or walk forward at a fixed speed. Figure 3 shows the experimental set-up. The subject controlled the avatar using leg motor imagery. 13 sessions, of on average 11 minutes of recording, were acquired over 141 days. For blind-test of the algorithms, the database has been split in two parts. The first 9 sessions are provided with ECoG data and labels, while the 4 last contain ECoG data and the satisfaction labels only.



Figure 3. Experimental set-up for Runner DB acquisition.

The second DB, called 6DoF, is described in Alexandre Moly et al 2022 [12]. BCI experimental data was acquired during experimental sessions where the first patient performed 6D control of a virtual avatar with 3 different states: move right arm (3D movement), move left arm (3D movement) and idle. Figure 4 shows the experimental set-up.



Figure 4. Experimental set-up for 6DoF DB acquisition.

The hand trajectories were described in 3D space by consecutive positions (3D coordinates). 16 sessions were recorded. For blind-test of the algorithms, the database has been separated in two parts. The first 10 sessions are provided with labels and ECoG data, while the 5 last sessions are used for a blind test.

The third DB called 5DoF was collected during the experiment with the second subject. He was seated in front of a computer screen showing a virtual environment with avatar having four possible movement states in addition to its idle state: grasping by the right hand, grasping by the left hand, flexion of the right elbow and flexion of the left elbow. The subject performed direct motor imagery to activate each of the motor states. Two sessions of 32 and 53 minutes were performed. For blind-test of the algorithms, some labels were removed from the database. Both sessions are provided with ECoG data, labels (when not removed), satisfaction labels, and the three flags of updates (for reinforcement and supervised training of motor decoder as well for updating of satisfaction decoder).

3. Results

This section shows blind tests accuracy assessment of experimental MCD models trained on the three DBs described above. Here we also compare the two approaches for extracting ECoG temporal features, namely:

- *Approach 1:* Shifting up amplitudes of the ECoG signals.
- *Approach 2:* Square Morlet transformation [34] of ECoG signals using 15 fundamental frequencies (from 10 to 150 Hz with a step of 10 Hz).

The model hyper-parameter values are given in Table 1. Some of them were chosen by trial and error, others will be subject of further tuning.

Table 1. Exemplar model hyper-parameters.

Hyperparameter	Value
3D-SNN	
membrane potential threshold V_{th}	-65.0 mV
refractory time t_{ref}	0 ms
STDP learning rate λ	0.001
ESN	
leaking rate a	0.5
number of reservoir neurons	15000
sparsity of reservoir connectivity matrix W^{res}	0.5

Figures 5, 6 and 7 present accuracy of trained MCD models on Runner DB, 6DoF DB and 5DoF DB respectively.

The evaluation of features extraction approaches on Figure 5 demonstrates that Morlet wavelet transform (*Approach 2*) improved MCD prediction accuracy up to about 65% for some fundamental frequencies (e.g. 30Hz and 40Hz) in comparison with *Approach 1* for which results are slightly above chance level of 50%. "Voting" models decide using models trained with features extracted by several or all Morlet fundamental frequencies. We observe that the best results were obtained by voting among the models using selected best fundamental frequencies.

F_{score} on idle class is much higher than on class walk, which can be explained by imbalanced number of examples from idle vs walk classes in this DB. The combination of features extracted via multiple fundamental frequencies could probably result in better classification accuracy at the expense of increased model dimension. Further investigation of 15 models and their features would allow us to optimize the model.

As it can be seen on Figure 5, the test accuracy of a MCD model based on Morlet transformation of ECoG signals depends on the fundamental frequency used. As demonstrated there, a comparative analysis is needed for the selection of the best input features from raw ECoG signals for a particular subject, before a 3D SNN model is trained on these features for this subject.

Hence a preliminary analysis of ECoG data collected from a new subject and optimal feature selection is a crucial step towards the design of an efficient neuromorphic MCD.

Results shown on Figure 6 were obtained using Morlet transform (*Approach 2* for features extraction) at 10Hz only. The gate state (whether the patient intend to move left or right hand) prediction accuracy improved significantly in comparison with results obtained using raw ECoG data (*Approach 1* for features extraction). Further tests with other fundamental frequencies of Morlet wavelet would result in better accuracy as in previous DB tests. The idle state F_{score} may be improved after balancing the data.

Cosine similarity is used to measure both hands' trajectory correctness. It shows the need for improvement that can be achieved through further optimization of hyperparameters as discussed below.

