The Journal of Open Source Software

BCImat: a Matlab-based framework for Intracortical Brain-Computer Interfaces and their simulation with an artificial spiking neural network

Enrico Ferrea[*1], **Pierre Morel**[1,4], **and Alexander Gail**[1,2,3]

1 German Primate Center, Sensorimotor Group, Goettingen, Germany **2** University of Goettingen, Georg-Elias-Mueller Institute of Psychology, Goettingen, Germany **3** Bernstein Center for Computational Neuroscience, Goettingen, Germany **4** Univ. Littoral Côte d'Opale, Univ. Lille, Univ. Artois, ULR 7369 - URePSSS - Unité de Recherche Pluridisciplinaire Sport Santé Société, F-59140 Dunkerque, France

DOI: 10.21105/joss.03956

Software

- Review
- Repository
- Archive

Editor: Olivia Guest

Reviewers:

- @mstimberg
- @puolival

Submitted: 20 August 2021
Published: 29 November 2021

Summary

Recent advances in intracortical Brain-Computer Interface (BCI) technology allowed motor disabled patients to partially regain lost motor functions (Aflalo et al., 2015; Ajiboye et al., 2017; Collinger et al., 2013; Hochberg et al., 2012). In these patients, intact neural activity is extracted from motor-related areas of the cerebral cortex via intracortical implanted electrodes and interpreted by a machine-learning algorithm to control a prosthetic device, thereby bypassing dysfunctional corticospinal projections that resulted, for example, from spinal cord lesions. BCIs are also used for basic neuroscientific studies to establish a specific transformation between the brain area under investigation and a specific behavior(Koralek et al., 2012; Sadtler et al., 2014) and therefore imposing a direct causal link between them. The software introduced here allows true online BCI control of a computer cursor based on physiological signals (e.g. from patients) as well as realistic real-time simulations for testing all algorithms based on artificial spiking neural network (SNN) data using the identical control architecture.

Statement of need

Most of the publicly available software for BCIs is designed for applications based on time-continuous electrophysiological signals, like electroencephalographic (EEG) or electrocorticographic (ECoG) signals (Stegman et al., 2020) and requires existing recording possibilities or pre-recorded data for testing. BCImat instead is a MATLAB framework for implementing and testing a BCI which is based on stochastic event time-series, particularly neuronal spiking signals recorded from large numbers of individual neurons that vary at a time-scale of milliseconds. For example, BCImat can use intracortical single-neuron signals acquired with a Cereplex system (Blackrock, Salt Lake City, USA). Importantly, BCImat can alternatively use the input from a built-in artificial spiking neural network (SNN) for simulating the later online BCI control. This allows testing BCI applications with the same algorithm and framework as intended for later use in patients, but before the availability of implanted subjects and specific recording hardware. This way, the full online decoding experiment or application can be developed in advance without the need for pre-recorded data files. The code is intended for use by anyone wanting to test closed-loop BCI methods or perform intracortical closed-loop BCI experiments.

*corresponding author

The Journal of Open Source Software

Overview

The BCI framework (Fig.1) interfaces bidirectionally with a simple behavioral task controller (here written in c++) allowing to perform center-out reach movements for decoder calibration and then to control cursor movements via the neural activity. The communication between the task-controller and the BCI framework is done via the Virtual-Reality Peripheral Network (VRPN) protocol, implementing a client-server application via TCP or UDP (Taylor et al., n.d.) on both sides. On the task controller side, a standard c++ client-server application is used. On the BCImat side, a MATLAB executable version of the server and client VRPN classes are implemented to read the parameters for calibration and send the decoded parameters. Since the communication is established via IP network, the task controller and the BCI can run on different computers. The framework is implemented in object-oriented MATLAB to exploit modularity. This makes it possible to interchange different decoders or to optimize decoding performance or to provide additional functionalities, for example, to perturb neural parameters for decoding in BCI learning experiments. Further supporting modularity, BCImat can communicate with other task controllers written in any other programming language provided that VRPN is used to stream and read information to and from the BCI framework. In addition, BCImat can be interfaced with any other recording hardware provided that spiking activity can be streamed in real-time and the Trial Spike Buffer format maintained. To use BCImat without recording hardware, an artificial spiking neural network (SNN) is implemented. The task-controller provided here allows performing a center-out reach task with computer mouse movements. While the subject performs the manual task, the simulated neurons fire accordingly with a Poisson process the frequency of which is proportional to the cosine of the angle between the direction of movements in the task and their preferred direction. Thereby, the SNN simulates neural response patterns of primate motor cortex during reaching tasks (Georgopoulos et al., 1982). In practice, the user would first execute the manual motor task to calibrate the parameters of the decoding algorithm. Once switching from manual task execution to "BCI control" after calibration, the decoder output, i.e., the computer cursor movements, determines the neurons firing pattern depending on their own dynamics (cosine model) relative to the actual movement direction. During this closed-loop online control in the simulation mode, the cursor movements are therefore controlled via the decoder by the SNN and no longer by real computer mouse movements. Later, in the application mode, the subject's brain activity would replace the SNN output. The code was successfully used from extracted neural activity of rhesus monkeys performing a similar center-out reach as previously described in a virtual-reality environment (Ferrea et al., 2021).
The implemented decoder is a Kalman-Filter for motor control BCI applications (Wu et al., 2006). We also implemented some important BCI features that are frequently used in the literature to provide efficient training and better decoding performance. In particular, we implemented the possibility to re-train the Kalman filter during online control [**RN1**], an assisted computer cursor control during closed-loop trials (Collinger et al., 2013) to perform calibration in absence of movements, rotation of unit preferred directions resulting in movement direction rotations (Jarosiewicz et al., 2008), and the possibility to perform open-loop testing of the decoder (for review see (Shenoy & Carmena, 2014)).

