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

Enrico Ferrea[1,5¶], **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** ULR 7369 - URePSSS - Unite de Recherche Pluridisciplinaire Sport Santé Sociéte, Univ. Littoral Côte d'Opale, Univ. Lille, Univ. Artois, Calais, France. **5** Institute for Neuromodulation and Neurotechnology, University Hospital and University of Tuebingen, Tuebingen, Germany ¶ Corresponding author

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Olivia Guest

Reviewers:

- @mstimberg
- @puolival

Submitted: 20 August 2021
Published: unpublished

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 of this type have been and still are being developed mostly in non-human primate animal models. Additionally, they are also used for basic neuroscientific studies in animals to establish a specific experimentally-controlled transformation between the brain area under investigation and a specific behavior(Koralek et al., 2012; Sadtler et al., 2014), thereby imposing a direct and controllable causal link between brain activity and behavior. The software introduced here allows true online BCI control of a computer cursor based on physiological signals. Importantly, it also allows realistic real-time neural data simulations from artificial spiking neural network (SNN). With this, all algorithms and the control architecture can be tested in silico identical to the physiological experiment.

Statement of need

Software for simulating, testing, and applying BCIs based on intracortical recordings of neural spiking activity is not publicly available. 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). BCImat instead is a Matlab framework for implementing and testing a BCI 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. Importantly, BCImat can alternatively use as input simulated data from a built-in artificial spiking neural network (SNN). This allows testing BCI applications with the same algorithm and framework as intended for later use in BCI experiments but before the availability of recordings from implanted animals or human patients. 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. Here the method was tested in rhesus monkeys but it is as well suited for use in other species (mice, humans).

Overview

The software that defines the BCI framework (BCImat, Fig. 1) interfaces bi-directionally with the software that serves as a simple behavioral task controller (here written in c++, Task Controller Fig. 1). The task controller allows computer-controlled behavioral experiments in which subjects perform center-out reach movements by moving a cursor on a screen with the computer mouse. The BCI framework interacts with the task controller for the purpose of decoder calibration, when first training the machine-learning algorithms to link the physical movements of the hand to the corresponding neural activity patterns, and then later to control cursor movements exclusively based on the neural activity patterns. 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 Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) (Taylor et al., 2001) on both sides. On the task controller side, a c++ server application is used to provide information to BCImat about the stages of the behavioral task and the position of the cursor on the screen. On the same side, a client application is implemented to update the cursor position on the screen that is read from BCImat. 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 Internet Protocol (IP) network, the task controller and the BCI can run on different computers. The BCI framework is implemented in object-oriented Matlab to exploit modularity. This makes it possible to use more advanced decoders (e.g., non-linear long-short-term-memory (LSTM) networks, Transformers) as alternatives in order to optimize decoding performance or to provide additional functionalities (e.g., to perturb neural parameters for decoding to experimentally induce neural plasticity in BCI learning studies). 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. For real intracortical BCI experiments (application mode, Fig. 1), the presented package is implemented for interfacing a Cereplex system (Blackrock, Salt Lake City, USA), but can easily be adapted to other common data acquisition systems. In fact, BCImat can be interfaced with any other recording hardware provided that the same internal buffer structure to store spiking activity in real-time is programed. To use BCImat without recording hardware, an artificial spiking neural network (SNN) is implemented (simulation mode, Fig. 1). In the simulation mode, we decided to implement the same spike buffer structure of the application mode in order to keep compatibility when switching among the two modes. The task-controller provided here allows performing a center-out reach task with computer mouse movements. While the subject performs the task manually, i.e. by actually physically moving the arm (manual task), the simulated neurons fire accordingly. Mimicking stochastic properties of neural activity in the brain, the firing pattern is simulated as a Poisson process. The frequency of firing 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 in the simulation mode, the user would first execute the manual task to calibrate the parameters of the decoding algorithm. After calibration, one will switch from manual task execution to closed-loop "BCI control". During this closed-loop, the cursor movements are controlled via the decoder output and no longer by real computer mouse movements. At the same time, the user should still perform the task with the mouse, since the cursor movement determines the neurons firing pattern of the SNN, which serves as input to the decoder. The SNN firing depends on its own neural dynamics driven by the cosine model relative to the mouse movement direction. Later, in the application mode, the subject's brain activity would replace the SNN output, and physical movements would be no longer required to move the cursor. To demonstrate functionality and performance, the BCImat code was successfully used with neural activity recorded in motor cortical areas of two rhesus monkeys performing a center-out reach task in a virtual-reality environment, similar to a setup that was previously described (Ferrea et al., 2022). Both animals were housed in social groups with one or two male conspecifics in facilities of the German Primate Center. The facilities provide cage sizes exceeding the requirements by German and European regulations, and access to an enriched environment including wooden structures and various toys. All procedures have been approved by the responsible regional government office [Niedersächsisches Landesamt für Verbraucherschutz und Lebensmittelsicherheit (LAVES)] under permit numbers 3392 42502-04-13/1100 and comply with German Law and the European Directive 2010/63/EU regulating use of animals in research. Both animals learned to control the cursor via BCImat. The time that the animals needed to move the cursor to the targets was in the same order of magnitude during manual and BCI task performance in both animals (animal 1: median hand movement time = 404 ms, animal 1: median BCImat movement time = 585 ms, animal 2: median hand movement time = 418 ms, animal 2: median BCImat movement time = 743 ms). Note that the observed decrease in performance comparing hand movements with BCI movements is generally expected. 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. These features were found to provide efficient training and better decoding performance. In particular, we implemented (i) the possibility to re-train the Kalman filter during online control (Gilja et al., 2012), (ii) an assisted computer cursor control during closed-loop trials (Collinger et al., 2013) to perform calibration in absence of movements, (iii) rotation of unit preferred directions resulting in movement direction rotations (Jarosiewicz et al., 2008), and (iv) the possibility to perform open-loop testing of the decoder (for review see (Shenoy & Carmena, 2014)).

