**MANUAL**

**ESTIMACIÓN DE CAMPOS RECEPTIVOS DESDE REGISTROS MEA**

**ALAND ASTUDILLO 2014**

**OBJETIVO**

Análisis de señales provenientes de registro de células ganglionares en retina utilizando MEA. Obtención de *spikes* por neurona, obtención de señal de sincronía con el estímulo, análisis de campos receptivos, curvas temporales y perfiles espacio - temporales y ajuste gaussiano del campo receptivo. Caracterización de las células encontradas.

**INSTRUCCIONES**

El resultado del registro MEA corresponde a un conjunto de archivos .mcd que contienen el registro para todos los canales de cada experimento. Es necesario tener claro el experimento realizado a través de la lectura de la bitácora (digital o foto) generada en el experimento para poder determinar que archivo analizar en el caso de la estimación de campos receptivos utilizando estímulos tipo checker board.

1. OBTENER SEÑAL DE SINCRONÍA

Esta etapa se realiza en *Matlab*.

Desde el archivo mcd que se va a analizar se debe obtener la señal de sincronía, para ello se debe usar el script de *Matlab*

**get\_synchrony\_signal.m**

Copiar el archivo m en la carpeta en donde se encuentra el archivo mcd. Los parámetros que se deben definir y cambiar en ese script son

**pathname**

**filename**

**filename2**

En **pathname** debe ir la ruta completa del lugar donde se encuentra el archivo mcd, por ejemplo

**pathname** = 'C:\Users\ALIEN3\Desktop\resultados\_26-12\';

En **filename** debe ir el nombre del archivo mcd del cual se desea extraer la señal de sincronía, por ejemplo

**filename** = 'datos0005.mcd';

En **filename2** va el nombre del archivo sin la extensión, por ejemplo

**filename2** = 'datos0005';

Previo a la ejecución del script, se debe añadir al directorio de *Matlab* la carpeta "**ns**", la cual contiene las funciones de *Neuroshare* para permitir leer archivos mcd desde *Matlab*. También se debe asegurar de tener en esa carpeta el archivo \*.dll específico (librería de *Neuroshare*) para el sistema operativo utilizado (caso de *Windows*) o el archivo so en el caso de *Unix*.

Luego de añadir al *path* la carpeta **ns** y modificar los campos mencionados, se puede ejecutar el script con F5 o con el boton *Run.* La salida del programa corresponde a un archivo \*.mat con la señal de sincronía y una imagen de la señal en formato **pdf**.

El nombre del archivo de salida es

**syn\_signal1\_xxxxx.mat**,

donde la última porción del nombre **xxxxx** corresponde al mismo nombre definido como el parámetro **filename2**.

1. ANALIZAR SEÑAL DE SINCRONÍA

Esta etapa se realiza en *Matlab*.

La señal obtenida en la etapa anterior es analizada en búsqueda de las posiciones exactas de cada frame en el tiempo. Se utiliza el archivo \*.mat generado como **syn\_signal1\_xxxx.mat**, donde **xxxx** corresponde al nombre del archivo mcd analizad, sin la extensión. El script para realizar esto se llama

**sync\_signal\_analizer3.m**

Se debe copiar el archivo \*.m en la carpeta en donde se encuentra el archivo \*.mcd. Los parámetros que se deben definir y cambiar en este script son

**datosname**

En **datosname** debe ir el nombre del archivo \*.mcd (sin la extensión), por ejemplo

**datosname** = 'datos0005';

Ejecutar el script usando F5 o el botón Run. Cuando se ejecute el script se mostrarán en pantalla los valores de promedio y desviación estándar de las distancias entre frames, en puntos. En este caso para 20khz, e imágenes a 30Hz (frame rate), la distancia promedio entre frames es de 334 puntos y una desviación estándar de 2 puntos. Si esto no se cumple, entonces hay un error en la detección de los frames en la señal de sincronización, y el análisis no se podrá hacer.

La salida corresponde a un archivo llamado

**inicio\_fin\_frame\_xxxx.txt**,

en donde **xxxx** corresponde al nombre del archivo \*.mcd analizado, determinado por el parámetro **datosname**, y contiene el tiempo de inicio y fin de cada frame, en puntos.

1. ANALIZAR ARCHIVO MCD: SPIKE SORTING

Esta etapa se realiza utilizando *Offline Sorter* de *Plexon*, sólo en sistema operativo *Windows*.

La entrada corresponde al archivo \*.mcd completo. Se recomienda que una vez cargado el archivo \*.mcd, este sea guardado y transformado a formato \*.plx, lo que permite trabajar de forma más rápida en la lectura de canales y la realización de todos los análisis de *spike* *sorting*.