Figure 7 shows testing accuracy of an exemplar MCD on 5DoF DB using *Approach 2* for features extraction. These results demonstrated accuracy that is above chance level, however it also shows the need for further improvement. Since in this DB ECoG signals were obtained only from a single-side (e.g. left) implant, we expect that a higher accuracy could be achieved with the use of two implants, located on the left and the right parts of the motor cortex Besides, these results were obtained using Morlet transform with 30 Hz fundamental frequency, so further investigation of other frequencies as well as a voting model might increase the testing accuracy. Another direction for improvement of

results could be changing the training approaches, since we apply both RLS and RL algorithms on the ESN in dependence on flags in DB.

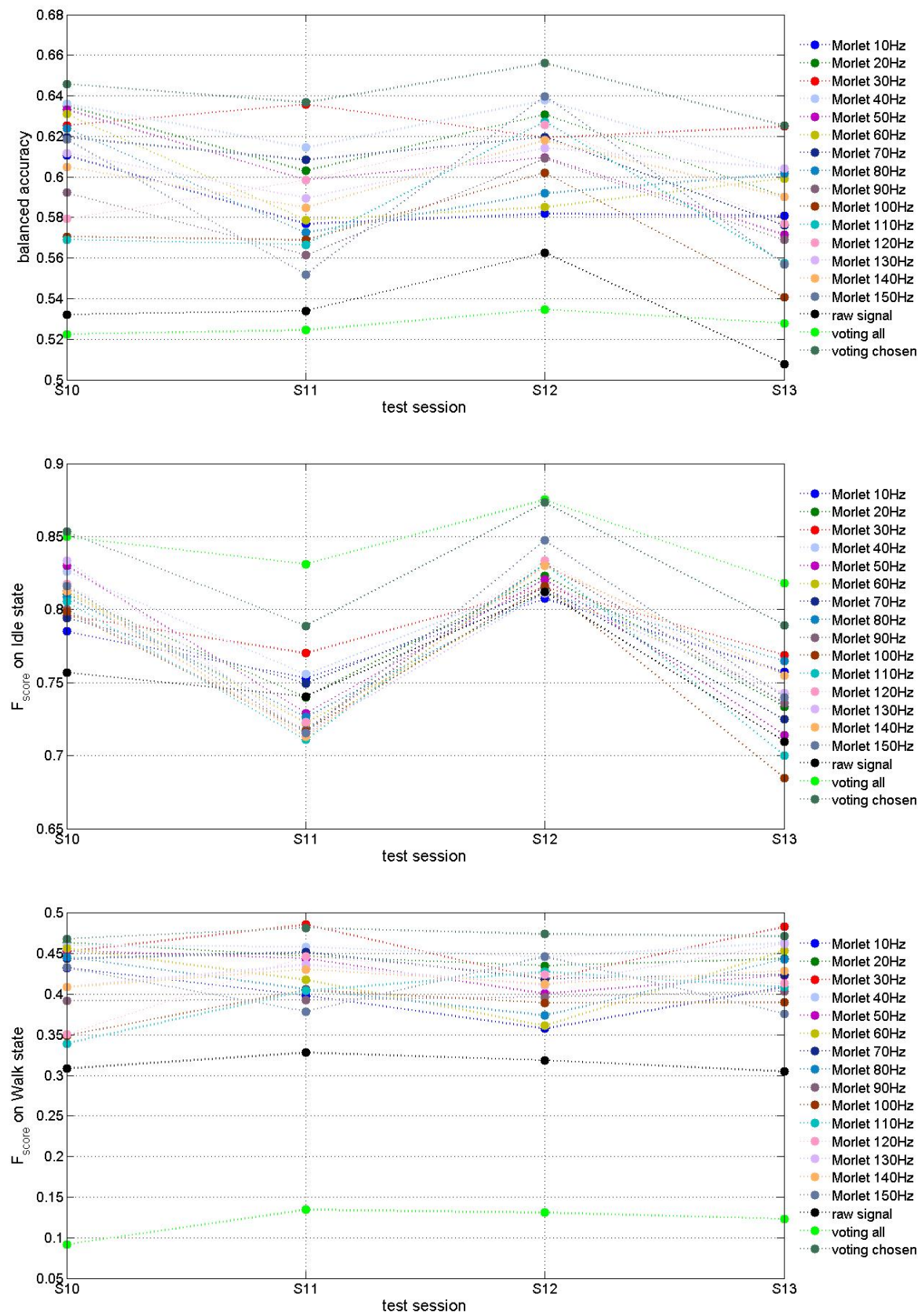


Figure 5. Testing accuracy of exemplar MCD on Runner DB.