The Journal of Open Source Software

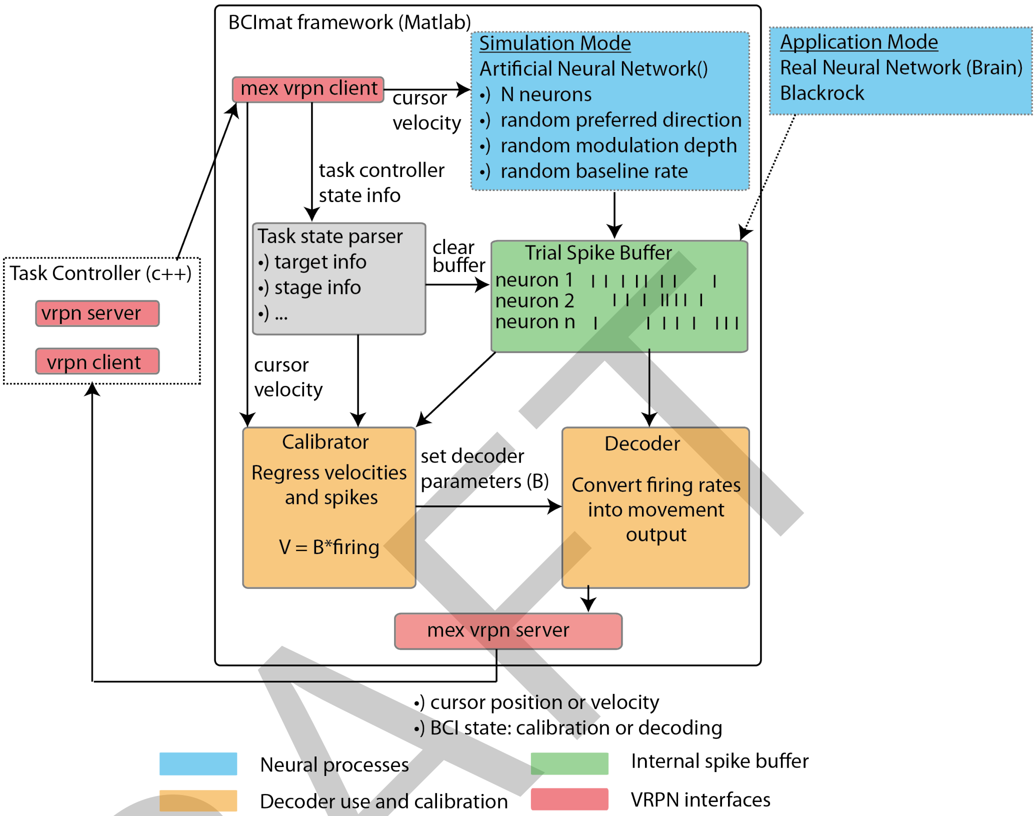

Figure 1: BCImat framework. The BCImat framework interfaces with a task controller to display movements of a cursor on a computer screen and with a neural interface providing spiking signal to ultimately (after decoder calibration) control cursor movements. The neural signal fills an internal spike buffer. The spike buffer is fed with input from an artificial spiking neural network (SNN) for stand-alone (patient independent) experiments (simulation mode), or with external intracortical physiological signals providing neural 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 in BCImat to handle received messages from the task controller. It can be expanded to handle any type of message. Here, for example, information about the completion of a movement is handled to clear the spike buffer regularly to avoid memory overload.

Funding:

German Research Foundation (DFG, Germany, grant number FOR-1847-GA1475-B2). Received by AG. German Research Foundation (DFG, Germany, grant number SFB-889). Received by AG. Federal Ministry for Education and Research (BMBF, Germany, grant number 01GQ1005C). Received by AG.

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. 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., Franke, J., Morel, P., & Gail, A. (2022). Statistical determinants of visuomotor adaptation along different dimensions during naturalistic 3D reaches. *Scientific Reports*, *12*(1), 1–14. https://doi.org/10.1038/s41598-022-13866-y

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. https://doi.org/10.1523/JNEUROSCI.02-11-01527.1982

Gilja, V., Nuyujukian, P., Chestek, C. A., Cunningham, J. P., Yu, B. M., Fan, J. M., Churchland, M. M., Kaufman, M. T., Kao, J. C., Ryu, S. I., & Shenoy, K. V. (2012). A high-performance neural prosthesis enabled by control algorithm design [Journal Article]. *Nature Neuroscience*, *15*(12), 1752–1757. https://doi.org/10.1038/nn.3265

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]. *The Proceedings of the National Academy of Sciences*, *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. https://doi.org/10.1109/THMS.2020.2968411

Taylor, R. M., Hudson, T. C., Seeger, A., Weber, H., Juliano, J., & Helser, A. T. (2001). VRPN: A device-independent, network-transparent VR peripheral system [Conference Proceedings]. *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, 55–61. https://doi.org/10.1145/505008.505019

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. https://doi.org/10.1162/089976606774841585