

01170

**UNIVERSIDAD NACIONAL AUTONOMA DE
MEXICO**

**DIVISION DE ESTUDIOS DE POSGRADO
DE LA FACULTAD DE INGENIERIA**



Sintetizador de señales neurofisiológicas extracelulares

T E S I S
que para optar por el grado de
Maestro en Ingeniería Eléctrica
presenta
Roberto Serna Herrera

Director de tesis : Dr. Ismael Espinosa Espinosa

1997

**TESIS CON
FALDA DE ORIGEN**



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

TESIS CON FALLA DE ORIGEN

*Por todo lo que significan para mi
a mi esposa e hijos :
Laurita, Lalo, Paco y Goyo*

*A mis padres
Elodia, Antonio
A mis hermanos
Toni, Moni, Lety y Martita
por todo su apoyo y cariño*

*A mis suegros
Emma y Goyo
cuñados
Lala y Carlos*

*A Laurita por todo su amor
con todo mi amor*

Agradecimientos :

Agradezco de manera muy especial al Dr. Ismael Espinosa Espinosa, coordinador del Laboratorio de Cibernética de la Facultad de Ciencias todo su apoyo y toda la paciencia que tuvo a lo largo de este trabajo para asesorarme resolviendo mis dudas.

Agradezco a los sinodales de esta tesis el tiempo dedicado a su revisión

Dr. Francisco J. García Ugalde

Dr. Ismael Espinosa Espinosa

Dr. Rogelio Alcántara Silva

Dr. Jesús Savage Carmona

Dr. Boris Escalante Ramírez

Agradezco también al Fis. Cristóbal Cárdenas Oviedo y la Mat. Graciela Rojas González y en general a la Universidad Iberoamericana el verme dado la oportunidad para concluir el presente trabajo.

De la misma manera agradezco a mis amigos del departamento de Matemáticas de la Ibero que siempre me han alentado : Cristóbal, Graciela, Alfredo, Humberto, Arely, Bertha, Julio, Guillermo, Gerardo, Toño, Edmundo, Juan y Maricarmen.

También agradezco a todos mis amigos del Laboratorio de Cibernética y a todos quienes en algún momento pasaron por ahí : Rul, Alberto, Rubén³, Irma, Javier, Juan Manuel, César, Gregorio, Omar, Jorge, Toño, Chucho y Tere.

Este trabajo fue parcialmente apoyado por la DGAPA de la UNAM con los proyectos : IN206189 y IN100593 a cargo del Dr. Ismael Espinosa Espinosa.

Indice

INDICE	1
INTRODUCCIÓN	3
ANTECEDENTES	5
CAPÍTULO 1	9
La Neurona.	9
Introducción.	9
Breve descripción de la Fisiología de una neurona.	9
La unión sináptica.	11
El potencial de acción.	12
El registro de potenciales de acción.	14
Técnicas de separación de potenciales de acción.	14
Introducción.	14
Componentes Principales.	15
Algunas técnicas en hardware.	18
Discriminadores.	18
Algunos paquetes comerciales para la separación de potenciales de acción.	19
Discovery.	19
MSD (Multi Spike Detector) Detector Multi-espigas.	20
Waveform Discriminator (SPS-8701) Discriminador de forma de onda.	20
Multichannel Neuronal Acquisition Processor (NMAP). Procesador multicanal de adquisición neuronal.	20
CAPÍTULO 2	22
El Simulador GESSIN.	23
Introducción.	23
Instalación de GESSIN.	24
Módulo de generación de la traza, UNE-VEC.	25
Módulo de conversión D/A, DAC-ESP.	28
Módulo de Correlación, XCOR.	29
Descripción del algoritmo de XCOR.	30
CAPÍTULO 3	35
Resultados	35
Planteamiento de las simulaciones.	35
Ejemplo de excitación.	37
Ejemplo de inhibición.	42

Ejemplo de independencia.	45
Ejemplo de oscilación.	48
Ejemplo de un circuito con 5 neuronas.	51
CAPÍTULO 4	61
Discusión de resultados y perspectivas.	61
APÉNDICE A	65
Listado del archivo de constantes comunes.	65
Listado del programa UNE-VEC	66
Listado del programa DAC-ESP	75
Listado del programa XCOR	79
APÉNDICE B	85
Especificaciones tarjeta A/D D/A.	85
REFERENCIAS	87
INDICE DE FIGURAS	89
INDICE ALFABÉTICO	93

Introducción

El estudio del cerebro es uno de los problemas más abordados en la actualidad, desde diferentes disciplinas y con diferentes técnicas, se trata de ahondar en la forma en la que un ser vivo con cerebro es capaz de realizar tareas tan complejas como el simple movimiento. Miles de conexiones entre neuronas se llevan a efecto cada segundo y a través de cada una de ellas viajan impulsos eléctricos que llevan información muy compleja [20][21]. Uno de los mayores problemas es el determinar qué neuronas, en un circuito (red neuronal biológica) se están comunicando y de que manera lo están haciendo. El uso de los microelectrodos de tungsteno ha permitido obtener registros de este tipo de señales, con lo cual se ha avanzado en el estudio de la conectividad neuronal pero siguen existiendo algunos problemas por resolver en este campo. Algunos de ellos son, por ejemplo, el separar, almacenar y procesar la información, porque aunque existen en la actualidad muchas formas de hacerlo, no hay una que sea completamente infalible y que sirva para cualquier tipo de registro.

