Implement the RIDE algorithm into the Unfold.jl toolbox

Master Thesis Proposal

1st Prölß Till
Socially Intelligent Robotics Lab
Institute for Artificial Intelligence
Stuttgart, Germany
st148545@stud.uni-stuttgart.de

***Abstract*—**

***Index Terms*—RIDE, Unfold, EEG, ERP, Julia**

I. Introduction

In an attempt to understand the brain's inner workings, researchers and doctors have been utilizing the Electroencephalography (EEG) method. Electrodes are attached to a subject's scalp, as seen in figure 1. These electrodes are then used to measure and record the electrical activity in the brain on an electrogram. This method can be helpful in the diagnosis and treatment selection of epilepsy or other brain disorders. Additionally, the gathered data can be analyzed to gain further insight into the brain's mechanisms.

One of these analysis methods is the event related potential (ERP). The idea behind ERP is to relate peaks in the EEG data to events that the subject experiences. In a typical experiment, the researcher applies a stimulus to the test subject in order to evoke a reaction in the brain. This stimulus can be anything that is detectable by the human senses, like the image of a face shown on a screen. After the stimulus is applied, a peak or potential should appear on the EEG, however such potentials are usually obscured by other unrelated brain activity.

Since this potential is triggered by the stimulus, we can assume that it will always appear at the same time point (offset) after the stimulus onset (the moment the stimulus is applied). This consistency is of course not present in the unrelated brain activity. We can use this fact to filter out the random background noise by repeating the same experiment multiple times, then aligning the recorded data by their stimulus onset and calculating the average. As the event we are interested in has a consistent offset, we can expect it to be present in every single trial of the experiment and it shouldn't be affected by the averaging process. The random background noise on the other hand should cancel itself out over many iterations. This averaging process can be seen in figure 2. Any positive or negative peaks in the resulting graph are then labeled, as seen in figure 3, and interpreted by the researcher. We refer to the labeled events as components.

Creating an ERP through simple averaging is a valid approach, but it isn't a perfect solution. There are two notable

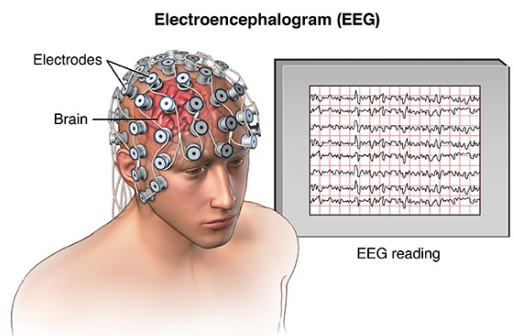

Fig. 1. Example of an EEG measurement setup[19]

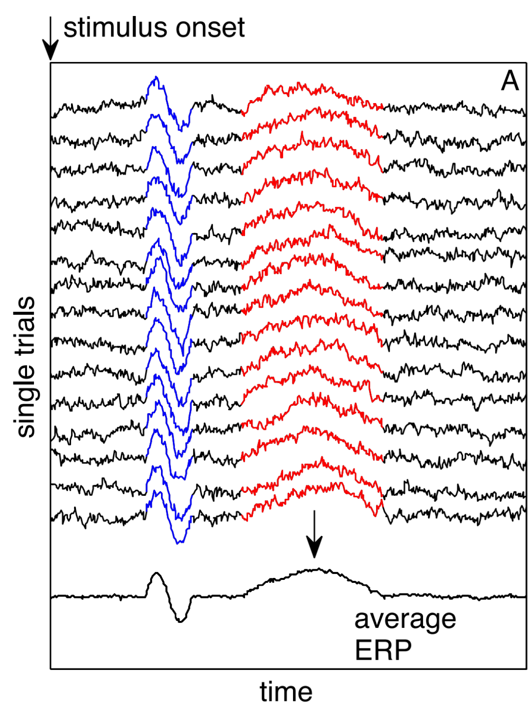

Fig. 2. This figure shows the construction of an ERP graph from multiple single EEG trials. All the trials at the top are averaged together to create the ERP graph at the bottom. Note how the blue and red component clusters are clearly visible in the resulting ERP while the unrelated black background noise has been filtered out. Adapted from [17]