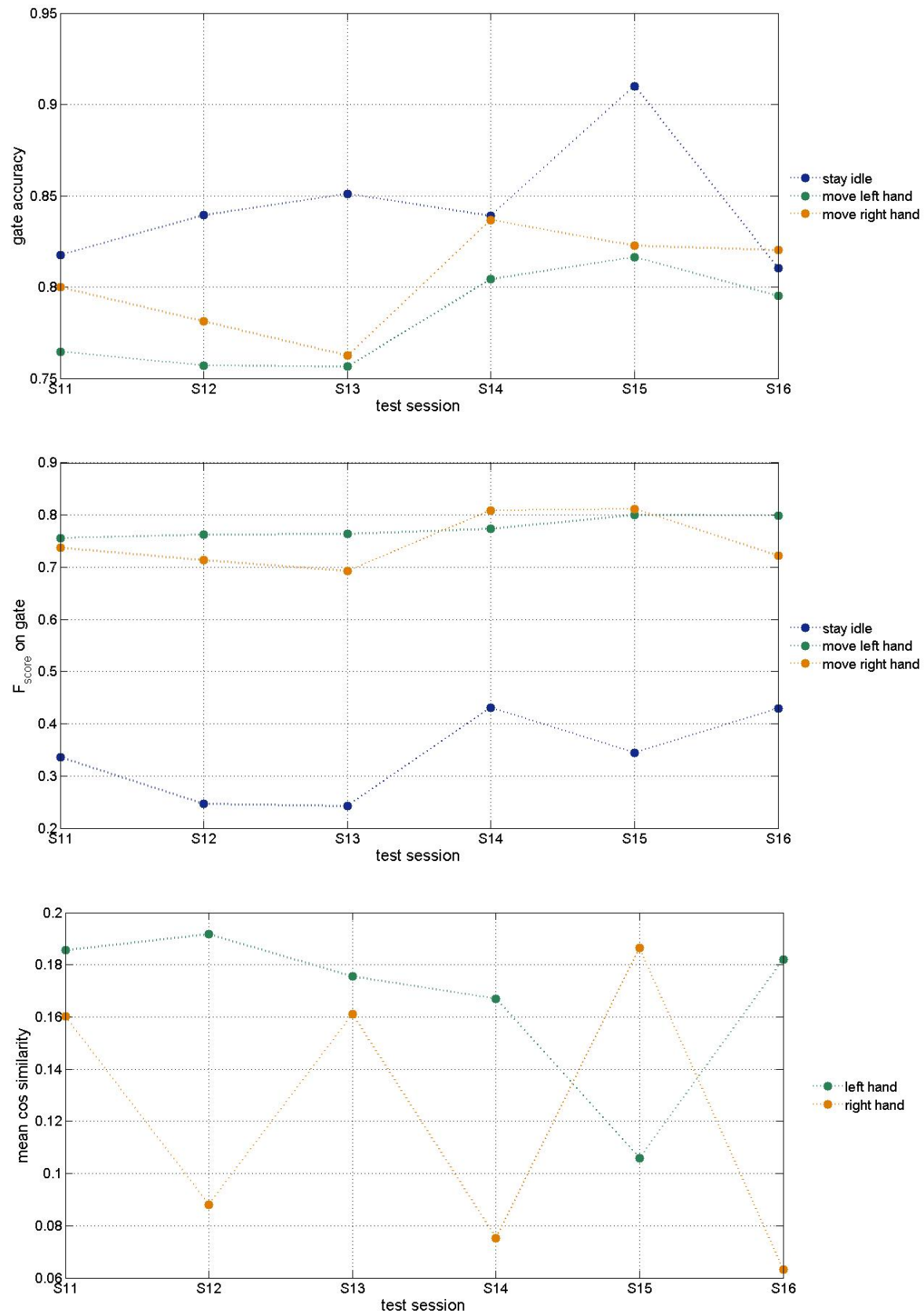


Figure 6. Testing accuracy of exemplar MCD on 6DoF DB.

