# BCI2000-Compatible Audio-Visual P3 Task

Gerwin Schalk, Jürgen Mellinger, Dennis Mc Farland

May 5, 2003

# Contents

**Abstract**

The main purpose of this task is to present a series of (auditory or visual) stimuli sequentially to the user of the BCI system. The sequence and nature of the stimuli can be defined by the investigator. In addition to stimulus delivery, the task can optionally be used in conjunction with BCI2000's P300 Signal Processing module (P3SignalProcessing.exe) to provide feedback to a selected stimulus in either a copy or a free mode.

# 1 Functionality

## 1.1 Stimulus Definition

Stimuli are set up through a parameter defined by the application module. This implicitly defines the total number of stimuli as well as the details of each stimulus. Each stimulus is defined by the following properties:

1. Caption

2. Icon file

3. Audio file

In addition to stimuli, the parameter contains a definition for a stimulus that announces what to focus on, and a stimulus that announces the result. These stimuli are only used when the task is set to copy or free mode.

The following table contains an example definition of two stimuli; for clarity, the parameter line defining the associated parameter is also given: [1]

|         | focuson | result | stimulus1 | stimulus2 |
|--------:|---------|--------|-----------|-----------|
| caption | Please focus on | The result was | Donkey | |
| icon | icons\focus on.bmp | icons\result.bmp | icons\donkey.bmp | icons\elefant.bmp |
| audio | wav\focus on.wav | wav\result.wav | wav\snicker.wav | wav\trumpet.wav |

---

[1]The line's parameter format contains to newly introduced extensions:

- In the place of a dimension value there may be a list of string labels that may be used instead of numerical indices, delimited by a matching pair of any of {},[],().

- A % character followed by up to two hexadecimal digits represents a byte value in hexadecimal notation, thus %20 represents a space character, and %, %0, and %00 all represent an empty string value; %% represents the % character itself.

```
"P3AV matrix Stimuli= "
    "{caption icon audio} "                    // row labels
    "{focuson result stimulus1 stimulus2} "  // column labels

    // focuson
    "Please%20focus%20on icons\\focus%20on.bmp wav\\focus%20on.wav "
    // result
    "The%20result%20was icons\\result.bmp wav\\result.wav "
    // stimulus1
    "Donkey icons\\donkey.bmp wav\\snicker.wav "
    // stimulus2
    "% icons\\elefant.bmp wav\\trumpet.wav "
```

*Comments:* The stimulus properties might contain white spaces. A caption/icon/audio file are not being presented, if they are not defined (e.g., see caption in stimulus2). The stimulus definition parameter does *not* contain a description on how the stimuli are presented. Stimulus numbers start at 1. Relative paths are interpreted relative to the application module's executable location.[2]

## 1.2   Stimulus Sequence

Stimuli are presented in a certain sequence. This sequence can either be deterministic, i.e., defined by the investigator, or random.

### 1.2.1   Deterministic Sequence:

The investigator defines the order by entering a list of stimulus IDs to be presented. As an example:

```
1 5 3 4 2
```

defines a sequence in which stimulus 1 is first presented, followed by stimulus 5, etc.

---

[2]Which, in general, is not the application module's working directory at runtime but the path portion of the first argument to `main()`.

### 1.2.2   Random Sequence:

The investigator defines absolute stimulus frequencies for each stimulus, with the sum $N$ of those values equaling the total number of stimulus presentations in the final sequence. The resulting random sequence is obtained by applying a random permutation to an arbitrary sequence that reproduces the given frequencies, with all $N!$ index permutations being equally probable (Block Randomization). [3]

As an example:

```
6 2 3
```

defines a sequence of 11 stimulus presentations with stimulus 1 being presented 6 times, stimulus 2 2 times, and stimulus 3 3 times. The resulting sequence $S_1$ will be a permutation of $S_0 = [1, 1, 1, 1, 1, 1, 2, 2, 3, 3, 3]$, and the probability for $S_1$ to equal $S_0$ will be $\frac{6! \times 2! \times 3!}{(6+2+3)!} = 1/4620$.

Multiple sequences can be generated from the given frequencies. The investigator can define how many sequences are generated and played.

## 1.3   Stimulus Delivery

For any stimulus, delivery occurs simultaneously for caption[4], icon, and audio. (A computer can only execute commands in sequence, but the time difference between start of presentation of caption, icon, and audio, is negligible). *Comment:* A knowledgeable investigator has to understand the implications of audio files that are of unequal length !

An investigator can specify:

- Size and position of the target window (using the same scheme/parameters as used by the RJB task, Oddball paradigm, or P3 speller).

