Writing Glimmer Scripts

What are Glimmer scripts for?

Glimmer scripts are Javascript programs used to control Glimmer's output based on signals from an EEG device. Some examples of the things that you can do with Glimmer scripts:

- You can use Glimmer scripts to tailor stimulation to an individual person (for instance, rather than picking a frequency somewhere in the alpha band, you can find a person's *individual* alpha band and deliver flicker at that frequency).
- You can design *neurofeedback* protocols to train specific brain states (i.e. the flicker amplitude decreases when alpha power increases)
- You can measure and record how a particular flicker pattern affects brain activity.

Anatomy of a Glimmer script

Basic concepts

The code in a Glimmer script gets run every time new data is available from the EEG (the time interval varies depending on how many seconds of data are requested). Glimmer scripts can access two frequency spectra: the "baseline" spectrum (representing an activity baseline acquired over a long period of time) and the "sample" spectrum (a shorter snapshot of activity acquired recently). Each spectrum covers a range from 0.1 to 100 Hz with a frequency resolution of 0.1 Hz.

Each time a Glimmer script is run, it performs computations on the baseline and sample spectra using built-in functions, and then sets Glimmer's flicker amplitude and frequency by setting the corresponding variables which are read by Glimmer at the end of each run.

Glimmer comes with an example script called boost_alpha which delivers stimulation at a person's peak alpha frequency. We'll use this as a simple example of Glimmer script syntax.

Setup parameters
The first thing we need to do is tell Glimmer what kind of data the script expects. This is done using the first three lines of the script (while these lines are not processed by the Javascript engine, they are are read when the script is first loaded):

```
//sampleLength=3
//baselineLength=10
//baselineMode=continuous
```

The first parameter, sampleLength, is a whole number representing how long each EEG sample should be in seconds. A Glimmer script only gets run when a new sample is available, so a script with a longer sample length will get called less frequently than one with a shorter sample length.

Frequency spectra have a speed/accuracy tradeoff; a short sample length will let your script "update" faster than a long sample length, but the resulting spectrum will also be noiser and have more spectral leakage.

The second parameter, baselineLength describes how many seconds of data are used to create the baseline frequency spectrum. It is usually longer than the sample length.

Finally, baselineMode can be set to either "continuous" or "startup". In continuous mode, the baseline spectrum is a moving average of the last baselineLength seconds of EEG activity, in startup mode the baseline is acquired when the script starts up and never updated.

Data processing
The next part of the script is actual javascript code that runs when a new sample is processed.

```
var peakFreq=maxFreq(lastSample,7.5,13);
```

Here, lastSample is an automatically generated variable containing the frequency spectrum of the last sample (you could also replace lastSample with "baseline" to analyze the baseline spectrum instead). This data gets fed into a builtin function, maxFreq, which returns the frequency with the highest power in the range 7.5 Hz to 13 hz.

There are a number of other builtin functions that you can use for analyzing data:

meanPower(spec, startFreq, endFreq) returns the average power of frequency spectrum spec between startFreq and endFreq

centerOfGravity(spec, startFreq, endFreq) returns a number between 0 and 1 reflecting how power in spec is distributed in the range startFreq to endFreq. A value of 0 indicates all power is concentrated at the low end of the range, a value of 1 indicates all power at the high end, and a value of 0.5 indicates that power is equally distributed across the range (the spectrum is "flat"). This function can be more reliable in detecting shifts in EEG frequency than the maxFreq and minFreq functions because it looks at the entire distribution of power, not just the peaks.

minFreq(spec, startFreq, endFreq) returns the frequency between startFreq and endFreq which has the lowest power in spec.

maxFreq(spec, startFreq, endFreq) returns the frequency between startFreq and endFreq which has the highest power in spec.

Outputs
Every Glimmer script must create two variables, frequency and amplitude:

```
var amplitude=1;
var frequency=peakFreq;
```

These variables are read at the end of execution to set the Glimmer flicker amplitude (from 0 to 1) and frequency (in Hz).

Optionally, scripts can log data by setting the logData variable:

logData=maxFreq(lastSample,7.5,13);

Logged data will be written to a log file with the time and date of the script's start.

Putting it together
The following script will generate flicker at a person's alpha frequency, with a constant amplitude:

```
//sampleLength=3
//baselineLength=10
//baselineMode=continuous
var peakFreq=maxFreq(lastSample,7.5,13);
var amplitude=1;
var frequency=peakFreq;
```