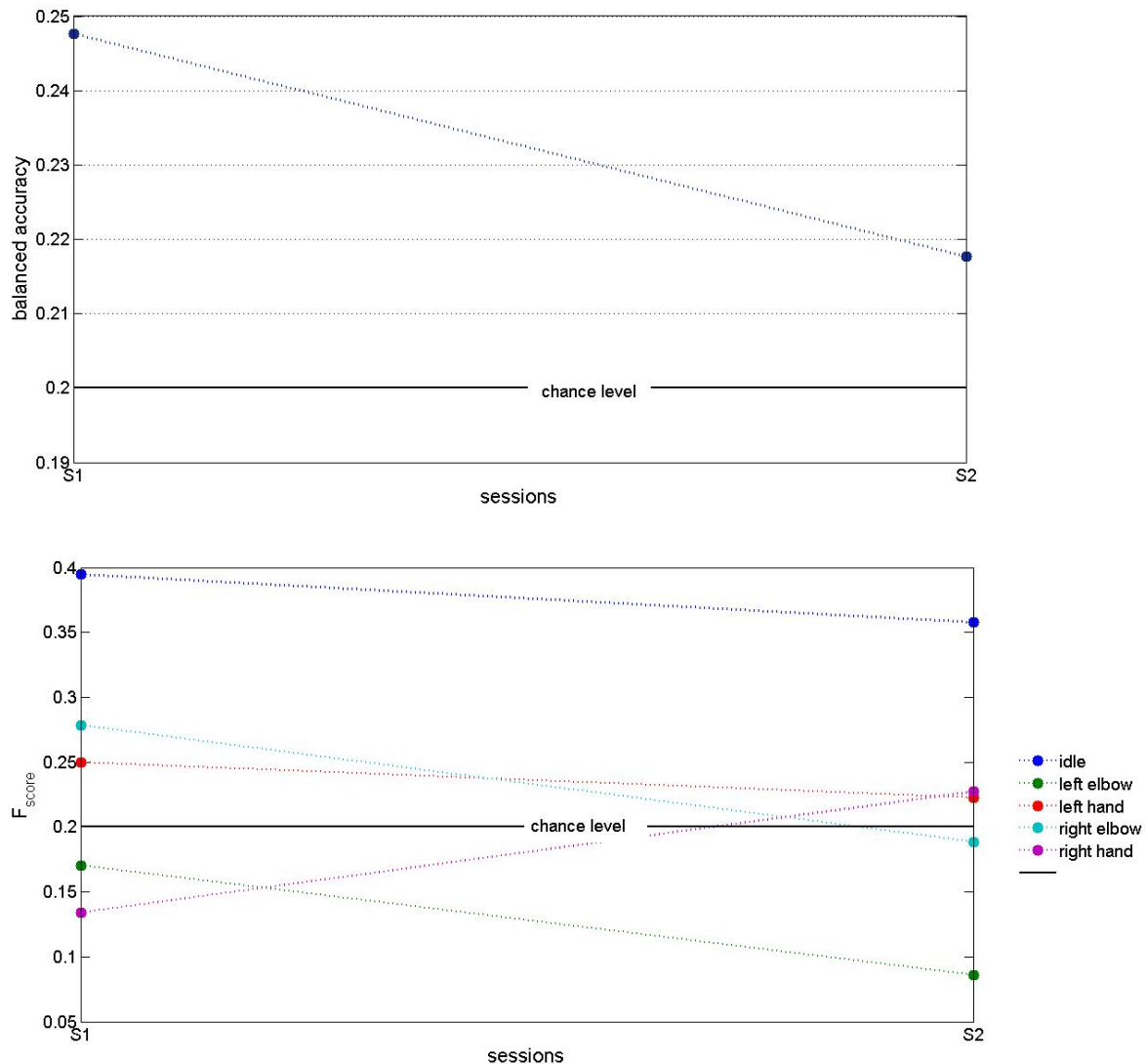


Figure 7. Testing accuracy of an exemplar MCD on 5DoF DB.

4. Discussion

Comparison with MCD accuracy on the same data sets reported in [10,12,29,30] shows the need of improvement of our model. Nevertheless, our approach has potential to increase MCD prediction accuracy subject to refinement of its multiple hyper-parameters.

The assessment of the neuromorphic model accuracy presented here was done on a given set of hyper-parameter values. Some of them were determined based on our preliminary grid search on much smaller data set [35]. The benchmark data set (EEG records for three different movements, 1 second per record, 60 examples, 20 per class) we had to test totally 243 000 combinations that took about 9 days on a supercomputer platform [26].