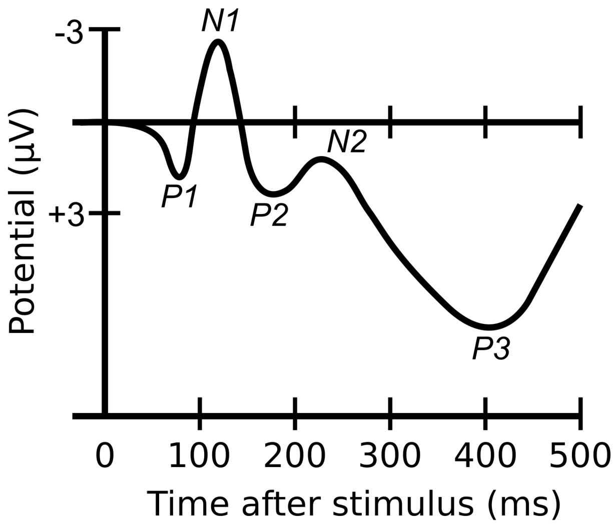

Fig. 3. Example of an ERP graph. The ERP is calculated by averaging the EEG data of many different trials. Every positive and negative peak is labeled and represents one component.[11]

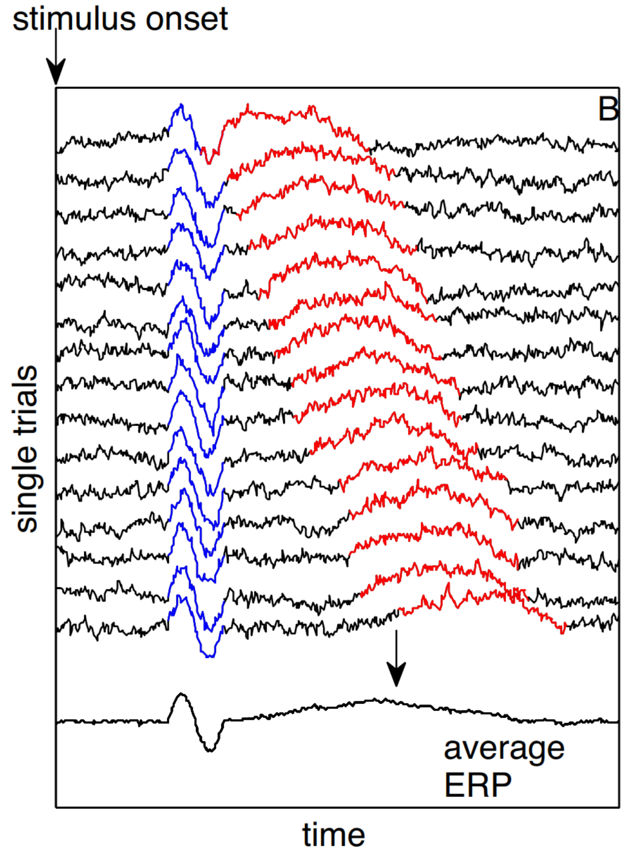

Fig. 4. This figure shows an EEG graph with multiple trials and their computed average ERP at the bottom. The blue component cluster has a stable latency from the stimulus onset, while the red component cluster shows some variability. You can see the smearing caused by the inconsistency of the red component in the averaged ERP at the bottom, compared to the clear representation of the blue cluster. Adapted from [17].

situations where averaging can lead to inaccurate results, namely when multiple components overlap or when there are components with a variable offset from the stimulus onset. The resulting graph doesn't accurately represent the underlying components, as seen in figure 4. To combat this issue, Ouyang et al. developed the residue iteration decomposition (RIDE) algorithm [18, 17]. RIDE is able to extract the individual components from a set of trials, even when they overlap or have a variable latency. These components can then be examined individually or reassembled to form a clean ERP graph.

