P-ary m-sequences database

Martínez-Cagigal et al. 2022

A total of 16 healthy subjects performed copy-spelling tasks in a c-VEP speller based on circular shifting. Nonbinary m-sequences were used with bases 2, 3, 5, 7, and 11; displayed at a 120 Hz rate. A g.USBamp equipment of 16 channels was used at a sampling rate of 256 Hz. The evaluation was carried out in a single session, in which users performed (1) a calibration of 30 trials by looking at a command encoded with the original m-sequence, and a (2) a copy-spelling online task of 32 trials. Each trial was composed by 10 cycles (repetitions of the same code). Also, questionnaires were fulfilled to assess satisfaction and the relative eyestrain while looking at each m-sequence.

NOTE: Recordings of user "zdvm" for bases 2, 3, 5, and 7 had a sampling rate of 600 Hz. The rest of recordings have all a sampling rate of 256 Hz.

Database structure:

- *_additional_info/requirements.txt* Requirements file for the python interpreter. Only the medusa-kernel package is being used (`pip install medusa-kernel`)
- *_scripts/example_load.py* Example script to load a file and export it as a dictionary
- *_scripts/example_processing.py* Example script to process files using a simple CCA-based algorithm on a subject and a specific m-sequence
- *.*zip/* Contains the data for all subjects in folders. Each username was randomized with 4 letters to protect their identity. For each subject, different recordings are available.
    - `Username_X_train.cvep.bson` This file contains the calibration recordings of each user. The X determines the m-sequence used. A total of 6 files like this are available for each subject and sequence, as the 30-trial calibration was split into 6 recordings of 5 trials each.
    - `Username_X_test.cvep.bson` This file contains the copy-spelling recordings of each user. The X determines the m-sequence used. A total of 2 files like this are available for each subject and sequence, as the 32-trial task was split into 2 recordings of 16 trials each. The true labels for each are the same: `ABCDEFGHIJKLMNOP`
    - `Username_X.cvep.mdl` This is the original c-VEP model that was trained with the calibration data.
    - `Username_X_labels.txt` The true labels of the copy-spelling tasks, which are the same for all recordings.
- *_additional_info/*
    - `_age.csv` Age and sex of all the participants
    - `_eyestrain60.csv` Relative eyestrain in the pre- condition when all m-sequences were displayed at 60 Hz, where 10 is a maximal eyestrain and 0 is no eyestrain.
    - `_eyestrain120.csv` Eyestrain ratings for the pre- condition at 120 Hz
    - `_eyestrain120during.csv` Eyestrain ratings for the task condition at 120 Hz

- `_order.csv` Order of presentation of each m-sequence for each subject
- `_questionnaire.csv` Satisfaction questionnaire ratings, where 1 means "totally disagree", 3 is the neutral rating, and 5 means "totally agree"
- *`original_questionnaire.pdf`* Satisfaction questionnaire

Recording file structure:

When loading a file as a Recording class or as a dictionary, useful data is structured as follows:

- *`eeg`* Class containing useful data relative to the EEG recording
  - *`times`* Timestamp samples of the EEG
  - *`signal`* EEG signal organized as samples x channels
  - `fs` Sampling rate
  - *`channel`*_set Contains a structure that organizes the channel set (lables, coordinates, etc). Use channel_set.l_cha to get the labels
- *`cvepspellerdata`* Class containing useful info relative to the paradigm
  - `comma`*`nds_info`* Position of each command in the matrix, including row, column, label and its code (already shifted). In calibration trials, only one command was shown displaying the original m-sequence.
  - *`fps_resolution`* Presentation rate in Hz
  - *`onsets`* Onsets vector containing a timestamp whenever a cycle is starting
  - *`cycle_idx`* For each onset, the cycle displayed
  - *`trial_idx`* For each onset, the trial displayed

Information about the m-sequences:

Although the encoding of each command is already included in `cvepspellerdata.commands_info`, the file `additional_info/p_ary_encoding.json` includes the following information regarding the codes. For each code:

- *`base`* The base of the p-ary m-sequence (i.e. the number of levels)
- *`order`* LFSR order
- *`polynomial`* LFSR primitive polynomial to generate it
- *`lags`* Lag points (in samples) used to shift the code and encode the commands. These lags are already optimized to avoid local minima and maxima
- *`length`* Length of the code
- *`tau`* Average difference in samples between lags
- *`monitor_rate`* Presentation rate in Hz