With much bigger data from NEMO-BMI project such tuning would need enormous time so the hyper-parameters in the tests on three DBs reported here were chosen based on preliminary results from [26]. Further fine tuning of hyper-parameters using different methods (e.g. evolutionary algorithms [15,36]) would result in higher model accuracy.

5. Conclusions

The paper presents a novel approach for the design of a personalized neuromorphic motor control decoder (MCD). The proposed MCD framework consists of modules that can be performed with the use of different methods and their parameters depending on the task and the available personal data.

The practical use of the framework requires analysis and understanding of the personalised data, selection of features, training a 3D SNN and an output classifier/regressor, along with optimization of the parameters of a model for a better performance for the individual in question [35]. Using different methods for classification and prediction [36,37] and rigorous parameter optimization could have an impact on the overall accuracy of the created personalized MCD.

The paper presents a software written in NEST and Python and discusses its possible implementation on a neuromorphic platform. As the proposed MCD framework is open for adding new methods to accomplish different functionality, further research is needed to establish the neuromorphic approach for efficient design of future MCD, that are minimal in size and highly power efficient [13,14,38].

Author Contributions: Conceptualization and methodology: N.K. and P.K.-H.; software and simulation experiments: G.R., S.Y., A.B. and S.N.; data curation: F.S.-S.; writing—original draft preparation: N.K., P.K.-H., F.S.-S.; visualization: S.N.; project administration: F.S.-S. All authors have read and agreed to the published version of the manuscript.

Funding: This research is funded by the European Commission under the HORIZON-EIC action “Auto-adaptive Neuromorphic Brain Machine Interface: toward fully embedded neuroprosthetics (NEMO-BMI)”, No 101070891/01.10.2022NEMO-BMI (<https://nemo-bmi.net>).

Informed Consent Statement: Not applicable. The experimental data for models’ training and testing was provided by NEMO-BMI project coordinator CEA.

Data Availability Statement: No new data were created or analyzed in this study. This study presents only the MCD model structure and accuracy assesment by simulations.

Acknowledgments: CEA Clinatec and CHUGA provided anonymized and processed data from the ongoing clinical trial BCI & tetraplegia registered on ClinicalTrials.gov as NCT02550522. CEA Clinatec team also computed the prediction performances. IICT-BAS team designed MCD structure and software and performed training and testing experiments on DBs provided by CEA. IICT team acknowledge the provided access to the infrastructure purchased under the National Roadmap for RI, financially coordinated by the MES of the Republic of Bulgaria (grant No D01-325/01.12.2023).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Rudroff, T., Decoding thoughts, encoding ethics: A narrative review of the BCI-AI revolution, *Brain Research*, vol. 1850, 2025, 149423
2. Sumithra, M.G., Dhanaraj, R.K., Milanova, M., Balusamy, B., Venkatesan, C. (Eds.), *Brain-Computer Interface: Using Deep Learning Applications*, 2023, ISBN:9781119857655, Wiley
3. Awuah, W.A., Ahluwalia, A., Darko, K., et al., *Bridging Minds and Machines: The Recent Advances of Brain-Computer Interfaces in Neurological and Neurosurgical Applications*, *World Neurosurgery*, vol. 189, 2024, pp. 138-153
4. Paul, D., Mukherjee, M., Bakshi, A., A Review of Brain-Computer Interface, In: Mukherjee, M., Mandal, J., Bhattacharyya, S., Huck, C., Biswas, S. (eds) *Advances in Medical Physics and Healthcare Engineering. Lecture Notes in Bioengineering*, 2021, pp. 507–531
5. Ajiboye et al., A. B., Restoration of reaching and grasping movements through brain-controlled muscle stimulation in a person with tetraplegia: a proof-of-concept demonstration, *The Lancet*, vol. 389, no. 10081, Art. no. 10081, May 2017, doi: 10.1016/S0140-6736(17)30601-3.
6. Buttfeld, A., Ferrez, P. W., Millan, Jd. R., Towards a robust BCI: error potentials and online learning, *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 14, no. 2, pp. 164–168, Jun. 2006, doi: 10.1109/TNSRE.2006.875555.
7. Elisayev, A. et al., Recursive Exponentially Weighted N-way Partial Least Squares Regression with Recursive-Validation of Hyper-Parameters in Brain-Computer Interface Applications, *Sci. Rep.*, vol. 7, no. 1, Art. no. 1, Dec. 2017, doi: 10.1038/s41598-017-16579-9.
8. Orsborn, A. L., Moorman, H. G., Overduin, S. A., Shانهchi, M. M., Dimitrov, D. F., Carmena, J. M., Closed-Loop Decoder Adaptation Shapes Neural Plasticity for Skillful Neuroprosthetic Control, *Neuron*, vol. 82, no. 6, pp. 1380–1393, Jun. 2014, doi: 10.1016/j.neuron.2014.04.048.