Another approach to solve the problem of convoluted signals is implemented in the Unfold [10] toolbox. It uses a regression-based approach to deconvolve overlapping signals, but is so far unable to account for any latency jitter between single components. The Unfold project is actively maintained and built in a modular fashion to give its users a high degree of control over how each step of the deconvolution is performed. Unfold also includes many other useful features, like the visualization of final or intermediate results with UnfoldMakie.jl [9] or the simulation of EEG data through UnfoldSim.jl [21]. This flexibility makes Unfold an attractive option for anyone performing ERP analysis.

Currently, the RIDE toolbox is written as a Matlab project[20]. The goal of this thesis is to re-implement the RIDE algorithm in Julia and integrate it as a new feature into the Unfold toolbox. This will make RIDE available to unfold users, allow the seamless execution of RIDE on Unfold compatible datasets and in turn allow for a much easier comparison between the results generated by RIDE and the Unfold regression-based approach.

In addition to translating the algorithm, a modification to the iterative part of the RIDE algorithm is also planned. It will be replaced with a regression-based method, which should again improve performance and is the first step in implementing the covariate estimation that is planned as an optional goal of this work.

II. Related Work

The RIDE algorithm itself is described in two papers from Ouyang et al.[17][18]. This includes the previous implementation of the algorithm in Matlab[20]. To replace the iterative step in RIDE with a regression-based approach, we plan on utilizing smiths et al. [22][23] work, as well as the current paper about the Unfold toolbox [10].

RIDE has found use in a number of studies to better understand the inner workings of the brain. It has been used to better understand brain-related conditions, such as ADHD[4] and Tourette syndrome[13]. Additionally, it has found use in mapping the brain's inner workings, e.g. by investigating the theory of event codes[6, 24] or analyzing the results of a flanker task[14].

Just like RIDE, the Unfold toolbox [10] has also been used in numerous studies to improve EEG data processing. One interesting avenue of study is the combination of EEG data

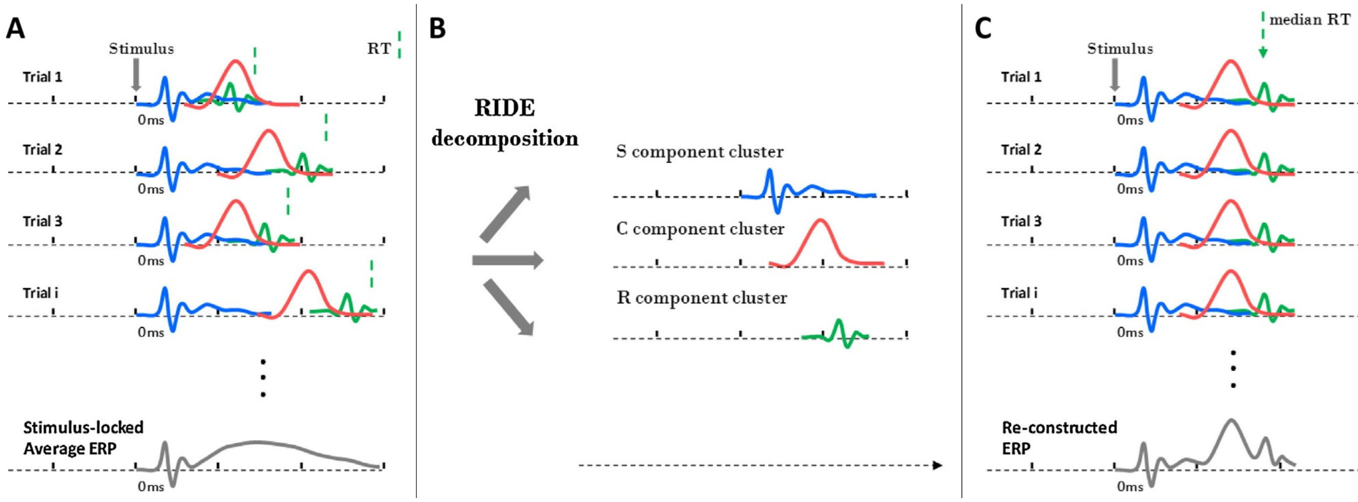

Fig. 5. This figure shows the different stages during the execution of the RIDE algorithm. Part A shows the individual trials at the beginning, with the three different component clusters in different colors. The S-cluster aligned to the stimulus onset in blue, the R-cluster aligned to the response time (RT) in green and the C-cluster with a variable latency in red. The calculated ERP is also shown at the bottom, which shows the smearing caused by the overlapping of the different clusters and the variable latency from the stimulus onset. Part B shows the extracted component clusters after the RIDE decomposition. The latency of the C-cluster is also estimated during this step. Part C shows the reconstruction of the trial data with each component cluster aligned on a common latency. The ERP is re-constructed from the new trial data and now clearly shows the different components. Adapted from [17]