Los microelectrodos de tungsteno permiten realizar registros electrofisiológicos extracelulares debido a sus características físicas como impedancia y superficie. Estos registros contienen tanto las contribuciones de la actividad eléctrica de varias neuronas como el ruido nervioso. Este tipo de registro es muy rico en información, pero presenta el problema de que necesita de métodos especiales para separar las actividades eléctricas individuales (trenes de espigas o de potenciales de acción) de cada neurona en el registro. El problema no es trivial y se le han dado muchas soluciones. En la época en que se utilizaba un microelectrodo únicamente, el problema se solucionaba parcialmente por medio de umbrales y ventanas electrónicas, aunque usualmente se separaba nada más la espiga más grande. En estudios en los que interesan todas las espigas (como en los estudios de conectividad funcional que nosotros hacemos) y no solamente la más grande, el método anterior tiende a volverse muy tedioso y, si en lugar de un microelectrodo, se utilizan varios (hasta 10), como se acostumbra en la actualidad, entonces el problema es peor [2][3]. En estos días existen tres o cuatro paquetes comerciales para realizar la separación por medios computacionales, pero distan mucho de ser amables en su manejo; todo lo contrario, son complejos y requieren de personal experto para utilizarse con beneficio, en el Capítulo 1 se hace una breve descripción de uno de esos paquetes, Discovery, que es el que estamos usando actualmente.

Dentro del Laboratorio de Cibernética "Alejandro Medina" del departamento de Física, de la Facultad de Ciencias de la UNAM hemos venido realizando experimentos encaminados a la adquisición y procesamiento de potenciales de acción extracelulares de cerebro de rata anestesiada utilizando microelectrodos de tungsteno, adquiriendo y procesando las señales con un software comercial llamado Discovery y una tarjeta de conversión Analógico-Digital, Digital-Analógico de 12 bits (DT2821). El lograr el montaje experimental para obtener un registro limpio y procesable tomó un esfuerzo considerable y sobre la marcha fueron surgiendo necesidades, algunas de más equipo el cual se ha ido agregando a la disposición original, y otras de software que no siempre ha estado disponible comercialmente [14][15][16][22]. Uno de los requerimientos más importantes fue el de un simulador que permitiera en un tiempo relativamente corto, obtener una señal sintética para poder evaluar los resultados obtenidos de

registros de señales reales. El objetivo de esta tesis es desarrollar ese simulador y evaluarlo con circuitos de neuronas simuladas cuyas relaciones temporales se conocen bajo diferente conexionamiento, con esto nace la idea de desarrollar GESSIN (Generador de Señales Sintéticas), cuyas características son planteadas en términos de las necesidades propias del Laboratorio, pero pensando también en un usuario general.

GESSIN se ha desarrollado para generar señales sintéticas (trenes de espigas) similares a las que se obtienen en registros electrofisiológicos extracelulares. El sistema permite sintetizar de manera analógica una señal compuesta en donde las espigas (disparos) de cada neurona simulada a través del simulador Neurored [24], pueden colocarse de acuerdo a los deseos del diseñador de la señal. La base de señales es bastante amplia (72 potenciales de acción diferentes) de manera que permite manipular, por ejemplo, la presencia de espigas muy parecidas unas de las otras.

A lo largo de este trabajo de tesis se discuten en primer lugar dentro del Capítulo 1 las ideas generales acerca de la separación de espigas y se hace una breve reseña de las técnicas más usadas actualmente y reportadas en la literatura, así como también se hace una descripción del Software comercial Discovery y algunos otros paquetes. En el capítulo 2 se describe el funcionamiento y la programación de GESSIN. En el capítulo 3 se presentan tanto los resultados de las simulaciones realizadas, como las herramientas para validar el buen funcionamiento del programa. En las conclusiones se evalúa el desempeño y la utilidad del simulador GESSIN y se proponen posibles líneas para continuar desarrollándolo. En el Apéndice A se presentan los listados fuentes de los programas para su rápida consulta en futuras modificaciones y en el Apéndice B se muestran las especificaciones técnicas de la tarjeta de conversión A/D D/A utilizada por Discovery.

En la actualidad la ingeniería eléctrica al igual que muchas otras disciplinas, no pueden ni deben quedar aisladas en el campo del conocimiento humano, la multidisciplinaria diversifica la investigación científica y crea nuevas disciplinas que también se van entrelazando de forma que ya no podemos poner límites tajantes de una disciplina a otra. Esta tesis contempla varias ramas de la ingeniería por tratarse de un trabajo multidisciplinario, se presenta dentro de un contexto de la ingeniería eléctrica por contener elementos que consideramos caen dentro de su campo, como son : procesamiento de señales, computación, simulación y electrónica digital. Ahora bien, como objetivo de esta tesis se presenta una herramienta desarrollada *exprofeso* para la interpretación de la conectividad funcional entre neuronas biológicas o simuladas que es un campo que pertenece tanto a la ingeniería eléctrica como a la neurociencia computacional y a las redes neuronales artificiales. ♦

Antecedentes

Como ya se mencionó en la Introducción, la necesidad del simulador GESSIN, surge de un montaje experimental para la adquisición, separación y análisis de señales neurofisiológicas. Describiré a continuación con algún detalle las características de dicho montaje. El corazón de nuestro dispositivo experimental es una computadora con el procesador AMD 586 a 133 MHz, con 256 Kb de memoria caché y 24 Mb de memoria RAM, bus PCI y ranuras EISA. Dentro de esta computadora tenemos una tarjeta A/D D/A DT2821-F-16SE de Data Translation, Inc. de 12 bits con error máximo de $\pm 1/2$ bit y una frecuencia máxima de muestreo de 32 kHz, tiene la capacidad de manejar 16 canales multiplexados en modo simple y 8 en modo bipolar (para todas las características de la tarjeta ver Tabla 7 en el Apéndice B). Aunado a lo anterior usamos una grabadora digital VR100A de Instrutech, Corp. que tiene la capacidad de grabar en cinta de videocasete VHS cuatro canales analógicos a la vez, y capacidad de reproducción analógica también de cuatro canales simultáneos, para esto usamos una videocasetera estéreo de alta fidelidad. Inicialmente tuvimos muchos problemas con una videocasetera monoaural porque en la reproducción aparecía mucho ruido aleatorio, dejando inservible cualquier registro, después de hacer algunas pruebas optamos por cambiarla a una de alta fidelidad, con esta última lo único que tuvimos que cuidar era que no estuviera activa ninguna de las opciones adicionales de pantalla con las que cuentan estas videocaseteras modernas.