9. Wodlinger, B., Downey, J. E., Tyler-Kabara, E. C., Schwartz, A. B., Boninger, M. L., Collinger, J. L., Ten-dimensional anthropomorphic arm control in a human brain-machine interface: difficulties, solutions, and limitations, *J. Neural Eng.*, vol. 12, no. 1, Art. no. 1, Dec. 2014, doi: 10.1088/1741-2560/12/1/016011.
10. Benabid, A. L. et al., An exoskeleton controlled by an epidural wireless brain-machine interface in a tetraplegic patient: a proof-of-concept demonstration, *Lancet Neurol.*, Oct. 2019, doi: 10.1016/S1474-4422(19)30321-7.
11. Lorach, H., Galvez, A., Spagnolo, V. et al., Walking naturally after spinal cord injury using a brain-spine interface. *Nature* 618, 126–133 (2023). <https://doi.org/10.1038/s41586-023-06094-5>
12. Moly, A., Costecalde, T., Martel, F., Martin, M., Larzabal, C., Karakas, S., Verney, A., Charvet, G., Chabardes, S., Benabid, A. L., Aksenova, T., An adaptive closed-loop ECoG decoder for long-term and stable bimanual control of an exoskeleton by a tetraplegic, *J. Neural Eng.* 19 (2022) 026021, <https://doi.org/10.1088/1741-2552/ac59a0>.
13. Behrenbeck, J. et al., Classification and regression of spatio-temporal signals using NeuCube and its realization on SpiNNaker neuromorphic hardware, *J. Neural Eng.*, vol. 16, no. 2, p. 026014, Feb. 2019, doi: 10.1088/1741-2552/aafabc
14. Davies, M., Srinivasa, N., Lin, T.-H., Chinya, G., Joshi, P., Lines, A., Wild, A., Wang, H., Mathaikutty, D., (2018). Loihi: A Neuromorphic Manycore Processor with On-Chip Learning. *IEEE Micro*. PP. 1-1. 10.1109/MM.2018.112130359
15. Kasabov, N., Time-Space, Spiking Neural Networks and Brain-Inspired Artificial Intelligence, Springer (2018) 750p., <https://www.springer.com/gp/book/9783662577134>
16. Blumberg, J., Rickert, J., Waldert, S., Schulze-Bonhage, A., Aertsen, A., Mehring, C., Adaptive Classification for Brain Computer Interfaces, In 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Aug. 2007, pp. 2536–2539. doi: 10.1109/IEMBS.2007.4352845.
17. Gürel, T., Mehring, C., Unsupervised adaptation of brain machine interface decoders, *Front. Neurosci.*, vol. 6, p. 164, 2012, doi: 10.3389/fnins.2012.00164.
18. Kumarasinghe, K., Kasabov, N., Taylor, D., Brain-inspired spiking neural networks for decoding and understanding muscle activity and kinematics from electroencephalography signals during hand movements, *Sci. Rep.*, vol. 11, no. 1, p. 2486, Jan. 2021, doi: 10.1038/s41598-021-81805-4.
19. Kumarasinghe, K., Kasabov, N., Taylor, D., Deep learning and deep knowledge representation in Spiking Neural Networks for Brain-Computer Interfaces, *Neural Netw.*, vol. 121, pp. 169–185, Jan. 2020, doi: 10.1016/j.neunet.2019.08.029.
20. Llera, A., Gómez, V., Kappen, H. J., Adaptive Classification on Brain-Computer Interfaces Using Reinforcement Signals, *Neural Comput.*, vol. 24, no. 11, pp. 2900–2923, Nov. 2012, doi: 10.1162/NECO_a_00348.
21. Rouanne, V., Śliwowski, M., Costecalde, T., Benabid, A., Aksenova, T., Detection of Error Correlates in the Motor Cortex in a Long Term Clinical Trial of ECoG based Brain Computer Interface, In Proceedings of the 14th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2021)- Volume 4: BIOSIGNALS, pp. 26-34 ISBN: 978-989-758-490-9. doi: 10.5220/0010227800260034.
22. Spüler, M. et al., Decoding of motor intentions from epidural ECoG recordings in severely paralyzed chronic stroke patients, *J. Neural Eng.*, vol. 11, no. 6, Art. no. 6, Dec. 2014, doi: 10.1088/1741-2560/11/6/066008.
23. Zeyl, T. J., Chau, T., A case study of linear classifiers adapted using imperfect labels derived from human event-related potentials, *Pattern Recognit. Lett.*, vol. 37, pp. 54–62, Feb. 2014, doi: 10.1016/j.patrec.2013.05.020.
24. Kasabov, N. K., NeuCube: A spiking neural network architecture for mapping, learning and understanding of spatio-temporal brain data, *Neural Networks*, vol. 52, pp. 62-76, 2014.
25. Kasabov, N., Dhoble, K., Nuntalid, N., Indiveri, G., Dynamic evolving spiking neural networks for on-line spatio- and spectro-temporal pattern recognition, *Neural Netw.*, vol. 41, pp. 188–201, May 2013, doi: 10.1016/j.neunet.2012.11.014.
26. Koprinkova-Hristova, P., Penkov, D., Nedelcheva, S., Yordanov, S., Kasabov, N.. On-line Learning, Classification and Interpretation of Brain Signals using 3D SNN and ESN. Proceedings of the International Joint Conference on Neural Networks, 2023-June, IEEE, 2023, ISBN:978-166548867-9, DOI:10.1109/IJCNN54540.2023.10191974, Number-23518355.
27. Guetig et al., Learning input correlations through nonlinear temporally asymmetric hebbian plasticity. *Journal of Neuroscience*, vol. 23, 2003, pp. 3697-3714
28. Jaeger, H., Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the “echo state network” approach. GMD Report 159, German National Research Center for Information Technology (2002)