with Eye tracking, which has been investigated by Dimigen and Ehinger et al. [7, 8]. They track the focus of the participant's eye to determine the moment when a particular image is noticed, which then corresponds to the stimulus onset for the EEG analysis. This approach enables them to do free-viewing experiments and seems to have been adopted by the research community for similar purposes [2, 1].

This free-viewing is unusual, as neural mechanisms are usually studied in strictly controlled experiments [2]. The deconvolution abilities of Unfold enable less stringent experiment setups, like matching a beat with a drum [26] or using a map to navigate a virtual city [3].

Another interesting avenue of study where Unfold has proven useful lies in analyzing the brain activity during reading activities [12, 16]. Some of these studies involved children with dyslexia [5, 15], in an attempt to better understand reading difficulties and help develop methods to improve reading ability.

III. Residue Iteration Decomposition (RIDE)

In the following chapter, we will go into detail how the RIDE algorithm operates. As an input, RIDE requires trial data from a controlled experiment.

During a typical experiment, a person is given some sort of stimulus, like showing an image of a face, and asked to respond to it in some manner, like pressing a button. The brain activity of the subject is measured using an EEG. Usually, we would now take the average of many different trials to create an ERP. In this case, we assume that our ERP contains some overlapping components or components with a variable latency. This results in smearing in our ERP, as can be seen in figure4 or figure5A, hiding the individual components. This is where RIDE becomes a viable option to deconvolve the individual components and create a clean re-constructed ERP.

In the RIDE algorithm, we separate the components into three different clusters, as seen in figure 5B:

- S-cluster (Stimulus-Locked): Aligned with stimulus onset; captures early sensory and perceptual responses.
- R-cluster (Response-Locked): Aligned with the participant's response; captures motor preparation and decision-related responses.
- C-cluster (Latency-Variable): Captures variability in latency that is not strictly aligned with stimulus or response; includes cognitive processes with variable timing.

The goal of the RIDE algorithm is to identify the different component clusters, extract them from the overall graph, then calculate the Latency of the C cluster. This is accomplished in four steps:

1) Initial estimation of the latency of C: A value for the latency of the C-cluster is required to run the algorithm and updated in each subsequent iteration. For the first iteration, an initial value is estimated by averaging the peak of a selection of single trials or by using woody's method [25].
2) Decomposition of the S/C/R clusters: The clusters are extracted one by one, starting with S which is achieved by subtracting C and R from every single trial and averaging the result. This step is repeated multiple times, until the resulting clusters stop differing from the previous run.
3) Re-estimate the latency of C: Using the extracted C from the previous step, the Latency of C is recalculated through pattern matching.
4) Iterate steps 2 and 3: Compare the latency of C to the initial estimation or the result of the previous iteration.

Repeat steps 2 and 3 when the difference is larger than a certain threshold.

In the current Matlab implementation[20], the iteration in the fourth of the above-mentioned steps is only run a maximum of four times. A comment in the code explains this as being "empirically limited to 4", but it is unclear what exactly this means. Further, the iterative process in the decomposition step is also limited to a maximum of 100 runs to safeguard against extreme iterations and because most iterations stop before 20 runs anyway. A similar reasoning could explain the limitation in step four, when more iterations are not expected to improve the results. However, four iterations seem like a very low limit, so it might require further investigation during the thesis as to why this limit was implemented and whether it is sensible.

After the algorithm concludes, the C and R clusters in each trial can be realigned to their median latency and a clean ERP reconstructed, as seen in figure 5C.

