Covert shifts of attention (005-2015)

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

September 5, 2016

1 Introduction

Covert shifts of visual attention induce changes in oscillatory alpha activity in posterior cortex, even in the absence of visual stimulation. The aim was to investigate whether different pairs of directions of attention shifts can be reliably differentiated based on the electroencephalogram. To this end, 8 healthy participants had to strictly fixate a central dot and covertly shift visual attention to one out of six cued directions. We found that covert attention shifts induced a prolonged alpha synchronization over posterior electrode sites (PO and O electrodes). Spectral changes had specific topographies so that different pairs of directions could be differentiated. There was substantial variation across participants with respect to the direction pairs that could be reliably classified. Mean accuracy for the best-classifiable pair amounted to 74.6%. Furthermore, an alpha power index obtained during a relaxation measurement showed to be predictive of peak BCI performance (r = .66). Results confirm posterior alpha power modulations as a viable input modality for gaze-independent EEG-based BCIs. The pair of directions yielding optimal performance varies across participants. Consequently, participants with low control for standard directions such as left-right might resort to other pairs of directions including top and bottom. Additionally, a simple alpha index was shown to predict prospective BCI performance.

For detailed information on the experiment and the method, refer to [1, 2]. In a further study the performance of subject-independent classifiers was investigated [3].

2 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/bbci/bbci_public/`). For an introduction to the toolbox, refer to `https://github.com/bbci/bbci_public/blob/master/doc/index.markdown`. Note that the data can easily be transformed into FieldTrip, EEGLab, and similar

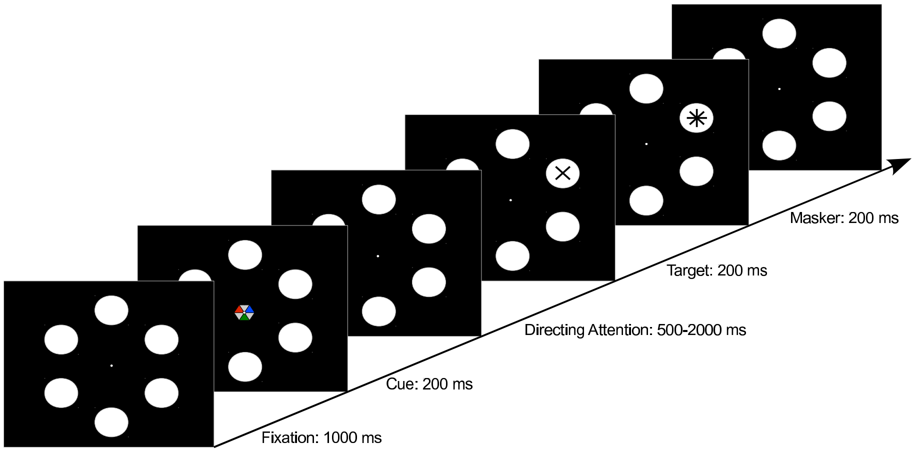

Figure 1: Covert attention task. After 1000 ms, a cue in form of a hexagon appeared. Participants had to attend to either the blue, red, or green face of the hexagon, and they had to covertly shift attention to the disc the face was pointing at. After a variable amount of time (500-2000 ms), a target ('+' or '×') appeared, followed by a masker ('*'). The participant indicated the perceived symbol by means of a button press with the right or left hand.

data types, although this needs to be done manually as currently there are no automated scripts for this.

3 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 × 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 × 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

4 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);`

5 Spatial locations of the cues

In `data.classes`, the cues numbered 1 to 6 correspond to the following locations in the displays: cue 1=top; cue 2=upper right; cue 3=lower right; cue 4=bottom; cue 5=lower left; cue 6=upper left.

