Parameter Fragments

BCI2000 parameters can be loaded from or saved to files. These files have the form ****.prm*** and are loaded and saved from the configuration menu. When the system is started parameters have default values. Loading a parameter file will replace parameter values that are present in the parameter file. A parameter file does not need to contain all of the parameters used by BCI2000. Only the parameter values present in the parameter file are replaced. All other parameters remain the same as before loading the parameter file. Thus it is possible to load specific sets of parameters with parameter fragments. These parameter fragments change only a subset of parameter values.

Consider the parameter fragment ***dualmonitor.prm***. When BCI2000 is initially launched the user screen is positioned on monitor 1 according to default values for ***WinHeight, WinWidth, WinXpos and WinYpos***. The contents of ***dualmonitor.prm*** are shown below:

```
UsrTask int WinXpos=   1024 0 0 1 // User Window X location
UsrTask int WinYpos=      0 0 0 1 // User Window Y location
UsrTask int WinWidth= 1024 0 0 1 // User Window Width
UsrTask int WinHeight=  768 0 0 1 // User Window Height
```

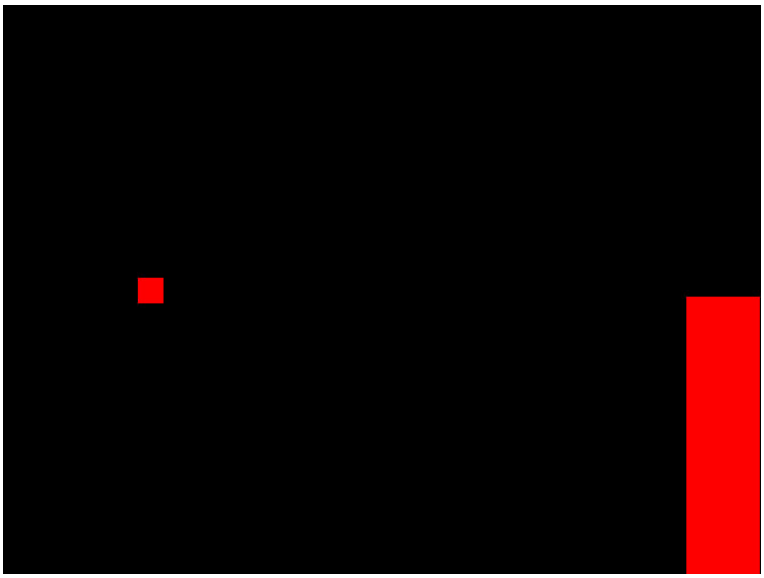
These provide new values that position the user screen on the second monitor of a dual monitor system (the format of parameter strings is discussed in). These values work for a dual monitir system with both monitors in 1024 x 768 pixels resolution.

Next consider the parameter fragment ***rjb2.prm***. If BCI2000 is launched with the D2Box.exe and ARSignalProcessing.exe modules, this fragment will configure the system to run a two-target task with vertical movement controlled by the user and horizontal movement at a constant rate. The contents of ***rjb2.prm*** are shown below:

```
UsrTask int FeedbackDuration= 20000 0 0 0 // Max Trial Duration
Targets int IncludeAllTargets= 1 0 0 1 // Test all target positions?
Targets int NumberTargets= 2 0 0 1 // Number of Targets
Targets float StartCursorX= 0.0 0 0 100.0 // Horizontal Start of Cursor
Targets float StartCursorY= 50.0 0 0 100.0 // Vertical Cursor Starting Position
Targets matrix TargetPos= { x%20start y%20start x%20length y%20height x%20code
y%20code adapt%20code } { target%201 target%202 } 90 90 0 51 10 10 50 50 0 0 -1 1 3 3 0 0 0
// Target Position Matrix - Values are 0-100
Statistics matrix BaselineCfg= 2 4 TargetCode 1 Feedback 1 TargetCode 2 Feedback 1 0 0 0 //
states to watch for baseline
Statistics matrix BaselineHits= 2 2 1 0.50 2 0.50 0 0 0 // proportion correct for each target
Statistics float XGain= -40 5.0 0.0 100.0 // Normal Filter Left/Right Slope
Statistics float XMean= 10 -5.0 -100.0 100.0 // Normal Filter Left/Right Intercept
Statistics float XPixelsPerSec= 630 70 0 400 // Horizontal Pixel Rate
Statistics int XTrendControl= 0 0 0 2 // X Intercept Adapt 0 none 1 mean 2 mean prop 3 slope
Statistics float YGain= 1000 5.0 -100.0 100.0 // Normal Filter Up / Down Slope
Statistics float YPixelsPerSec= 472 70 0 400 // Desired pixels per second
```