El análisis corresponde al filtrado de la señal (pasa alto 100Hz *Butterworth*) en cada canal del multi electrodo (252 electrodos), la detección de los *spikes* (umbral inferior para detección de los *spikes* en su parte negativa) utilizando como medida umbral un factor de la desviación estándar del ruido estimado o de la señal. Los *spikes* encontrados deben ser alineados o centrados y luego se realiza una extracción de características (PCA) para finalizar con una etapa de *clustering* (manual o automática). El resultado corresponde a clústeres, cada clúster corresponde a una unidad celular potencial.

Una vez terminado el análisis, se guarda un archivo \*.plx, que contiene solo los *spikes* y tiempos de *spikes* de las unidades celulares encontradas.

El archivo \*.plx es nombrado como

**xxxx\_ts\_fecha.plx**,

en donde **xxxx** corresponde al nombre del archivo mcd original, y **fecha** corresponde a la fecha del análisis, en formato **dd-mm-aaaa**.

1. GUARDAR TIEMPOS DE SPIKES

Utilizando el programa *NeuroExplorer (Windows)*, abrir el archivo \*.plx guardado con los *spikes* y tiempos de *spikes* de las unidades obtenidas. Ir a la pestaña de *timestamps* y copiar toda la tabla de datos que aparece. Crear un archivo de texto nuevo (*Block de notas*, *Notepad++)* pegar y guardar con extensión \*.txt.

Abrir el script readwrite\_timestamps\_txt.py con un editor. Editar los parámetros

**tsfolder**

**filetimestamps**

**tsfolder** corresponde a la carpeta en donde se guardaran los tiempos de *spike* de cada unidad. Cada unidad tendrá sus tiempos de spikes en archivos txt aislados, por ejemplo

**tsfolder** = “ TS\_datos0003”

**filetimestamps** corresponde al nombre del archivo de texto en donde se guardaron los tiempos de *spikes* desde el archivo \*.plx en el paso anterior.

**filetimestamps** = “xxxx\_ts\_fecha.plx”

donde **xxxx** y **fecha** dependen del archivo real revisado. Guardar los cambios y ejecutar desde command windows

**python readwrite\_timestamps\_txt.py**

Se creará la carpeta respectiva (según el nombre definido por el parámetro **tsfolder**) y dentro se crearán los archivos \*.txt para cada unidad o célula hipotética.

1. SPIKE SORTING AVERAGE (STA)

Para realizar el STA se pueden utilizar dos opciones. La primera opción es usar un código para realizar el STA de una unidad específica. La segunda forma es utilizar otro código para realizar el STA a un conjunto de unidades consecutivas.

Para utilizar la primera forma, se utiliza el script

**sta\_v4\_sspy\_xxxx.py** ,

en donde **xxxx** debe ser reemplazado por el nombre del archivo \*.mcd analizado, sin la extensión \*.mcd. Copiar el script **sta\_v4\_sspy\_xxxx.py** en la carpeta en donde se encuentre la carpeta **tsfolder**, y el archivo **inicio\_fin\_frame\_xxxx.txt**. Además, es necesario conocer el lugar en donde se encuentra la carpeta contenedora de las imágenes (estímulos, imágenes \*.png o \*.jpg guardadas según índice). En este ejemplo la carpeta contenedora de las imágenes es denominada **checkImages**.

La primera vez que se realice el STA, se pueden hacer un par de pruebas de lectura de archivos (opcionalmente). Además, se debe obtener la imagen promedio (**mean\_image.mat**) y generar un archivo que contiene los nombres de las direcciones de todas las imágenes (**image\_filenames.mat** y **image\_filenames.txt**).

Abrir el archivo script y editar los siguientes parámetros

**stafolder** : corresponde a la carpeta que contendrá los resultados del STA, por ejemplo

**stafolder** = 'STA\_datos0003'

**imageruta** : corresponde a la ruta o *path* en donde se encuentra la carpeta que contiene las imágenes o estímulos, por ejemplo

**imageruta** = 'D:/'

**imagefolder** : corresponde al nombre de la carpeta contenedora de imágenes, por ejemplo

**imagefolder** = 'checkImages'

**imagefiltro** : corresponde al tipo de imágenes que serán cargadas, por ejemplo

**imagefiltro** = '\*.png'

**timefolder** : corresponde al nombre de la carpeta que contiene las unidades y sus respectivos *timestamps*, por ejemplo

