

# Music BCI (006-2015)

Matthias Treder, Benjamin Blankertz  
*Technische Universität Berlin, Berlin, Germany*

September 5, 2016

## 1 Introduction

We investigated the suitability of musical stimuli for use in a P300 paradigm. To this end, 11 subjects listened to polyphonic music clips featuring three instruments playing together. We devised a multi-streamed oddball paradigm, with each of the 3 instruments playing a repetitive standard musical pattern, interspersed with a randomly occurring deviant musical pattern. Subjects were cued to attend to one particular instrument and ignore the other two. Using regularised linear discriminant analysis, we were able to differentiate between deviants in the attended and deviants in the unattended instruments.

For detailed information on the experiment and the method, refer to [1]. In a further study [2], we analysed the neural representation of tone onsets in the same data using a spatio-temporal filtering approach.

## 2 Experimental paradigm

The experiment had four different experimental conditions featuring different types of music clips. Before each clip, one out of three possible instruments was cued. The subject had to attend to the instrument and mentally count the number of deviants for the cued instrument. After the end of the clip, the subject had to type in the count.

- *SynthPop*: A minimalistic adaptation of *Just can't get enough* by the Synth-Pop band Depeche Mode. A corresponding sample score is depicted in figure 1. It features three instruments: drums consisting of kick drum, snare and hi-hat, a synthetic bass and a keyboard equipped with a synthetic piano sound. The instruments play an adaptation of the chorus of the original song with the keyboard featuring the main melody of the song. Deviants are defined as follows: for the drums, the kick drum on the first quarter note is replaced by eight notes featuring snare and then kick drum; for the bass, the whole 4-tone standard sequence is transposed up by five semitones; for the keyboard, tones 4–6 of the 8-tone standard sequence are transposed. The relative loudness of the instruments has been

set by one of the authors such that all instruments are roughly equally audible.

Panning: none (all instruments panned to center).

Beats-per-minute: 130.

- *Jazz*: Stylistically, the Jazz clips are located half-way between a minimalist piece by Philip Glass and a jazz trio comprising double-bass, piano and flute. Each of the three voices is generated through frequent repetition of a standard pattern composed of 3–5 tones, once in a while replaced by a deviant pattern that differs from the standard pattern in one note. One clip consists of three overlaid voices. The Jazz music clips differ from the SynthPop clips in various ways. The Jazz clips sound more natural. This is achieved by selecting samples of acoustic instruments. In addition, loudness and micro-timing are manually adjusted for each tone of the basic pattern in order to make the entire phrase sound more musical. Apart from timbre (double-bass, piano, flute) and pitch range (low, medium, high), for the Jazz clips, another parameter is used to make the voices independent from each other. Each voice consists of patterns of different length, namely of 3, 4 and 5 beats per pattern. Through rhythmical interference a polymetrical rhythmical texture is generated. For better separation of the musical instruments, also panning is chosen to locate musical instruments in different directions from the user. This independence of the different voices is aimed at helping the user to focus on one particular instrument. The relative loudness of the instruments has been set by one of the authors such that deviants in all instruments are roughly equally audible. In particular, the double-bass had to be amplified, while the flute was turned down.

Panning: flute left, bass central, piano right.

Beats-per-minute: 120

- *SynthPop Solo*: Solo versions of the SynthPop music clips with one of the instruments playing in isolation. Solo versions have been produced for all instruments.
- *Jazz Solo*: Solo versions of the Jazz music clips with one of the instruments playing in isolation. Solo versions have been produced for all instruments.

For each music condition, ten different music clips were created with variable amounts and different positions of the deviants in each instrument. Additionally, we exported solo versions with each of the instruments playing in isolation. Sample stimuli are provided.

### 3 Data

For details on data pre-processing, refer to the papers in the reference list. The data has been pre-processed with the BBCI Matlab toolbox (<https://github.com>).

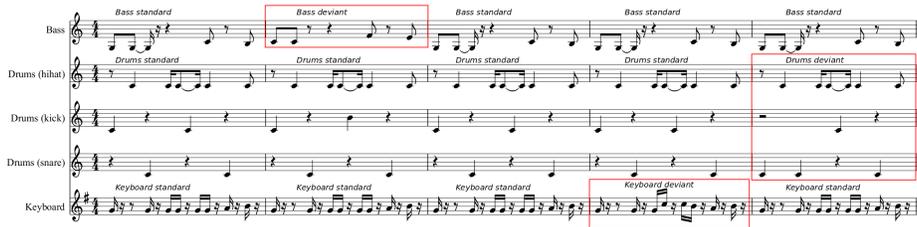


Figure 1: Extract of score for SynthPop stimulus showing the three instruments. The deviant events are marked by red boxes. Event markers for deviant events designate the start of the deviant pattern.

