

$$\text{ch_1} \in R^{\text{number of samples}}$$

ch_2

ch_3

ch_4

ch_5

上図のような行列を作成. トリガの配列も同様に作成. 低レイテンシで刺激の提示が可能となる.
提示する系列はcsvファイルから読み出す.

CSVファイルを作成する時点で, 提示系列が確定することに注意してください.

重要

この手法だと, メモリを非常に食います. 例えば, **8ch**で**10**分間提示したとします.

データ行列のサイズは, $8 * 44100(\text{fs}) * 10 * 60 * 4\text{byte}(\text{float}) = 800\text{MB}$ になります.

30分なら2.4GBです.

Windowsにおいては, **32bit**アプリケーションが利用できるメモリは**2GB**までと決まっているので, **64bit**で動かします. **Visual Studio**の場合は, 上の方で**x86**と**x64**を切り替えられると思うので, **x64**で使いましょう. また, その場合は**portaudio**も**NIDAQmx**も**64bit**用でないといけないので注意してください.

・csvファイルのフォーマット

改行区切りのcsvファイルを作成する. matlabの場合は, 列ベクトルを作成してwritematrix関数を実行することで保存可能. (stim_generator.mを参照)

実際のcsvファイルの例.

```
2
0.8
0
0
1
1.6
0
```

0
1

csv[0] : チャンネル数
実際のデータはcsv[1]以降で指定
4つのパラメータを順に指定
1 : 時間 (秒)
2 : 提示するチャンネル (インデックスは0で開始)
3 : 刺激の種類 (インデックスは0で開始)
4 : トリガ番号
ステレオデータとかを再生したい場合は,
0.8 //時間
0 //左チャンネル
0 //左チャンネルのデータ
1 //trig
0.8 //時間
1 //右チャンネル
1 //右チャンネルのデータ
1 //trig
とすれば, ステレオデータを再生できる.

・Cソースの設定

siAuditory内に記述されているsiSequencerクラスを用いて, トリガ配列, データ行列を作成する.
以下のように使用する.

```
siSequencer seq;  
seq.load_csv("filename.csv", int Fs); // filename.csvを読み込. fsも指定  
seq.gen_array(tones, duration);  
tones : 刺激音データの2重配列 (float** tones)  
duration : tones[i]のデータ長が格納された配列 (float* duration)
```

また, seq.load_csv(float* data, int N, int Fs)も利用可能. Nはdataの長さ.
この場合は, float型の配列にcsvファイルに記載するのと同じ内容を入れておけば利用できるので, コンパイルするたびに提示系列がランダムになるようなプログラムも作成できる.

drwav等を使用すれば, wavファイルから刺激音を取り込むことも可能.
#define ENTRIG : 1 -> トリガ有効, 0 -> トリガ無効
#define Fs : サンプルレート指定
#define FRAMES_PER_BUFFER : 1回あたりDACに送信するサンプル数を指定

FRAMES_PER_BUFFERについて...

Cソース内のdsp関数が実行された際に, DAC(オーディオIF)に送信されるサンプル数を指定します. 少なければレイテンシが減少し, 多ければレイテンシは増大します. しかし, 少なくすると, 音飛びとかが起きる可能性があります. 仮に1に設定したとします. dsp関数が実行されて, 1サンプル分のデータがDACに送られます. DACは1/fs秒間そのデータを出力します. その1/fs秒の間に次のサンプルが送信されてくれば問題ないですが, なんらかの理由でそれが遅れると音飛び

等不具合を起こします。現状は48に設定しています。一般的には512とかでもリアルタイムと言われます。

・ターミナルに出力される警告

Warningがターミナルに表示されることがあります。その内容と原因を記載します。

Warning : This session is NOT running on ASIO.

ASIOドライバで動いていない場合です。portaudio_xxx.libが、非ASIO用の可能性があります。レイテンシが増大し、トリガとの同期が取れなくなります。

刺激のテスト等をするさいに非ASIOの(排他的でない)portaudioを使うことがあるかと思いますが、そのまま忘れて実験でしようしないように、警告がでるようにしました。

Warning : amplitude exceeds 1, max amplitude : xxx

提示する系列によっては、複数の音刺激が同時に再生されるような場合もあると思います。その際に、振幅の最大値が1を超えると表示されます。1を超えると波形が歪むので、この警告が出た場合は、音刺激の振幅を小さくするなどの対策が求められます。

・考えられるエラー

bad_alloc系のエラーが出た場合、32bitで動かしていないか確認してください。bad_allocはメモリ領域が確保できなかった際に発生します。

・トリガ遅延計測結果

8ch使用, 15分間, 1800回分の音刺激を計測した結果。

mean : 0.52ms

std : 0.31ms

min : -0.05ms

max : 1.09ms