En la Figura 1 se muestra la forma en que los componentes de nuestro montaje

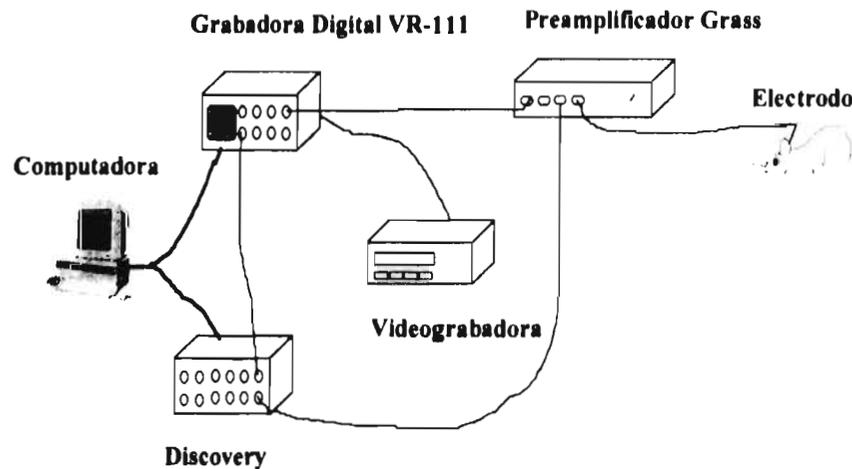


Figura 1 Disposición física del montaje experimental para el registro de señales neurofisiológicas

experimental se encuentran conectados. Primeramente la rata anestesiada es colocada en un estereotáxico que cuenta con un micromanipulador por medio del cual se puede colocar, con bastante precisión, un microelectrodo en una zona específica, usando un atlas del cerebro de la rata. La zona que particularmente nos interesa es el hipocampo. El electrodo manda las señales

censadas a un amplificador GRASS. Las señales, antes del amplificador son del orden de microvolts y después de él tenemos una señal que varía pico-a-pico alrededor de ± 10 mV. Esta es la que vamos a procesar usando software. El amplificador la manda a la grabadora digital para grabarla en cinta y poder hacer procesamiento fuera de línea. Pero también es necesario el análisis en línea porque con él podemos decidir como mover el electrodo para llegar a zonas de actividad dentro del hipocampo y tener señales ricas en información, para lograr esto es necesario contar con un software lo suficientemente rápido. En el análisis fuera de línea el tiempo no es lo primordial, los resultados obtenidos en él, obviamente no nos sirven para decidir que hacer mientras el electrodo se encuentra dentro del cerebro de la rata viva, más bien nos van a dar información acerca de como se estaban comunicando las neuronas en un instante determinado durante el experimento. Un simulador nos sirve en este tipo de experimentos para poder suponer el tipo de actividad y cotejarla con los resultados obtenidos dentro del análisis de señales reales, con esto podríamos identificar rápidamente el tipo de actividad que hay entre las neuronas que están siendo captadas por el o los electrodos.

En la Figura 1 se ve dibujada una caja que representa el software y hardware llamado Discovery, por medio del cual podemos separar potenciales de acción y analizarlos con diferentes herramientas como pueden ser: Transformada rápida de Fourier, análisis de correlación post y pre-estímulo, extracción de parámetros de la forma de onda, separación de espigas por características de su forma de onda y ajuste matemático. Aún con toda esta herramienta es necesario hacer alguna simulación antes del experimento para conocer de antemano el tipo de resultados a obtener. También en la Figura 1 se puede apreciar que la misma señal que se está grabando en cinta es alimentada directamente a Discovery, es ahí donde se puede hacer algún tipo de análisis en línea como puede ser la separación de espigas (Figura 2) usando parámetros de su forma de onda.

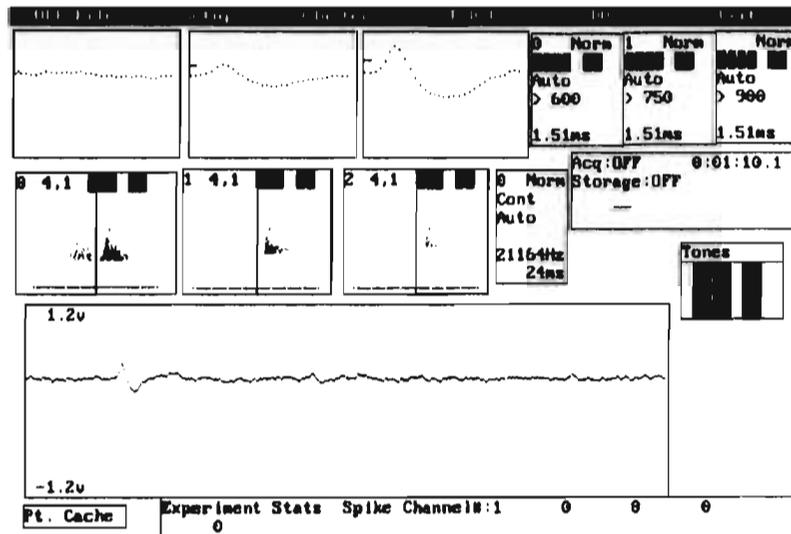


Figura 2 Ejemplo de un pantalla de adquisición de Discovery.

En la Figura 2 se puede observar una de las pantallas usadas en la captura de señales de tipo real. En este ejemplo se usaron tres canales de captura. En la parte superior se ven tres ventanas en las que se aprecian tres espigas separadas por Discovery, en las tres ventanas

siguientes hacia abajo se tienen gráficas XY en donde se están graficando proyecciones de las espigas usando dos de los ocho parámetros con los que Discovery permite hacer la clasificación. Cada punto en esas gráficas representa una espiga que cumple con tener características en su forma de onda o tiempos de ocurrencia, que caen dentro de un umbral previamente fijado por el usuario, es deseable por lo tanto que esos grupos de puntos se unan lo más posible en la gráfica XY, porque con ello tendríamos espigas con características similares, identificadas por Discovery. La ventana más grande, parte inferior, es simplemente una ventana tipo osciloscopio en donde podemos observar en tiempo real la señal que esta siendo adquirida por alguno o todos los canales en uso. Las demás ventanas son de control y contienen información previamente cargada por el usuario. Toda la pantalla de captura es

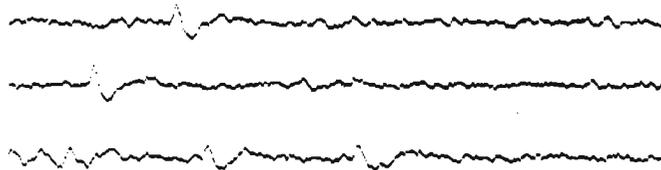


Figura 3 Ejemplo de una señal real, tres tramos de 20 ms

editable por el operador, quién puede cambiar tamaño y número de las ventanas dentro ella.