29. Rouanne, V. 2022, Adaptation of discrete and continuous intracranial Brain-Computer Interfaces using neural correlates of task performance decoded continuously from the sensorimotor cortex of a tetraplegic, PhD thesis, Université Grenoble Alpes (<https://theses.hal.science/tel-03917556>)
30. Rouanne V. et al 2022, Unsupervised adaptation of an ECoG based brain-computer interface using neural correlates of task performance, *Scientific Reports*, 12(1), 21316 (<https://doi.org/10.1038/s41598-022-25049-w>)
31. Sutton, Richard S. and Andrew G. Barto, *Reinforcement Learning: An Introduction*, second edition, 2018, The MIT Press, Cambridge, Massachusetts, London, England
32. Spreizer, S. et al., NEST 3.3 (3.3), 2022, Zenodo. <https://doi.org/10.5281/zenodo.6368024>
33. Mestais, C. S. et al 2015, WIMAGINE: wireless 64-channel ECoG recording implant for long term clinical applications, *IEEE Trans. Neural Syst. Rehabil. Eng.*, 23, p. 10-21 (<https://doi.org/10.1109/TNSRE.2014.2333541>)
34. Morlet, J. (1983). *Sampling Theory and Wave Propagation*. In: Chen, C.H. (eds) *Issues in Acoustic Signal — Image Processing and Recognition*. NATO ASI Series, vol 1. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-82002-1_12
35. Penkov, D., Koprinkova-Hristova, P., Kasabov, N., Nedelcheva, S., Ivanovska, S., Yordanov, S. (2024). Grid Search Optimization of Novel SNN-ESN Classifier on a Supercomputer Platform. In: Lirkov, I., Margenov, S. (eds) *Large-Scale Scientific Computations. LSSC 2023. Lecture Notes in Computer Science*, vol 13952. Springer, Cham. https://doi.org/10.1007/978-3-031-56208-2_45
36. Schliebs, S., Defoin-Platel, M., Worner, S., Kasabov, N. (2009). Integrated feature and parameter optimization for an evolving spiking neural network: Exploring heterogeneous probabilistic models. *Neural Networks*, 22(5-6), 623-632.
37. Kasabov, N., *Evolving Connectionist Systems*, Springer, 2007.
38. Furber, S. B., Galluppi, F., Temple, S., Plana, L. A., The SpiNNaker Project, In *Proceedings of the IEEE*, vol. 102, no. 5, pp. 652-665, May 2014, doi: 10.1109/JPROC.2014.2304638

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.