P300 Tutorial

2003 Gerwin Schalk, Wadsworth Center, NYSDOH

Objective

This tutorial intends to give you a step-by-step guide on how to configure BCI2000 for the classification of P300 ERPs. Please see the User Manual for a complete reference. This tutorial applies to the P3SignalProcessing module in combination with the oddball task, the P300 speller, or the P300 Audio-Visual task.

Steps

Before you can configure the system, you need to run the system and analyze the signals by comparing the target responses from the non-target responses. The end result of this step is that you know what features (i.e., samples at particular channels at particular time points) provide the best linear discrimination.

As an example, let's assume that you would like the system to use the samples at channels 6 and 8, both at 300ms, for classification.

Next, you can configure the system as follows:

Sampling Rate:
240 (The sampling rate per channel. 240 should easily suffice for P300s.)

SampleBlockSize:
8 (In this example, the system will "pump" data through the system in blocks of 8 samples. This results in a system update of 240/6=40 times per second (or, 25 ms per update).

TransmitCh:
4 (or however many channels, as a subset of the total number of channels, you want to transmit to Signal Processing for P300 processing)

TransmitChList:
6 7 8 9. Provide a list of the channels that you would like to submit to Signal Processing. The number of items in the list has to match *TransmitCh*. In this example, channels 6-9 become channels 1-4 inside SignalProcessing.

SourceChOffset/SourceChGain
The first step in Signal Processing is calibration. Signals are converted from A/D units into μV. The used transformation is: result in μV=(A/D units-offset)*gain. Specify these variables as appropriate for your system.

SpatialFilteredChannels:
4 (BCI2000 can provide any linear spatial filtering operation. This number (e.g., 4) specifies how many output channels result from the spatial filtering operation).

SpatialFilterKernal

1 0 0 0
0 1 0 0
0 0 1 0
0 0 0 1

(using this unity matrix, the output of the SpatialFilter will equal whatever is transmitted into SignalProcessing (in this case, the channels 6-9 in μV). You obviously have to adjust matrix size if you use different # channels). See the User Manual for further details.

NumERPsToAverage
15 (This number specifies how many ERPs, for each stimulus, P3SignalProcessing averages before it classifies the waveform. Unless you are an expert, this number should match *NumberOfSequences*).

NumSamplesInERP
144 (The number of samples after stimulus elicitation that can be used for classification. 144 samples corresponds to 600ms at a 240Hz sampling rate.)

TargetERPChannel
1 - This is the channel (inside Signal Processing) whose average waveforms will be displayed in the ERP display. In this example, it will display input channel 6 (the first transmitted channel).

OffTime/OnTime
These parameters specify the duration (in units of SampleBlocks) that the stimuli (e.g., rows/columns in the P300 Speller) will be intensified. Using 1 and 4, respectively, corresponds in our example to a stimulus intensification of 100ms (4*25ms) and a non-intensified period of 25ms (1*25ms).

OnlineMode
0=copy mode (user has to copy TextToSpell), 1=free spelling mode

Finally, you need to tell the program which features to use. You do this by configuring MUD. You need to convert the times that you would like to use (e.g., 300ms) into samples using the current Sampling Rate. 240*300/1000=72.

MUD

1 72 1
3 72 1

This will classify signals using the output of waveform averaging. Make sure that you set *ClassMode* to 1. In this example, the program will calculate

sample at channel 1 in signal processing (i.e., channel 6 of the recorded channels) and sample 72 (i.e., 300 ms) times a weight of 1
+ sample at channel 3 in signal processing (i.e., channel 8 of the recorded channels) and sample 72 times a weight of 1

From this number, the system subtracts UD_A, and it will the multiply the result by UD_B. Make sure that *InterceptControl* is turned off so that the system doesn't adapt these values. You could use 0 for UD_A and 100 for UD_B.

The system will then transmit the classification results for each stimulus to User Application. In the case of the P3 Speller, it will then pick the highest classification result for all rows, and the highest result for all columns to determine the predicted character.

Warning: There is one caveat in that the system can only transmit these results to User Application (e.g., the P3 Speller) as integer values between -32768 and +32767. Thus, assuming that you calibrated the system to produce μV as described above, and used weights of around 1, you should choose UD_B at around 100 to bring the result into this range. If you used only a small UD_B, you lose much of the possible resolution.

Good luck with your P300 experiments!