The Journal of Open Source Software

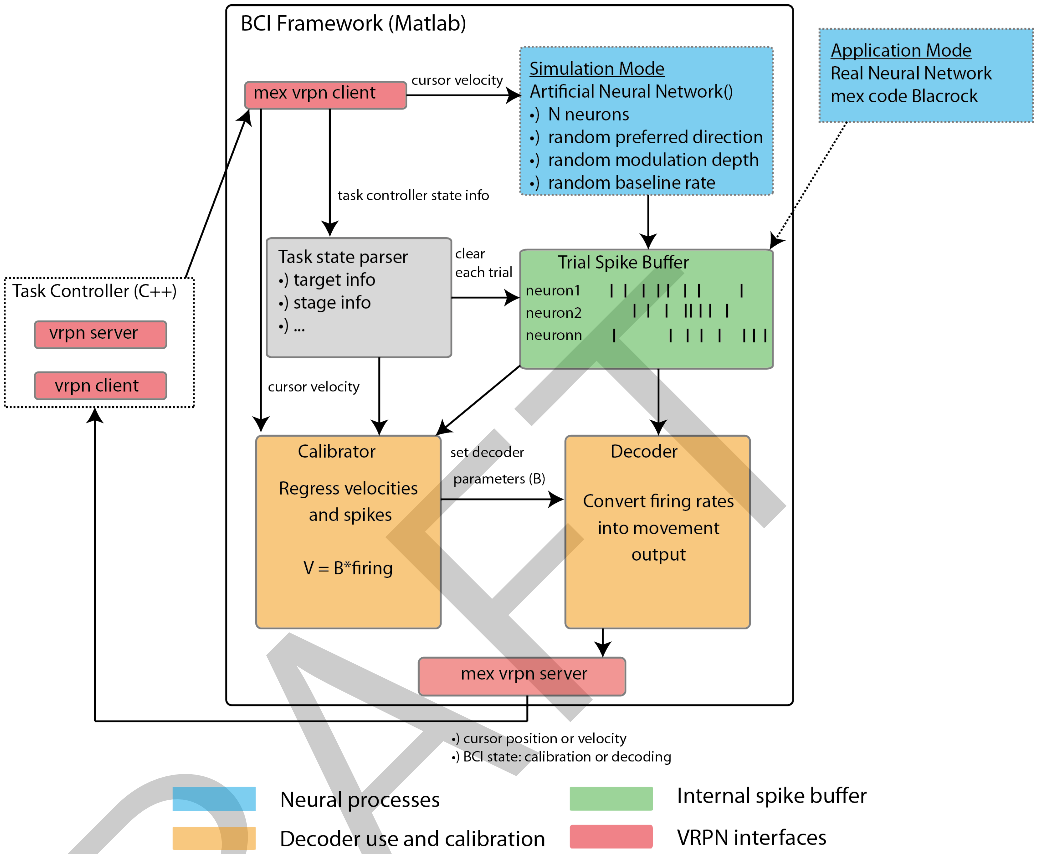

Figure 1: Figure 1: BCI framework schematics. The BCImat framework interfaces with a task controller to display movements of a cursor on a computer screen and with a neural interface providing spiking signal to ultimately (after decoder calibration) control cursor movements. The neural signal fills an internal spike buffer. The spike buffer can be interfaced with an artificial spiking neural network (SNN) for stand-alone (patient independent) experiments (simulation mode), as well as with an external physiological intracortical interface providing spike data in real-time (application mode). Our system was tested in application mode with a Matlab executable (mex) code to stream online spikes recorded with a Cereplex system (Blackrock, Salt Lake City, USA). Task controller and BCI framework exchange messages, cursor position or velocities data via VRPN clients and servers. Inside the BCI framework, the cursor velocities are used to calibrate the decoder by regressing them with simultaneously recorded neural activity. A task state parser is used internally to the framework to handle received messages from the BCI. It can be expanded to handle any type of message. Here for example, movement stage information and position of the target are handled to clear the spike buffer regularly to avoid excessive memory overload.