- Width and height of caption and icon in percent of screen width/height[5]

- Whether captions, icons, or audio files will be presented (i.e., a global switch – for each stimulus, captions/icons/wave files will not be presented individually, if not defined)

---

[3]Block randomization routines, as well as a pseudo random generator better than the one available via `rand()`, are available in the P3 Speller module.

[4]A caption, if defined, always appears in front on an icon.

[5]Using, e.g., the `TImage::Stretch` property for scaling.

- The volume for audio playback as a float value between 0 and 1

- Window background color in RGB[6]

- Caption color in RGB

- The duration during which a stimulus is presented (in units of Sample-Blocks)[7]

- The duration of an inter-stimulus interval following stimulus presentation (in units of SampleBlocks)[8]

- A minimum and maximum time (in units of SampleBlocks) that will be added randomly to the inter-stimulus interval, with probability distributed uniformly over time intervals. If these variables are both set to 0, the actual inter-stimulus interval will always be exactly as defined above. If these variables are set to, for example, 0 and 3, inter-stimulus intervals will randomly be longer by between 0 and 3 units, with a probability of 1/4 for the occurrence of any one of the four time intervals possible.

- A Comment. A user can enter comments to the specific run in a string parameter.

## 2  Processing of Classification Results

The task can be configured to interpret results communicated to it by the P3 Signal Processing module. These results represent a judgment on which of the stimuli was most likely selected. Handling of these results is identical to the P3 Spelling Task.

When it transmits a classification result, Signal Processing sets the state *StimulusCodeRes* to the stimulus code that was originally transmitted to it by the user application. For example, when it sets *StimulusCodeRes* to 3, it indicates that it transmits classification results for stimulus 3. In addition, it sets *StimulusTypeRes* to reflect the type of the stimulus (0=non-target,

---

[6]For convenience, RGB values may be entered in hexadecimal notation, e.g. `0xff0000` for red.

[7]Playback of audio extending above the specified duration will be muted.

[8]During the inter-stimulus interval, the screen is blank and audio is turned off.

1=target) when the system is in copy mode. Signal Processing transmits the classication result as one number (i.e., the first control signal).

## 2.1 Free Mode

The task can be configured to operate in free mode. In this case, the sequence of stimulus delivery is followed by a time period, in which the result of Signal Processing's classification is announced. The final classification result is the stimulus with the highest classification result.

In order to deliver this announcement, the system uses the stimulus defined in the `result` column of the stimuli parameter. This announcement is followed by delivery of the determined stimulus. In other words, after a sequence of stimulus delivery, the system might play a .wav file that says: "the result is," followed by a .wav file that says "yes." (assuming "yes" represents the stimulus that produced the highest classification result).

Finally, the task sends this result to the operator module as an ASCII text message so that it appears in a log window.

Free mode does not terminate until the investigator suspends operation.

## 2.2 Copy Mode

Copy mode is similar to free mode. In copy mode, the investigator can define a list of stimuli to be copied (e.g., "3 5 4"). In this example, the user has to attend to stimulus 3 for the first sequence, 5 for the second sequence, etc.

In addition to an announcement of the result, in copy mode the delivery of stimuli is preceded by an announcement that describes which stimulus the user has to attend to. This announcement uses the stimulus that is defined in the `focuson` column of the stimuli parameter. This announcement is followed by delivery of the desired target stimulus. As an example, the system might say "Please focus now on" ... "yes," before it starts with the sequence of stimulus delivery.

Copy mode terminates (i.e., the task suspends) when the user finished copying all stimuli specified by the investigator.

# 3 Parameters

Fill in the parameters incl. description, once completed.

| State Name | Bits | Description |
|---|---|---|
| SelectedStimulus | 7 | stimulus ID of finally selected stimulus |
| | | only >0 when determining selected stimulus |
| PhaseInSequence | 2 | 0 in inter-stimulus interval |
| | | 1 period prior to stimulus sequence (if any) |
| | | 2 stimulus sequence period |
| | | 3 period after stimulus sequence (if any) |
| StimulusTime | 16 | time in ms as calculated in P3 Speller |
| StimulusCode | 7 | stimulus ID of currently visible stimulus |
| | | or 0 if no stimulus visible |
| StimulusType | 1 | 0 in free mode; copy mode : |
| | | 1 when current stimulus equals target stimulus |
| | | 0 otherwise |
| Flashing | 1 | 1 during stimulus presentation, 0 otherwise |

Table 1: Encoding scheme for this task.

# 4 States

The time line of stimulus delivery is encoded in state variables as defined in Table 1.