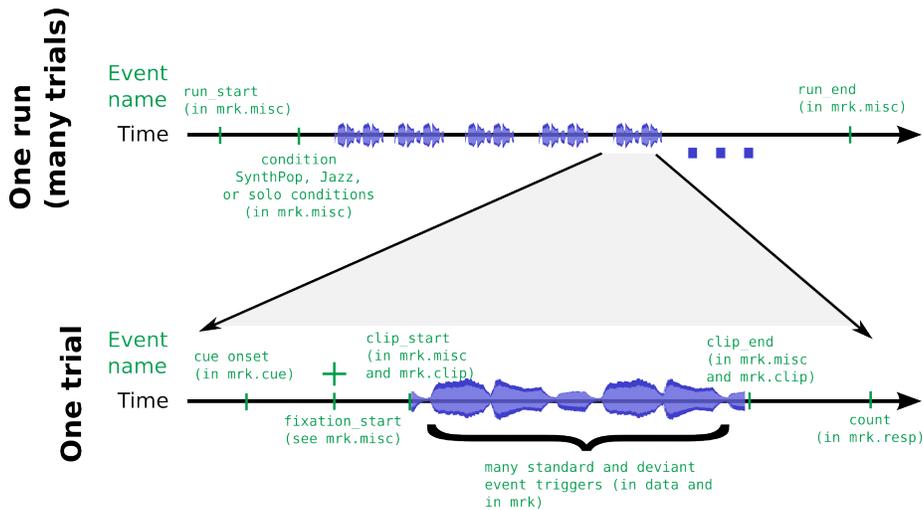


Figure 2: Structuring of the experiment. Each experiment consists of 8 runs. Upper panel: Each run features a particular experimental condition and consists of a number of music clips (indicated by the blue traces). Lower panel: Each music clip is preceded by cue and fixation cross markers. There are many event triggers for attended and unattended standard tones and deviants while the music clip is being played. After the music clip has finished, the subject types in the deviant count.

com/bbci/bbci\_public/). For an introduction to the toolbox, refer to [https://github.com/bbci/bbci\\_public/blob/master/doc/index.markdown](https://github.com/bbci/bbci_public/blob/master/doc/index.markdown). Note that the data can easily be transformed into FieldTrip, EEGLab, and similar data types, although this needs to be done manually as currently there are no automated scripts for this.

## 4 General description of Matlab structs

`data` contains the EEG data. `mrk` contains timing information about the events (i.e. markers/triggers). `mnt` contains details about the montage useful for plotting including channel positions in 2D and 3D.

- `data`
  - `X`: matrix of size [number of time points  $\times$  number of channels]. Contains the continuous EEG data.
  - `fs`: Sampling frequency in Hz.
  - `clab`: Cell array with channel labels.
  - `trial`: time of events in samples relative to the start of the measurement. Can be transformed into milliseconds by `data.trial/data.fs * 1000`.
  - `classes`: Cell array with labels for the different types of events.
  - `y`: Vector of the same length as `trial` indicating, for each trial, which class it belongs to. For instance, if the first 3 entries of `y` are 1, 3, 2, then the first trial belongs to class 1, the second trial belongs to class 3, and the third trial belongs to class 2. In rare occasions, if an entry is 0, it means there is no trial information for this particular event and it should probably be discarded.
  
- `mrk`
  - `time`: time of events in milliseconds relative to the start of the measurement.
  - `y`: logical matrix of size [number of classes  $\times$  number of events]. It contains the same information as `data.y`, but the format is different. Each row of `mrk.y` corresponds to one experimental condition where 1's indicate that the corresponding trial belongs to this condition. Indices correspond to the indices in the `data` struct, that is for the `k`-th class, `find(mrk.y(k,:))` and `find(data.y == k)` give equal results.
  - `className`: same as `data.classes`
  - `event`: The event field contains additional information for each of the events

The `mrk` structure usually contains further sub-structures which have the same structure as the `mrk` struct. These substructures usually carry additional information taken from additional event triggers that were recorded.