IV. Unfold toolbox

The Unfold toolbox[10] is a software toolkit designed for the analysis of EEG data, with a particular focus on non-linear modeling of event-related potentials (ERPs). It was originally written in Matlab, similar to RIDE, and later translated into Julia. By now, the toolbox is split into six different packages, each providing different functionality, ranging from visualization to data simulation and statistics. The main purpose of the toolbox is to enable deconvolution regression, which is implemented in the core Unfold.jl package. A full explanation of the deconvolution mechanism is outside the scope of this expose, but can be found in the related paper by Ehinger et al. [10].

The toolbox is written in a modular fashion, enabling the user to perform the regression step by step and examine intermediate results. Many of the components, like the formula, spline and solver are also customizable, giving the user fine control over the regression process.

When comparing RIDE to the Unfold approach, both methods can handle the disentanglement of convoluted signals, however only RIDE is capable of dealing with variable latency components.

V. Structure of the Thesis

The plan is to first analyze the existing Matlab toolbox. The code will be categorized into different modules, that can be translated and tested individually. Each module should also receive some documentation about its purpose, expected input and output.

A dataset has to be simulated for testing purposes, which we plan to accomplish with the UnfoldSim.jl package. The Matlab version will be run with the simulated data to generate an expected output for each module. This might require some transformation of the data into a Matlab and RIDE-compatible format. Later, the dataset will also be used for testing purposes with the Unfold implementation during the implementation process.

After the modules are defined, we will begin translating the code into Julia and directly integrating it into Unfold. Working with individual packages allows us to properly track the progress of this step. Each module will receive its own dedicated unit test suite to constantly validate the correctness of any necessary modifications. Most of the modifications will most likely be refactorings to increase the readability and structure of the code.

The iterative step of the RIDE algorithm has to be replaced with a regression-based solution. We intend to first formulate a theoretical solution before modifying the Unfold version. How exactly this solution will function and how it can be tested will be developed during the thesis.

Depending on the progress of the project, we will attempt to complete our optional goals. We first plan on preparing a real-world dataset and running both versions of the Unfold RIDE algorithm on said dataset. Then implement the option to estimate covariates as an additional feature to the regression-based RIDE solution.

The final month is allocated to writing the paper, although parts of the work have already been completed during the overall development process. Preparing a proper presentation would also be a part of this segment.

A. *Mandatory and optional goals*

Mandatory goals:

- Re-Implement RIDE in Julia
- Integrate RIDE into the Unfold.jl toolbox
- Replace the iterative step with a single regression estimation
- Evaluate the new implementation with simulated data

Optional goals:

- Evaluate the Unfold implementation with real-world data
- Enable covariate estimation

VI. Schedule

The predicted schedule is outlined in figure 6. All the mandatory goals are planned to be achieved after three and a half months, with the final month reserved for finishing the written paper and preparing the presentation. This leaves one and a half months to serve as a buffer or work on optional goals. Work on the written paper will proceed throughout the entire project to make sure results are immediately documented and to keep the workload in the final month manageable.

References

[1] Andrey R. Nikolaev et al. "Episodic memory formation in unrestricted viewing". In: 266 (Feb. 1, 2023). MAG ID: 4312083095, pp. 119821–119821. DOI: 10.1016/j.neuroimage.2022.119821.

[2] Anna L Gert et al. "WildLab: A naturalistic free viewing experiment reveals previously unknown EEG signatures of face processing". In: *European Journal of Neuroscience* (Sept. 16, 2022). MAG ID: 4296164644. DOI: 10.1111/ejn.15824.

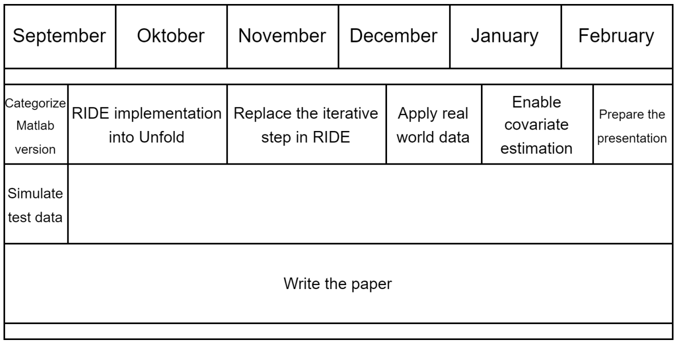

Fig. 6. Predicted work schedule over six months.