Las señales adquiridas con el ejemplo de la pantalla de adquisición (Figura 2) anterior se muestran en la Figura 3. Esta es una señal real en su traza continua, mostrando sólo tres segmentos de 20 milisegundos, en ella se puede observar algo de ruido pero la razón señal-ruido es muy grande por lo que Discovery fue capaz de separarlas adecuadamente. Esta señal es de un experimento realizado dentro del Laboratorio de Cibernética (septiembre 1992) [15].◆

Capítulo 1

La Neurona.

Introducción.

Para hablar de comunicación entre neuronas es necesario hacer una breve descripción anatómica de ellas, en ésta tesis sólo hablaré de conceptos generales relacionados sin llegar a profundizar en ninguno de ellos pero para aquellos que pudiesen estar interesados cito una serie de referencias a lo largo del texto a las que pueden acudir para más detalles del tema.

Examinaremos primero a la neurona individual y después las uniones sinápticas, para seguir hablando acerca de los potenciales de acción y terminar este capítulo haciendo un recuento de las técnicas modernas más usadas para la separación de potenciales de acción. Dentro de este último tema hablaré también del paquete comercial que dentro del Laboratorio de Cibernética, hemos venido utilizando en los experimentos para detectar y separar espigas (potenciales de acción) del cerebro de rata.

Breve descripción de la Fisiología de una neurona.

Nuestro cerebro y médula espinal, contienen células especializadas llamadas células nerviosas o neuronas, a las cuales colectivamente nos referimos como el Sistema Nervioso Central (SNC). Durante algún tiempo se pensó que el sistema nervioso era continuo, pero ahora sabemos, con bases firmes, que la neurona es la unidad discreta fundamental del SNC.

El sistema nervioso es extremadamente complejo y se estima que el número de neuronas en un cerebro humano, sin contar el cerebelo, son de alrededor de 10^{10} , lejos de estar simplemente agrupadas, forman vías, circuitos y capas con una estructura altamente flexible, en la que cada neurona ocupa un lugar, recibe estímulos de muchas otras y se conecta mediante su axón y sus ramificaciones solamente con ciertas neuronas.

De región a región dentro del SNC, las neuronas difieren en funciones y propiedades, sin embargo podemos definir cuatro componentes básicos (ver Figura 4). Estos componentes y sus funciones usuales son:

Cuerpo de la célula o SOMA.

Esta es la parte central de la neurona desde la cual se derivan las estructuras de comunicación. Delimita perfectamente las partes de entrada o recepción de información de las partes de salida o transmisión de información.

Dendritas.

Hay usualmente varias dendritas y estas se ramifican en una estructura de árbol, a esas estructuras se les conoce como arboles dendríticos. Sobre las dendritas pueden haber contactos de otras neuronas en sitios especializados llamados sinapsis, aunque estos a menudo se pueden encontrar sobre el soma. Las dendritas son los principales componentes de manejo de

información, ellas algunas veces realizan la integración y procesamiento de la información de entradas sinápticas.

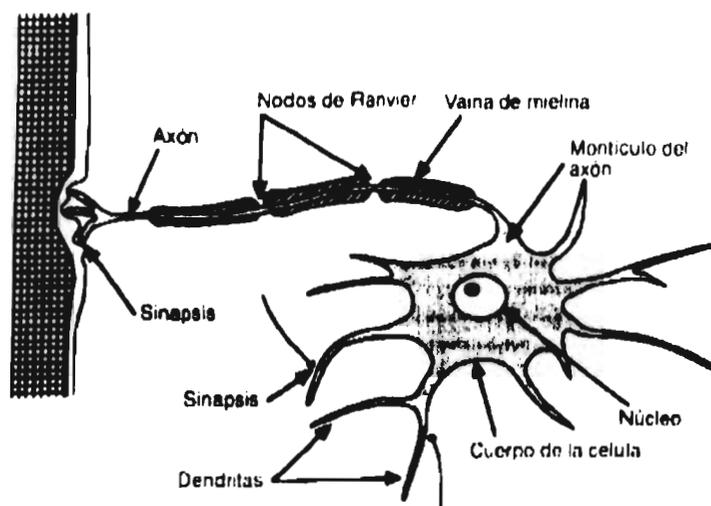


Figura 4 Esquema de la estructura de una neurona biológica. [19]

Axón.

El axón es el componente a través del cual las señales eléctricas de salida se propagan. La señal de salida es llamada "potencial de acción" y puede viajar a velocidades en un rango entre 2 hasta 120 m/s, para eventualmente llegar a neuronas destino a través de los contactos sinápticos. El axón de muchas neuronas está rodeado por una membrana que se denomina vaina de mielina, los nodos de Ranvier (ver Figura 4) la interrumpen periódicamente a lo largo de aquél. En las sinapsis se conecta el axón de la neurona con distintas partes de otras neuronas.

Terminal Axónica.

También conocida como terminal nerviosa, es la porción terminal del axón. Es una estructura bien diferenciada que en un sitio específico de su membrana tiene un engrosamiento que contiene pequeñas bolsitas llamadas vesículas sinápticas. Este sitio en proximidad con otro sitio de la membrana de una dendrita o un soma neuronal de otra neurona, constituye la sinapsis.