It is easy to subselect specific spatial locations. For instance, the 'top' and 'bottom' directions can be selected together by the following code: `idx = find(data.y == 1 | data.y == 4); data.trial = data.trial(idx); data.y = data.y(idx);`

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 most important:

  - `target_latency`: Target latency (i.e., the time between cue onset and target onset) in milliseconds, where the target was either a '+' or a '×'. Target latency was 2000 ms after cue offset in 50% of the trials. To ensure that the participants shift their attention immediately after the appearance of the cue, 30% of the trials featured a short target latency of 500 ms. In the remaining trials, the target latency was randomized between 500 ms and 2000 ms in order to ensure that attention is sustained continuously until target appearance. Type `hist(mrk.event.target_latency)` to see the distribution of the latencies across trials.
  - `valid_cue`: Indicates whether the cued location was indeed the location where the target appeared (1=yes, 0=no). Cues were valid in 80% of the trials. In the other 20% of the cases, the target appeared at a different random location. The target symbol was randomly chosen, with equal chances for '+' and '×'.
  - `ishit`: Indicates whether the subject correctly identified the target (1=correct) or the response was incorrect (0=incorrect).
  - `latency`: response latency in milliseconds. Indicates how fast the subject pressed the button after the onset of the target. Type `hist(mrk.event.latency)` to see the distribution of the response latencies across trials.

  For the following substructures, the onset information in samples is contained in the `pos` field. To convert into milliseconds, use `pos/data.fs * 1000`.

- `mrk.target_location`: Spatial location at which the target appeared. The numbers correspond to the cue numbers, i.e. 1 corresponds to 'top' with clockwise numbering.
- `mrk.target_type`: Indicates the type of the target, '+' or '×'.
- `mrk.misc`: Miscellaneous events.

  - `pause_start`: Not used.
  - `pause_end`: Not used.
  - `run_start`: Onset of a run. In principle, there should be a total of 6 runs, each run featuring 100 trials. For some subjects the `run_start` triggers are missing.
  - `run_end`: End of run.
  - `countdown`: Start of the countdown at the beginning of each run. There should be 6 such triggers.
  - `fixation`: Appearance of the fixation dot. The fixation dot appeared 1000 ms before the cue (see Figure).
  - `directing_attention`: Start of the attention directing phase. Coincides with the offset of the cue (see Figure).
  - `masker`: Onset of the masker '*' that masks the target (see Figure).
  - `response_period`: Onset of the response period in which the subject was supposed to press a button.
  - `rest`: Onset of rest period following the button press.

- `mrk.resp`: For each trial, indicates whether the subject pressed the '+' button or the '×' button.

7 Selecting events using the `mrk` struct

7.1 Selecting all trials with a target latency of 2000 ms

In [1], we only considered trials with the maximum target latency of 2000 ms, i.e. where the target appeared 2000 ms after the cue **off**set. This corresponds to 2200 ms after cue onset since the cue was on the screen for 200 ms. Due to screen refresh and variabilities in presentation time, the 2200 ms were not exactly hit. Therefore we selected all trials with a latency of at least 2000 ms after cue onset. Example code: `idx2000 = ( mrk.event.target_latency >= 2000); data.trial = data.trial(idx2000); data.y = data.y(idx2000);`

7.2 Selecting all trials with a correct response

To select all trials wherein the subject correctly identified the target ('+' or '×'), you can used the following example code: `hits = (mrk.event.ishit == 1); data.trial = data.trial(hits); data.y = data.y(hits);`

References

[1] M S Treder, A Bahramisharif, N M Schmidt, M A van Gerven, and B Blankertz. Brain-computer interfacing using modulations of alpha activity induced by covert shifts of attention. J Neuroeng Rehabil, 8:24–24, 2011.

[2] N M Schmidt, B Blankertz, and M S Treder. α-modulation induced by covert attention shifts as a new input modality for EEG-based BCIs. Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on , 481-487, 2011.

[3] HJP Wouters, MAJ van Gerven, MS Treder, T Heskes, and A Bahramisharif. Covert Attention as a Paradigm for Subject-Independent Brain-Computer Interfacing. Machine Learning and Interpretation in Neuroimaging: International Workshop, MLINI 2011, Held at NIPS 2011, Sierra Nevada, Spain, December 16-17, 2011, Revised Selected and Invited Contributions, 156–163, 2012.