[3] Bingjie Cheng et al. “Using spontaneous eye blink-related brain activity to investigate cognitive load during mobile map-assisted navigation”. In: 17 (Feb. 14, 2023). MAG ID: 4320723898. DOI: 10.3389/fnins.2023.1024583.

[4] Annet Bluschke et al. “Neuronal Intra-Individual Variability Masks Response Selection Differences between ADHD Subtypes—A Need to Change Perspectives”. In: *Frontiers in Human Neuroscience* 11 (June 28, 2017). Publisher: Frontiers. ISSN: 1662-5161. DOI: 10.3389/fnhum.2017.00329. URL: https://www.frontiersin.org/journals/human-neuroscience/articles/10.3389/fnhum.2017.00329/full (visited on 08/27/2024).

[5] Catherine Manning et al. “Visual Motion and Decision-Making in Dyslexia: Reduced Accumulation of Sensory Evidence and Related Neural Dynamics.” In: *The Journal of Neuroscience* 42.1 (Jan. 5, 2022). MAG ID: 4226134057, pp. 121–134. DOI: 10.1523/jneurosci.1232-21.2021.

[6] Witold X. Chmielewski, Moritz Mückschel, and Christian Beste. “Response selection codes in neurophysiological data predict conjoint effects of controlled and automatic processes during response inhibition”. In: *Human Brain Mapping* 39.4 (2018). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/hbm.23974, pp. 1839–1849. ISSN: 1097-0193. DOI: 10.1002/hbm.23974. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/hbm.23974 (visited on 08/27/2024).

[7] Olaf Dimigen and Benedikt V. Ehinger. *Analyzing combined eye-tracking/EEG experiments with (non)linear deconvolution models*. Pages: 735530 Section: New Results. Mar. 26, 2020. DOI: 10.1101/735530. URL: https://www.biorxiv.org/content/10.1101/735530v2 (visited on 08/31/2024).

[8] Olaf Dimigen and Benedikt V. Ehinger. “Regression-based analysis of combined EEG and eye-tracking data: Theory and applications”. In: *Journal of Vision*. 1st ser. 21.1 (Jan. 7, 2021), p. 3. ISSN: 1534-7362. DOI: 10.1167/jov.21.1.3. URL: https://doi.org/10.1167/jov.21.1.3 (visited on 08/27/2024).

[9] Benedikt Ehinger. *unfoldtoolbox/UnfoldMakie.jl: v0.1.4*. Version v0.1.4. May 9, 2022. DOI: 10.5281/zenodo.6531996. URL: https://zenodo.org/records/6531996 (visited on 08/28/2024).

[10] Benedikt V. Ehinger and Olaf Dimigen. “Unfold: an integrated toolbox for overlap correction, non-linear modeling, and regression-based EEG analysis”. In: *PeerJ* 7 (Oct. 24, 2019). Publisher: PeerJ Inc., e7838. ISSN: 2167-8359. DOI: 10.7717/peerj.7838. URL: https://peerj.com/articles/7838 (visited on 08/27/2024).

[11] *Event-related potential*. In: *Wikipedia*. Page Version ID: 1231450475. June 28, 2024. URL: https://en.wikipedia.org/w/index.php?title=Event-related_potential&oldid=1231450475 (visited on 08/27/2024).

[12] Joshua Snell et al. “Parallel word reading revealed by fixation-related brain potentials”. In: 162 (May 1, 2023). MAG ID: 4322124135, pp. 1–11. DOI: 10.1016/j.cortex.2023.02.004.

[13] Maximilian Kleimaker et al. “Increased perception-action binding in Tourette syndrome”. In: *Brain* 143.6 (June 1, 2020), pp. 1934–1945. ISSN: 0006-8950. DOI: 10.1093/brain/awaa111. URL: https://doi.org/10.1093/brain/awaa111 (visited on 08/27/2024).