La membrana de la neurona separa el plasma intracelular del fluido extracelular, es permeable a ciertos iones, y actúa de tal forma que mantiene una diferencia de potencial entre el fluido intracelular y el extracelular. Este efecto se consigue primordialmente mediante la acción de una bomba de sodio-potasio. También están presentes otros tipos de iones, como son los iones de cloro e iones orgánicos negativos.

Todos los tipos de iones se pueden difundir a través de la membrana, con excepción de los iones orgánicos, que son demasiado grandes, como estos no pueden salir, su carga negativa neta dificulta la entrada de iones de cloro por difusión, por lo tanto, habrá una concentración más alta de iones de cloro fuera de la célula. La bomba de sodio-potasio determina una concentración más alta de potasio dentro y un más baja de sodio dentro de la neurona.

La membrana celular es selectivamente más permeable a los iones de potasio que a los de sodio. El gradiente químico del potasio tiende a hacer que los iones de potasio salgan de la célula por difusión, pero la fuerte atracción de los iones orgánicos negativos tienden a mantener dentro al potasio. El resultado de estas fuerzas opuestas es que se alcanza un equilibrio en el

cual hay más iones de sodio y cloro fuera de célula, y más iones orgánicos y de potasio dentro de ella. Además el equilibrio resultante produce una diferencia de potencial a través de la membrana de unos 70 a 100 milivolts, siendo el más negativo el fluido intracelular.

Los impulsos nerviosos que llegan al final de un axón, producen cambios electroquímicos en las sinapsis y pueden dar lugar a cambios locales en el potencial del soma de la neurona receptora. Estos, llamados potenciales graduados o potenciales de entrada, pueden propagarse por el soma de la célula. Pueden ser o bien excitatorios (que hacen disminuir la polarización de la célula) o bien inhibitorios (que incrementan la polarización de la célula). Los

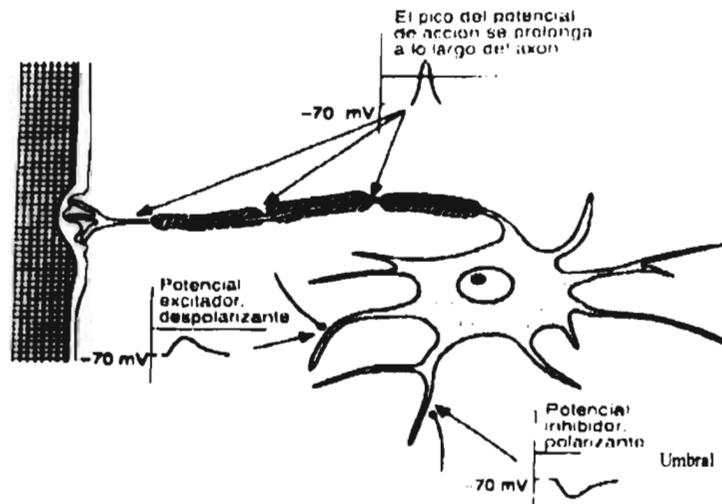


Figura 5 Generación de un potencial de acción.[19]

potenciales de entrada se suman en el montículo del axón. Si la cantidad de despolarización del montículo del axón es suficiente, se genera un potencial de acción; este potencial viaja a lo largo del axón, alejándose del cuerpo principal de la célula (ver Figura 5).

Las fibras nerviosas en sí son malos conductores. La transmisión del potencial de acción a lo largo del axón es el resultado de una serie de despolarizaciones que tienen lugar en los nodos de Ranvier. Cuando uno de los nodos se despolariza, se desencadena la despolarización del siguiente nodo. El potencial de acción viaja a lo largo de la fibra en forma discontinua de un nodo a otro. Una vez que el potencial de acción ha pasado por un cierto punto, ese punto no puede volver a ser excitado durante cosa de 1 milisegundo, que es el tiempo que tarda en volver a su potencial de reposo. Este período refractario limita la frecuencia de transmisión de los impulsos nerviosos a unos 1,000 por segundo.

La unión sináptica.

Actualmente podemos hablar mucho acerca de la actividad sináptica, sin embargo aquí sólo daré una muy esquemática descripción de lo que ahí ocurre, una descripción mas biológica y química del proceso se puede consultar en [20] y una más matemática en [17].

La comunicación entre neuronas tiene lugar como resultado de la liberación de unas sustancias llamadas neurotransmisores por parte de la célula presináptica, al ser absorbidas estas sustancias por la célula postsináptica. La Figura 6 muestra esta actividad. Cuando el

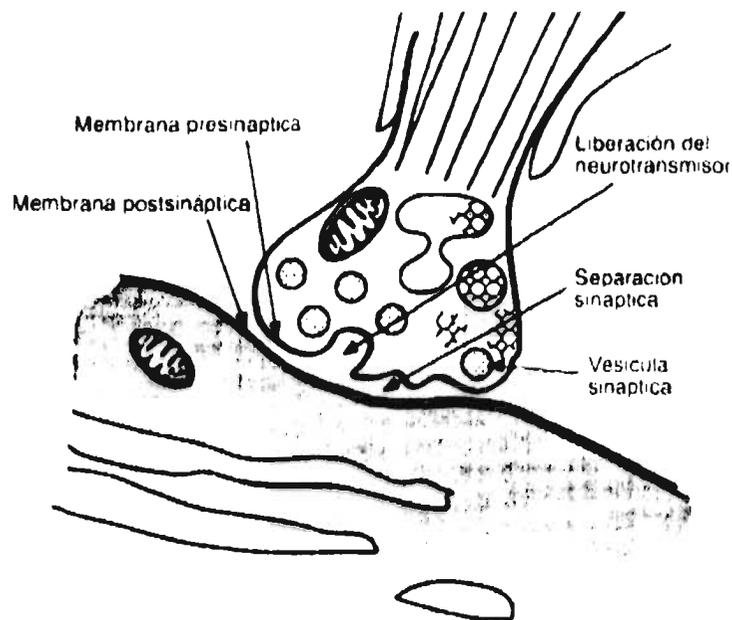


Figura 6 Descripción de la región sináptica.