**timefolder** = 'TS\_datos0003/'

**samplingRate** : corresponde a la tasa de muestreo en Hz de los datos adquiridos originalmente, por ejemplo

**samplingRate** = 20000

**numberframes** : corresponde al número de frames antes del *spike* que serán utilizados para obtener el STA, por ejemplo

**numberframes** = 13

**numberframespost** : corresponde al número de frames después del *spike* que serán utilizados para calcular el STA, por ejemplo

**numberframespost** = 5

**synchronyfile** : corresponde al nombre del archivo de texto que contiene los índices de tiempos de los frames, y que es el resultado del análisis de la señal de sincronía, visto en una de las etapas previas de este manual, por ejemplo

**synchronyfile** = 'inicio\_fin\_frame\_datos0003.txt'

**sizex** : corresponde al tamaño en el eje x de la imagen a cargar, en pixeles, por ejemplo **sizex** = 380

**sizey** : corresponde al tamaño en el eje y de la imagen a cargar, en pixeles, por ejemplo **sizey** = 380

**dolog** : corresponde a la opción de guardar los resultados del STA escalados usando logaritmo , por ejemplo **dolog** = 0

Una vez editados estos parámetros, guardar los cambios y ejecutar desde command windows. El script necesita 5 parámetros de entrada:

**getimagenames** : se utiliza la primera vez como parámetro activo para generar el archivo con la lista de los nombres y *path* de las imágenes

**openimagesandwrite** : corresponde a un test para leer y escribir imágenes usando el *package* respectivo (opcional)

**calculatemeanrf** : se utiliza la primera vez como parámetro activo para calcular el estímulo promedio desde todas las imágenes.

**tipoalgoritmo** : corresponde al tipo de algoritmo a utilizar (cargar conjunto de imágenes completo o realizar una carga secuencial). El tipo de algoritmo utilizado actualmente corresponde al 2, ya que el primero está obsoleto momentáneamente.

**timestampName** : corresponde al nombre de la unidad a analizar, por ejemplo si la unidad a analizar corresponde a "C12a", se ingresa esa palabra.

Para ejecutar el código por primera vez para analizar la unidad C12a se escribe en command windows

**>> python sta\_v4\_sspy\_xxxx.py 1 1 1 2 C12a**

Luego de que se cumplan todas las etapas, se creará una carpeta denominada C12a dentro de la carpeta de STA (según el nombre definido como parámetro) que contendrá un número de frames y el promedio de esos frames. Además, la matriz STA y otros parámetros estarán guardados como archivos \*.mat. En el mejor de los casos el campo receptivo podrá ser visto en los diferentes frames.

Para ejecutar el código, luego para cualquier otra unidad, se puede escribir

**>> python sta\_v4\_sspy\_xxxx.py 0 0 0 2 G8a.**

Los parámetros iniciales ya no son necesarios, ya que tanto el archivo de imagen promedio y la lista de imágenes ya fueron creados.

Cada vez que se realice el análisis en un computador se debe crear una sola vez estos archivos. Si para varios análisis de STA de diferentes fechas las imágenes son las mismas utilizadas (es decir que la semilla para generar el checkerboard es la misma) es posible saltarse el paso de obtener la imagen promedio y obtener la ruta de todas las imágenes. Sin embargo, se debe considerar siempre que para distintas fechas de análisis (un archivo de uno u otro experimento) tendrán señales de sincronía distintas. Por lo tanto el análisis de señal de sincronía no es opcional.

Los archivos de salida que genera el script STA estarán contenidos en la carpeta STA definida por parámetro. Dentro de la carpeta estarán las carpetas de unidades, cuyos nombres son los respectivos a cada unidad analizada. Dentro de cada carpeta para cada unidad se encontrará la colección de frames en formato png correspondientes a la estimación del campo receptivo en una representación espacio temporal, el STA promedio, y archivos \*.mat que contienen la matriz STA escalada para visualización y otro archivo \*.mat conteniendo cada timestamp e índice de frame respectivo (para verificación). Los archivos se denominan **stavisual\_lin\_array\_xxxx.mat** y **spikeframe\_matrixxxxx.mat**, respectivamente, donde **xxxx** corresponde al nombre de la unidad analizada.

Nota: el tiempo de cálculo de un STA dependerá, entre otros, de la cantidad de spikes que tiene la unidad analizada. En promedio, un análisis de este tipo para una unidad que tiene 1000 spikes, puede tomar entre 10 a 15 minutos.

Para ejecutar el análisis STA de un bloque o conjunto de unidades se debe usar el script

**sta\_chain2\_xxxx.py**

(en conjunto con el script **sta\_functions2.py**). Donde **xxxx** es reemplazado por el nombre del archivo \*.mcd analizado.