[14] Moritz Mückschel et al. “The norepinephrine system shows information-content specific properties during cognitive control – Evidence from EEG and pupillary responses”. In: *NeuroImage* 149 (Apr. 1, 2017), pp. 44–52. ISSN: 1053-8119. DOI: 10.1016/j.neuroimage.2017.01.036. URL: https://www.sciencedirect.com/science/article/pii/S1053811917300435 (visited on 08/27/2024).

[15] Najla Azaiez et al. “Brain Source Correlates of Speech Perception and Reading Processes in Children With and Without Reading Difficulties”. In: *Frontiers in Neuroscience* 16 (July 19, 2022). MAG ID: 4285798973. DOI: 10.3389/fnins.2022.921977.

[16] Nora Hollenstein et al. “The ZuCo Benchmark on Cross-Subject Reading Task Classification with EEG and Eye-Tracking Data”. In: (Mar. 8, 2022). MAG ID: 4221052551. DOI: 10.1101/2022.03.08.483414.

[17] Guang Ouyang, Werner Sommer, and Changsong Zhou. “A toolbox for residue iteration decomposition (RIDE)—A method for the decomposition, reconstruction, and single trial analysis of event related potentials”. In: *Journal of Neuroscience Methods*. Cutting-edge EEG Methods 250 (July 30, 2015), pp. 7–21. ISSN: 0165-0270. DOI: 10.1016/j.jneumeth.2014.10.009. URL: https://www.sciencedirect.com/science/article/pii/S0165027014003690 (visited on 08/27/2024).

[18] Guang Ouyang et al. “Residue iteration decomposition (RIDE): A new method to separate ERP components on the basis of latency variability in single trials”. In: *Psychophysiology* 48.12 (2011). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1469-8986.2011.01269.x, pp. 1631–1647. ISSN: 1469-8986. DOI: 10.1111/j.1469-8986.2011.01269.x. URL:

https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1469-8986.2011.01269.x (visited on 08/27/2024).

[19] *Papers with Code - EEG*. URL: https://paperswithcode.com/task/eeg-1 (visited on 08/27/2024).

[20] *RIDE matlab toolbox*. URL: https://cns.hkbu.edu.hk/RIDE_files/Page308.htm (visited on 08/27/2024).

[21] Judith Schepers et al. *UnfoldSim.jl*. Version v0.3.2. Feb. 27, 2024. DOI: 10.5281/zenodo.10714844. URL: https://zenodo.org/records/10714844 (visited on 08/28/2024).

[22] Nathaniel J. Smith and Marta Kutas. "Regression-based estimation of ERP waveforms: I. The rERP framework". In: *Psychophysiology* 52.2 (2015). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/psyp.12317, pp. 157–168. ISSN: 1469-8986. DOI: 10.1111/psyp.12317. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/psyp.12317 (visited on 08/27/2024).

[23] Nathaniel J. Smith and Marta Kutas. "Regression-based estimation of ERP waveforms: II. Nonlinear effects, overlap correction, and practical considerations". In: *Psychophysiology* 52.2 (2015). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/psyp.12320, pp. 169–181. ISSN: 1469-8986. DOI: 10.1111/psyp.12320. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/psyp.12320 (visited on 08/27/2024).

[24] Adam Takacs et al. "Connecting EEG signal decomposition and response selection processes using the theory of event coding framework". In: *Human Brain Mapping* 41.10 (2020). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/hbm.24983, pp. 2862–2877. ISSN: 1097-0193. DOI: 10.1002/hbm.24983. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/hbm.24983 (visited on 08/27/2024).

[25] Charles D. Woody. "Characterization of an adaptive filter for the analysis of variable latency neuroelectric signals". In: *Medical and biological engineering* 5.6 (Nov. 1, 1967), pp. 539–554. ISSN: 1741-0444. DOI: 10.1007/BF02474247. URL: https://doi.org/10.1007/BF02474247 (visited on 08/27/2024).

[26] Yan Yan, Laurence T. Hunt, and Cameron D. Hassall. "Reward positivity biases interval production in a continuous timing task". In: (July 7, 2023). MAG ID: 4383499942. DOI: 10.1101/2023.07.06.548049.