potencial de acción llega a la presináptica, los cambios de permeabilidad de la membrana dan lugar a un flujo entrante de iones de calcio. Estos iones dan lugar a que las vesículas que contienen los neurotransmisores se fundan con la membrana sináptica, liberando así sus neurotransmisores en la separación sináptica. Los neurotransmisores se difunden a través de la unión y se unen a proteínas en la membrana postsináptica para ciertos tipos de iones. Un flujo entrante de corrientes iónicas positivas hacia la célula tenderá a despolarizar el potencial de reposo; este efecto es excitatorio. Si entran iones negativos, se producirá un efecto hiperpolarizante; este efecto es inhibitorio. Estos dos efectos son locales, y actúan tan sólo a lo largo de una pequeña distancia hacia el interior de la célula; se suman en el montículo del axón (ver Figura 4). Si la suma es mayor que un cierto valor umbral se genera un potencial de acción (ver Figura 5).

El potencial de acción.

El potencial de acción es, como ya se mencionó en párrafos anteriores, un impulso eléctrico que viaja a través del axón como consecuencia de una polarización y despolarización de la membrana por el efecto de corrientes iónicas. Un ejemplo de valores de típicos de un potencial de acción para una motoneurona son: para la despolarización de entre 12 y 15 mV (ver Figura 7) al umbral cuando el potencial de acción aparece, el total de la despolarización durante el potencial de acción es del orden de 100 mV. Puesto que el potencial de reposo de la membrana esta cerca de unos -70 mV, la polaridad de la célula es brevemente negativa. Cabe hacer notar que el evento principal es la drástica despolarización y esta ocurre en alrededor de

un milisegundo. A lo largo de esta tesis y algunas textos es común referirse a un potencial de

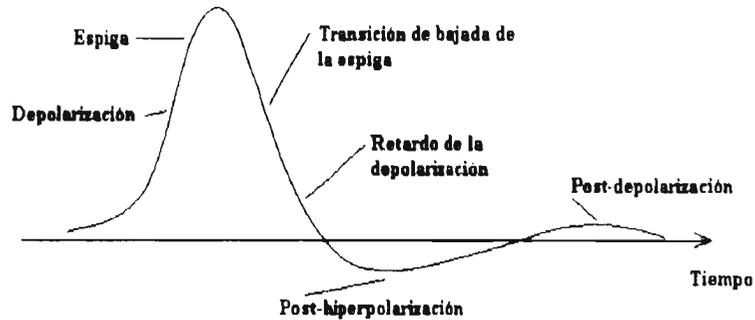


Figura 7 Detalles del potencial de acción de una neurona.

acción como una “espiga”¹, debido precisamente a que durante despolarización la caída de potencial es muy brusca lo que provoca que el potencial de acción sea realmente muy “espigado”.

En siguiente apartado, hablaré de la separación de espigas pero es necesario, para ello, que aquí defina los parámetros de la forma del potencial de acción que se usarán entonces, la definición de estos lo haré en base al estándar que usa el paquete comercial Discovery, pero en

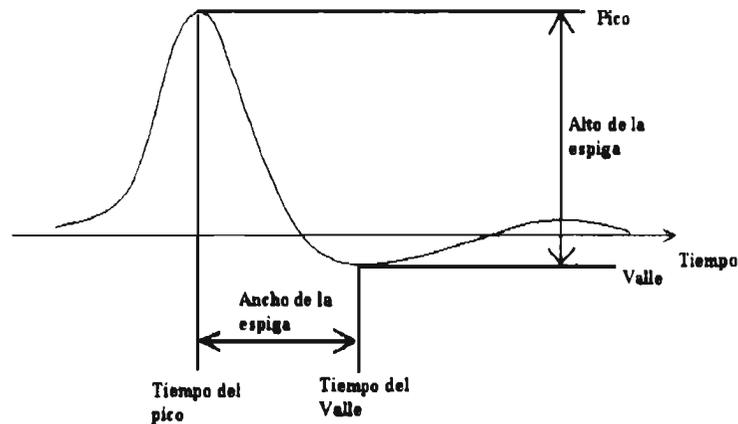


Figura 8 Parámetros de la forma de onda de un potencial de acción.

la mayoría de los paquetes comerciales usan las mismas definiciones de ellos para extraer y clasificar las espigas de la traza de una señal real, aunque algunos de ellos también usan algunas características calculadas como componentes principales, y plantillas generadas a partir de los parámetros básicos como son: pico, valle, tiempo de pico, tiempo de valle alto de la espiga ancho de espiga . En la Tabla 1 se definen estos parámetros y en la Figura 8 se pueden ver gráficamente.

¹ En inglés: spike.

Pico	Máxima amplitud de la espiga
Valle	Mínima amplitud de la espiga
Tiempo de pico	Tiempo al cual ocurre el máximo de la espiga
Tiempo de valle	Tiempo al cual ocurre el mínimo de la espiga
Alto de la espiga	La diferencia: Pico - Valle
Ancho de la espiga	La diferencia: Tiempo de valle - Tiempo de pico.

Tabla 1 Definición de los parámetros de la forma de onda de la espiga de una neurona

El registro de potenciales de acción.

En los simuladores de la actividad eléctrica de las neuronas se toman como componentes base a los potenciales de acción [26] registrados experimentalmente por medio de microelectrodos de metal o de vidrio. Con microelectrodos de metal se registra la actividad llamada extracelular que incluye a las neuronas que están cerca del microelectrodo. La señal obtenida de esta manera está compuesta de las contribuciones de cada neurona y representa el potencial de campo. Por otra parte, si se utiliza una micropipeta de vidrio, lo que se registra es la actividad eléctrica de la neurona en que ésta se clava y que se conoce como registro intracelular. La señal es el potencial de la membrana. En este trabajo nuestro interés está centrado en los registros extracelulares porque son los que llevan información de varias neuronas y, por eso, pueden servir para conocer si tales neuronas se comunican entre ellas. Si se hace, por medio de registro extracelular, un muestreo suficientemente amplio de la actividad eléctrica neuronal de una zona, es posible reunir una base de señales suficientemente grande que puede servir tanto como para reconocimiento de potenciales como para producir simuladores de señales extracelulares, entre otras aplicaciones [1][27]. Así pues, es posible simular señales como las que se obtienen al registrar con microelectrodos de metal. La disponibilidad de señales sintéticas facilita las tareas de análisis e interpretación de los datos que se obtienen experimentalmente [23][15] y ese es precisamente el objetivo de este trabajo.

