AxoPy: A Python Library for Implementing Human-Computer Interface Experiments

Kenneth R. Lyons[1] **and Benjamin W. L. Margolis**[1]

1 University of California, Davis

Summary

AxoPy is a system for creating human-computer interface experiments involving the use of electrophysiological signals, such as electromyography (EMG) or electroencephalography (EEG). It is intended to provide an infrastructure for rapidly developing common kinds of experiments while allowing for more complex, customized designs.

In human-computer interface studies, experiment designs can often be organized as a series of tasks. Each task in the experiment may need to record data from a data acquisition system, log data, read data from previous tasks, and provide a graphical user interface. AxoPy offers a framework for implementing these tasks and their various input/output operations. While AxoPy doesn't include as much built-in functionality as alternatives such as PsychoPy (Peirce, 2007) or Expyriment (Krause & Lindemann, 2014), the task-based design is convenient for implementing complex experiment designs that require many inter-dependent tasks and processing operations. It also emphasizes the use of Python code as the experiment implementation, encouraging best practices in creating experiments that are reproducible and extensible.

AxoPy provides the following core functionality:

- Graphical interface:

  Central to AxoPy is the graphical user interface providing visual feedback to the subject and controlling the flow of the experiment. The GUI is backed by PyQt5, and the researcher is free to implement customized graphical elements if those built in to AxoPy aren't sufficient.

- Data acquisition:

  AxoPy establishes an API for communicating with input hardware, so all that's usually needed is a bit of middleware to get going. A couple input devices are built in (keyboard, noise generator), so examples can be run without needing special hardware.

- Data storage:

  Data is stored in a file structure with common file formats (CSV and HDF5) so the researcher can a) start working with data as soon as an experiment session is over and b) standard tools (e.g. pandas, h5py) and even different programming languages can be used to work with datasets. A high-level interface to the storage structure is also provided to make traversing a dataset painless.

- Pipeline processing:

  Estimating intentions of a user from raw electrophysiological signals often involves a large number of processing operations. AxoPy facilitates flexible construction of pipelines that can be reused in different parts of an experiment and re-used for offline post-processing, and many processing operations are built in such as windowing, filtering, and feature extraction.

AxoPy has been used to implement research experiments involving the use of surface electromyography for applications like prosthesis control (Lyons & Joshi, 2018) and communication devices (O'Meara, Shyr, Lyons, & Joshi, 2019).

References

Krause, F., & Lindemann, O. (2014). Expyriment: A Python library for cognitive and neuroscientific experiments. *Behavior Research Methods*, *46*(2), 416–428. doi:10.3758/s13428-013-0390-6

Lyons, K. R., & Joshi, S. S. (2018). Effects of mapping uncertainty on visuomotor adaptation to trial-by-trial perturbations with proportional myoelectric control. In *2018 40th annual international conference of the IEEE engineering in medicine and biology society (EMBC)* (pp. 5178–5181). Honolulu, HI, USA. doi:10.1109/EMBC.2018.8513412

O'Meara, S. M., Shyr, M. C., Lyons, K. R., & Joshi, S. S. (2019). Comparing two different cursor control methods which use single-site surface electromyography. In *2019 9th international IEEE/EMBS conference on neural engineering (NER)*. San Francisco, CA, USA.

Peirce, J. W. (2007). PsychoPy - psychophysics software in Python. *Journal of Neuroscience Methods*, *162*(1–2), 8–13. doi:10.1016/j.jneumeth.2006.11.017