- `mnt`
  - `x`: x-position of electrode for 2D plotting
  - `y`: y-position of electrode for 2D plotting
  - `pos_3d`: 3D-position of electrode for 3D plotting

## 5 Selecting events from data

If the data contains different classes/experimental conditions, the event indices corresponding to the first class can be selected using `find(data.y == 1)`. Then, `find(data.y == 2)` yields the event indices for the second class, and so on. The indices can be saved in a variable and then used to obtain the onset times of trials corresponding to the selected class. The onsets can then be used for epoching the data. Example code: `idx1 = find(data.y == 1); trialOnsetsClass1 = data.trial(idx1);`

## 6 Additional information contained in the `mrk` struct

- `mrk.event`

The event field contains additional information for each of the events described in the `y` and `time` fields. In the present dataset, the following additional information is available:

- `condition`: contains the numbers 0, 1, 2, 3, denoting the experimental condition. (0 = SynthPop, 1 = Jazz, 2 = SynthPop Solo, 3 = Jazz Solo).
  - `instrument`: contains the numbers 1, 2, 3 denoting which of the three instruments the onset time corresponds to. In the SynthPop condition, the mapping is 1 = Drum, 2 = Bass, 3 = Keyboard. In the Jazz condition, the mapping is 1 = Flute, 2 = Bass, 3 = Piano.
  - `deviant`: denotes whether the event corresponds to a standard or a deviant tone (0 = standard stimulus, 1 = deviant stimulus).
- `mrk.cue`: Onset of the cue before the start of each music clip. The cue designates which instrument the subject has to attend to. Refer to `mrk.cue.className` for name of the cued instrument. Note that the class names correspond to the SynthPop condition. In the Jazz condition, drums corresponds to the flute, and keyboard corresponds to the piano.

- `mrk.clip`: Onset of the music clips. For each condition, 10 different music clips were created. The classes indicate the number of the clip file.
- `mrk.misc`: Miscellaneous events.
  - `run.start`: Onset of a run. There should be a total of 10 runs, each run featuring one experimental condition (SynthPop, Jazz, SynthPop Solo, or Jazz Solo). Runs 3, 4, 7, and 8 were designated as solo runs.
  - `run.end`: End of run. For some subjects, runs have not been completed due to technical problems. In this case, the run might have been restarted. To use only complete runs, compare the `run.start` and `run.end` onset times (unfinished runs should not have a corresponding `run.end` event).
  - `trial.start`: not used.
  - `trial.end`: not used.
  - `fixation.start`: Onset of the fixation cross that subjects were instructed to fixate throughout the duration of the music clip.
  - `clip.start`: Onset of music clip. These events are the same as the events in `mrk.clip`, but the information about the number of the music clip was dropped.
  - `clip.end`: Offset of music clip. Sometimes clips might not have been completed due to technical problems. To use only complete clips, compare the `clip.start` and `clip.end` onset times (unfinished clips should not have a corresponding `clip.end` event).
  - `condition.depmo`: Start of run in the SynthPop condition.
  - `condition.hendrik`: Start of run in the Jazz condition.
  - `solo.condition.depmo`: Start of run in the SynthPop Solo condition.
  - `solo.condition.hendrik`: Start of run in the Jazz Solo condition.
- `mrk.resp`: Response of the subject. The `count` vector contains the number of deviants counted by the subject during each music clip. The length of the vector should correspond to the number of completed music clips (compare to `clip.end` in `mrk.misc`).

## 7 Selecting events using the `mrk` struct

### 7.1 Selecting an experimental condition

The experimental condition corresponding to an event is saved in `mrk.event.condition` (see above for a specification). Example code for selecting only events belonging to the polyphonic SynthPop condition: `idx = find(mrk.event.condition == 1); data.trial = data.trial(idx); data.y = data.y(idx);`

## 7.2 Selecting an instrument

The instrument corresponding to an event is saved in `mrk.event.instrument` (see above for a specification). Example code for selecting only events corresponding to the Keyboard: `idx_keyboard = find(mrk.event.instrument == 3); data.trial = data.trial(idx_keyboard); data.y = data.y(idx_keyboard);`

## 7.3 Selecting all deviants (attended and unattended)

Indices of only deviant stimuli can be selected by `idx_deviant = find(data.y == 1 | data.y == 2)`. In [1], we selected only deviant stimuli and then trained a classifier to classify between deviants in the attended instrument versus deviants in the unattended instrument.

## References

- [1] Matthias Sebastian Treder, Hendrik Purwins, Daniel Miklody, Irene Sturm, and Benjamin Blankertz. Decoding auditory attention to instruments in polyphonic music using single-trial EEG classification. *J Neural Eng*, 11:026009, 2014.
- [2] Irene Sturm, Matthias Sebastian Treder, Daniel Miklody, Hendrik Purwins, Sven Dähne, Benjamin Blankertz, and Gabriel Curio. Extracting the neural representation of tone onsets for separate voices of ensemble music using multivariate eeg analysis. *Psychomusicology: Music, Mind, and Brain*, 25:366–379, 2015.