Copiar y pegar el script **sta\_chain2\_xxxx.py** en la misma carpeta en donde se encuentra la carpeta que contiene los timestamps, en este ejemplo la carpeta se llama TS\_datos0003, y el script queda como sta\_chain2\_datos0003.py.

Editar el script y modificar los siguientes parámetros

**archivosruta** : corresponde a la ruta completa de donde se encuentra la carpeta que contiene los timestamps aislados en archivos \*.txt, por ejemplo

**archivosruta** = 'D:/Experimentos\_CINV/datos\_20-11-2013\_bga50um\_2/'

**archivosfolder** : corresponde al nombre de la carpeta que contiene los timestamps de las unidades a analizar, por ejemplo

**archivosfolder** = 'TS\_datos0003/'

**archivofiltro** : corresponde a la extensión de los archivos que contienen los tiempos de spike (timestamps) de las unidades, y que funciona como filtro para cargar los archivos, en este caso se utiliza por defecto

**archivofiltro** = '\*.txt'

Los parámetros **getimagenames**, **openimagesandwrite**, **calculatemeanrf** y **tipoalgoritmo** corresponden a los mismos definidos en el caso anterior y si no es primera vez que se ejecuta el código se pueden dejar como

**getimagenames** = 0

**openimagesandwrite** = 0

**calculatemeanrf** = 0

**tipoalgoritmo** = 2

Los siguientes parámetros son similares a los definidos en el caso anterior, y se pueden dejar, por ejemplo

**stafolder** = 'STA\_datos0003\_2'

**imageruta** = 'D:/'

**imagefolder** = 'checkImages'

**imagefiltro** = '\*.png'

**samplingRate** = 20000

**numberframes** = 18

**numberframespost** = 2

**synchronyfile** = 'inicio\_fin\_frame\_datos0003.txt'

**sizex** = 380

**sizey** = 380

**dolog** = 0

Los últimos parámetros que se editan son **inicio** y **final**, con los cuales se define el intervalo de índices de las unidades que serán analizadas. Para saber el índice de una unidad, se debe tener en paralelo una tabla Excel en donde se ordenan las unidades con sus nombres e índices. Estos parámetros pueden quedar, por ejemplo

**inicio** = 10 -1

**final** = inicio + 5

lo cual significa que se analizará desde la unidad 10 hasta la unidad 10+5, es decir 5 unidades, cuyos nombres se pueden ver en la tabla según su índice. Al guardar los cambios y ejecutar, se obtendrá la creación de las carpetas respectivas y el análisis de cada unidad.

Nota: el script **sta\_chain2\_xxxx.py** utiliza al script **sta\_functions2.py**. En este script se ha modificado para permitir guardar la matriz STA real y la matriz STA re escalada para visualización. Esto no sucede así en el caso del script que analiza las unidades de manera individual, pero para la siguiente versión si estará incluido para ambos casos.

1. CURVAS DE PERFILES TEMPORALES

Una vez que se tiene la salida del STA para cada unidad, se utiliza el archivo \*.mat de cada unidad para obtener la curva de perfil temporal, y a su vez la matriz de curvas de perfiles temporales para todas las unidades de interés.

(…)

1. AJUSTE GAUSSIANO DE LOS CAMPOS RECEPTIVOS

Para cada unidad analizada con STA, y que se ha encontrado por inspección que corresponde a un campo receptivo, se puede realizar el ajuste gaussiano. Para ello se utiliza el script **plot\_sta.m**, cuyos parámetros de configuración son

**nombre\_cell\_grupo** : corresponde al nombre de la unidad a analizar para el ajuste gaussiano, por ejemplo

**nombre\_cell\_grupo** = 'A2a';

**carpeta** : corresponde al nombre de la carpeta que contiene los resultados del STA, y se debe reemplazar sólo la primera parte del nombre, por ejemplo

**carpeta** = ['STA\_datos0003\_2/',nombre\_cell\_grupo,'\_lineal/'];

**spiketimestamps\_file** : corresponde a la carpeta que contiene los timestamps de cada unidad, y se debe reemplazar sólo la primera parte del nombre, por ejemplo

**spiketimestamps\_file** = ['TS\_datos0003\_2/', nombre\_cell\_grupo ,'.txt'];

La salida de este script corresponde a varias imágenes \*.pdf (4 archivos) que contienen los resultados del ajuste para el frame. El frame elegido es aquel que contiene el valor extremo máximo (o mínimo) de todos los frames que componen la matriz STA. Además, se guardan los datos del ajuste (parámetros, vectores y valores) en un archivo \*.mat y opcionalmente una hoja de Excel.