References

Aflalo, T., Kellis, S., Klaes, C., Lee, B., Shi, Y., Pejsa, K., Shanfield, K., Hayes-Jackson, S., Aisen, M., Heck, C., Liu, C., & Andersen, R. A. (2015). Neurophysiology. Decoding motor imagery from the posterior parietal cortex of a tetraplegic human [Journal Article]. *Science*, *348*(6237), 906–910. https://doi.org/10.1126/science.aaa5417

Ajiboye, A. B., Willett, F. R., Young, D. R., Memberg, W. D., Murphy, B. A., Miller, J. P., Walter, B. L., Sweet, J. A., Hoyen, H. A., Keith, M. W., Peckham, P. H., Simeral, J.

The Journal of Open Source Software

D., Donoghue, J. P., Hochberg, L. R., & Kirsch, R. F. (2017). Restoration of reaching and grasping movements through brain-controlled muscle stimulation in a person with tetraplegia: A proof-of-concept demonstration [Journal Article]. *Lancet*, *389*(10081), 1821–1830. https://doi.org/10.1016/S0140-6736(17)30601-3

Collinger, J. L., Wodlinger, B., Downey, J. E., Wang, W., Tyler-Kabara, E. C., Weber, D. J., McMorland, A. J., Velliste, M., Boninger, M. L., & Schwartz, A. B. (2013). High-performance neuroprosthetic control by an individual with tetraplegia [Journal Article]. *Lancet*, *381*(9866), 557–564. https://doi.org/10.1016/S0140-6736(12)61816-9

Ferrea, E., Morel, P., Franke, J., & Gail, A. (2021). Statistical determinants of visuomotor adaptation in a virtual reality three-dimensional environment [Journal Article]. *bioRxiv*.

Georgopoulos, A. P., Kalaska, J. F., Caminiti, R., & Massey, J. T. (1982). On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex [Journal Article]. *Journal of Neuroscience*, *2*(11), 1527–1537.

Hochberg, L. R., Bacher, D., Jarosiewicz, B., Masse, N. Y., Simeral, J. D., Vogel, J., Haddadin, S., Liu, J., Cash, S. S., Smagt, P. van der, & Donoghue, J. P. (2012). Reach and grasp by people with tetraplegia using a neurally controlled robotic arm [Journal Article]. *Nature*, *485*(7398), 372–375. https://doi.org/10.1038/nature11076

Jarosiewicz, B., Chase, S. M., Fraser, G. W., Velliste, M., Kass, R. E., & Schwartz, A. B. (2008). Functional network reorganization during learning in a brain-computer interface paradigm [Journal Article]. *Proc Natl Acad Sci U S A*, *105*(49), 19486–19491. https://doi.org/10.1073/pnas.0808113105

Koralek, A. C., Jin, X., Long, 2nd., J. D., Costa, R. M., & Carmena, J. M. (2012). Corticostriatal plasticity is necessary for learning intentional neuroprosthetic skills [Journal Article]. *Nature*, *483*(7389), 331–335. https://doi.org/10.1038/nature10845

Sadtler, P. T., Quick, K. M., Golub, M. D., Chase, S. M., Ryu, S. I., Tyler-Kabara, E. C., Yu, B. M., & Batista, A. P. (2014). Neural constraints on learning [Journal Article]. *Nature*, *512*(7515), 423–426. https://doi.org/10.1038/nature13665

Shenoy, K. V., & Carmena, J. M. (2014). Combining decoder design and neural adaptation in brain-machine interfaces [Journal Article]. *Neuron*, *84*(4), 665–680. https://doi.org/10.1016/j.neuron.2014.08.038

Stegman, P., Crawford, C. S., Andujar, M., Nijholt, A., & Gilbert, J. E. (2020). Brain–computer interface software: A review and discussion [Journal Article]. *IEEE Transactions on Human-Machine Systems*, *50*(2), 101–115.

Taylor, R. M., Hudson, T. C., Seeger, A., Weber, H., Juliano, J., & Helser, A. T. (n.d.). VRPN: A device-independent, network-transparent VR peripheral system [Conference Proceedings]. *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, 55–61.

Wu, W., Gao, Y., Bienenstock, E., Donoghue, J. P., & Black, M. J. (2006). Bayesian population decoding of motor cortical activity using a kalman filter [Journal Article]. *Neural Computation*, *18*(1), 80–118.