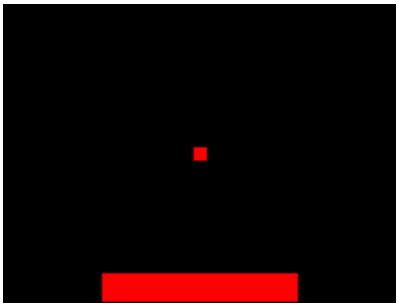
Statistics int YTrendControl= 1 0 0 2 // Y Intercept Adapt 0 none 1 mean 2 mean prop 3 slope

This fragment contains parameter values that control the task appearance and parameter values that control on-line adaptation. When this fragment is used the user screen appears as shown below:

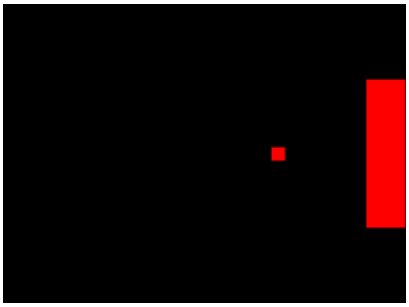


This task is described in McFarland, D. J., Sarnacki, W. A., & Wolpaw, J. R. (2003) Brain-computer interface (BCI) operation: optimizing information transfer rates. Biological Psychology, 63, 237-251. The fragment contains information about the task appearance. However information concerning the EEG features that control cursor movement are not provided by this fragment.

Two alternative 1-dimensional tasks can be configured by the fragments ***ud.prm*** and ***lr.prm***. The screen appearance for these are shown below:



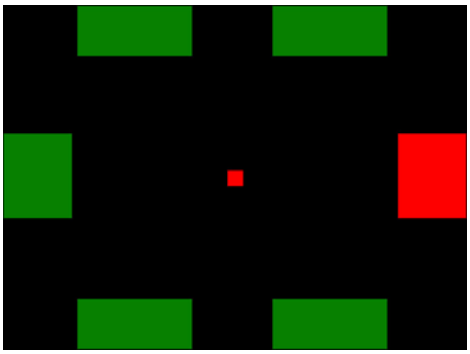
UD.prm



LR.prm

Both of these tasks are two-target, user-paced, 1-dimensional tasks. The difference is in the direction of cursor movement and the positioning of the targets.

A 2-dimensional task can be configured by the fragment ***Six2dtargets.prm.*** The screen appearance for this is shown below:



For this display, the parameter ShowAllTargets in the Targets menu has been set to 1. This task allows for cursor movement in two dimensions. UD.prm and LR.prm can be used for preliminary 1-dimensional training prior to starting 2-dimensional training.

These and other parameter fragments are listed below:

1-D fragments
- rjb2.prm
- ud.prm
- lr.prm

2-D fragments
- 2d4target.prm
- 4small2dtargets.prm
- 6small2dtargets.prm
- Eight2dtargets.prm
- Six2dtargets.prm
- Sixteen Targets.prm
- twenty 2D atrgets.prm

stdlib
- dualmonitor.prm

AVScreening
- HandsFeet.prm
- LeftRight.prm

AVScreening provides parameters for using the P3AV.exe module for mu screening.