Técnicas de separación de potenciales de acción.

Introducción.

Inicialmente los neurofisiólogos no contaban con herramientas computacionales, sus registros eran impresos en papel usando agujas entintadas, el análisis de éstos eran, sin lugar a dudas, muy tediosos y en muchos casos un tanto imprecisos. Generalmente este tipo de registros se hacían utilizando ventanas de umbral. Con el desarrollo de la computación, la electrónica y la aparición de los circuitos integrados VLSI¹, el registro neurofisiológico tuvo grandes adelantos, ya que ahora se podía hacer preprocesamiento en línea durante la adquisición. Los DSP's² son circuitos integrados que se usan actualmente, entre otras cosas, para separar y clasificar espigas, un ejemplo de su uso en este contexto se puede consultar en

¹ Very Large Scale Integration (Muy alta escala de integración)

² Digital Signal Processor (Procesador de Señales Digitales).

[16]. A continuación haré un breve recuento de las técnicas más usadas para la separación y clasificación de espigas, para más detalles al respecto ver [2][3].

Componentes Principales.

Muchos de los paquetes comerciales que actualmente se pueden conseguir en el mercado se basan en la detección de espigas usando comparación contra plantilla y/o calculando componentes principales. A continuación se dan los principios básicos de estas ideas [1].

Al discretizar una señal neurofisiológica continua en el tiempo, la podemos considerar como vector en un espacio lineal. Como estas señales tienen un ancho de banda limitado $[0, \Delta F]$, por el teorema de Nyquist deben ser muestreadas por los menos a una frecuencia $2\Delta F$, además son impulsivas, es decir que

$$\int_{-\infty}^{\infty} |X(t)| dt < \infty$$

por lo que sólo nos interesan en el intervalo $[T_1, T_2]$ en donde está contenida la mayor parte de la energía de la señal. Para señales neurofisiológicas tenemos que ΔT es del orden de los 2 milisegundos para un ancho de banda de 10 kHz lo cual implica que los vectores deben ser de por lo menos 40 puntos, entonces es posible tratar a las señales como vectores en un espacio $2\Delta F\Delta T$ -dimensional con $2\Delta F\Delta T$ vectores linealmente independientes. Este mismo espacio puede ser usado como base para la descripción de la señal. Por ejemplo las funciones

$$\begin{aligned} \text{sen}\left(\frac{2\pi k t}{\Delta T}\right) & \quad k = 1, 2, \dots, \Delta F\Delta T \\ \text{cos}\left(\frac{2\pi k t}{\Delta T}\right) & \quad k = 1, 2, \dots, \Delta F\Delta T \end{aligned}$$

pueden representar tal base. La transformada clásica de Fourier es la representación de nuestra señal en esa base y su inversa es la reconstrucción de la señal en su representación en el dominio del tiempo desde su representación sobre el dominio de la frecuencia. Pero lo anterior es válido para señales no impulsivas por lo que para las señales que aquí nos atañen, tenemos que generar un base diferente. Para hacerlo conviene definir la siguiente notación:

$\langle x$	Una señal (vector renglón).
$\tilde{x} \rangle$	El mismo vector transpuesto.
$\langle x, g \rangle$	Producto interno = $\sum_{i=1}^n x_i g_i$ para el caso discreto
	$\int_{T_1}^{T_2} x(t)g(t)dt$ para el caso continuo
$\langle x, \tilde{x} \rangle^{1/2} \equiv \ x\ $	La norma de la señal.
$\tilde{x} \rangle \langle x$	Una matriz o kernel. Si $\langle x$ es el miembro de un proceso estocástico, el valor esperado $E[\tilde{x} \rangle \langle x]$ es la función de autocorrelación del proceso. En el caso discreto $E[\tilde{x} \rangle \langle x]$ es usualmente llamado la matriz de correlación.

Desde el punto de vista matemático es más conveniente usar vectores ortonormales para generar una base con las siguientes propiedades: Si $\langle x$ es una señal y $g \rangle$ es un patrón normal, el producto interno $\langle x, g \rangle$ representa la mejor aproximación de $\langle x$ dado $g \rangle$ en el sentido de

que el error $\langle E_r \rangle = \langle x - \alpha \rangle$ es la mínima longitud para $\alpha = \langle x, g \rangle$. Por lo tanto, dado un conjunto de n plantillas ortogonales $\langle g_1 \rangle, \langle g_2 \rangle, \dots, \langle g_n \rangle$ y una señal $\langle x \rangle$, la mejor aproximación (mínimos cuadrados) de la forma:

$$\langle x_{aprox} \rangle = \alpha_1 \langle \tilde{g}_1 \rangle + \alpha_2 \langle \tilde{g}_2 \rangle + \dots + \alpha_n \langle \tilde{g}_n \rangle$$

se da cuando

$$\alpha_1 = \langle x, \tilde{g}_1 \rangle, \alpha_2 = \langle x, \tilde{g}_2 \rangle, \dots, \alpha_n = \langle x, \tilde{g}_n \rangle.$$

También bajo estas circunstancias

$$\|x_{aprox}\|^2 = \alpha_1^2 + \alpha_2^2 + \dots + \alpha_n^2$$

$$\|E_r\|^2 = \|x\|^2 - \|x_{aprox}\|^2$$

Una descripción gráfica de lo anterior la podemos ver en la Figura 9.

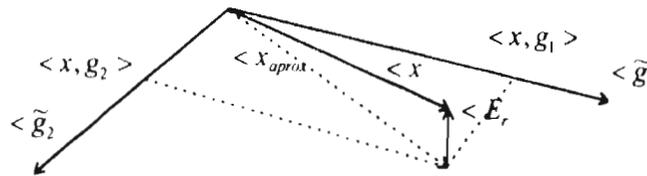


Figura 9 Representación gráfica del ajuste por mínimos cuadrados. Se hace la aproximación al vector medido $\langle x \rangle$ con la combinación lineal de dos vectores ortonormales $\langle \tilde{g}_1 \rangle$ y $\langle \tilde{g}_2 \rangle$. La mejor aproximación se da cuando el vector de error $\langle E_r \rangle$ es perpendicular al plano generado por los vectores ortonormales.

Si el error es relativamente pequeño respecto al tamaño de la señal, podemos considerar que la secuencia $\alpha_1, \alpha_2, \dots, \alpha_n$ es representativa de la señal. Esta representación es isomorfa a un punto en un espacio n -dimensional¹. Por lo tanto nosotros siempre podemos describir completamente nuestra señal con $2\Delta F\Delta T$ plantillas ortogonales.

El uso de análisis espectral es en muchos casos conveniente porque las señales están ampliamente distribuidas en el dominio del tiempo, pueden ser ampliamente concentradas cuando se describen en dominio de la frecuencia.

Realización de plantillas. Dada una señal continua $\langle x = x(t) \rangle, 0 \leq t < \Delta T$ y una plantilla $\langle g \rangle = \langle g(t) \rangle, 0 \leq t < \Delta T$, el producto interno

$$\langle x, g \rangle = \int_0^{\Delta T} x(t)g(t)dt$$

es similar a la convolución $x * h$ evaluada en $t = 0$, si

$$h(-t) = \begin{cases} g(t) & \text{si } 0 \leq t < \Delta T \\ 0 & \text{de otra manera.} \end{cases}$$

recordemos la integral de convolución

$$\int_{-\infty}^{\infty} x(\tau)h(t - \tau)d\tau$$

para nuestro caso

¹ Esta propiedad es usada por el paquete Discovery.

$$x * h = \int_{-\infty}^{+\infty} x(\tau)h(t-\tau)d\tau = \int_{-\infty}^{+\infty} x(\tau)g(t-\tau)d\tau$$

y evaluada en $t = 0$

$$x * h|_{t=0} = \int_{-\infty}^{+\infty} x(\tau)g(\tau)d\tau.$$

Por lo tanto podemos alimentar muestra señal a través de un conjunto de filtros ortonormales con respuesta al impulso $h_i(t) = g_i(-t)$ y muestreando las salidas al tiempo $t = 0$ uno puede obtener eléctricamente la representación de la señal en la base $\langle \tilde{g}_i \rangle$. Un filtro real no puede tener una respuesta al impulso extendida dentro tiempos negativos como se requiere $h(t) = g(-t)$, $0 \leq t < T$; sin embargo, retardando todo el proceso un tiempo ΔT , es decir, $h(t) = g(\Delta T - t)$, $0 \leq t < T$ resulta un sistema realizable en el cual la salida al tiempo ΔT ,

$$x * h_i|_{t=\Delta T} = \langle x, g_i \rangle$$

representa nuestra señal en términos de los elementos base $\langle \tilde{g}_i, i = 1, 2, \dots, n \rangle$.

Con técnicas digitales los vectores base $\langle \tilde{g}_i \rangle$ se pueden calcular como los vectores propios de la matriz $E[\tilde{x} \langle x \rangle]$, con valores propios λ_i asociados tales que $\lambda_i = E[\langle x, s_i \rangle^2]$ hace que todos los vectores $\langle x \rangle$ sean normales. En procesos estocásticos a estos vectores

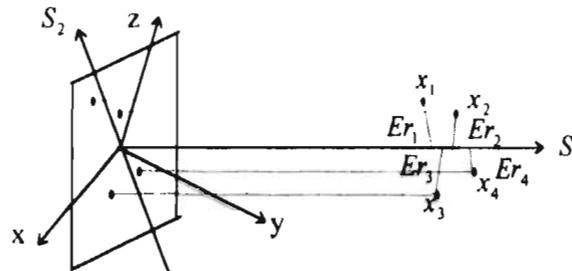


Figura 10 Representación gráfica de los componentes principales.

propios se les llama funciones Karhunen-Loève, en ciencias sociales componentes principales. Una interpretación gráfica de ellos la podemos ver en la Figura 10. Ahí los cuatro vectores de señales x_1, x_2, x_3, x_4 son representados como cuatro puntos en un espacio de tres dimensiones. El vector S_1 es la primera componente principal, es decir, es dirección que minimiza la proyección media-cuadrática de las señales x , o minimiza los errores cuadráticos medios Er_i . Después la componente S_1 de cada vector señal ha sido quitada y se han dejado únicamente los vectores residuales Er_1, \dots, Er_4 , perpendiculares a S_1 . Estos vectores son proyectados dentro de un plano que pasa por el origen y es ortogonal a S_1 . Este conjunto de vectores residuales proyectados son representados por cuatro puntos en un espacio bidimensional. S_2 es obtenido similarmente.

Como se había mencionado, uno obtiene las componentes principales a partir de calcular la función de autocorrelación de las señales. La autocorrelación de espigas puede ser estimada experimentalmente al adquirir muchas de ellas $\{\langle x_i \rangle\}$ provenientes de neuronas diferentes y calcular la autocorrelación promedio:

$$C = \frac{1}{N} \sum_{i=1}^N \tilde{x}_i \langle x_i \rangle.$$

Se requieren dos preprocesamientos de las señales $\langle x_i \rangle$, antes de poder calcular la autocorrelación. 1o.) Todos los vectores tienen que inicialmente ser normalizados. 2o.) Todas

las señales se suponen iniciadas al tiempo cero, pero en la práctica se tienen señales continuas y la decisión del tiempo cero para cada una de ellas es arbitraria y sujeta a errores. Para superar este problema las señales pueden ser recorridas en el tiempo hasta que entre sus formas ocurra una correlación máxima. Específicamente, después de la normalización de las señales se calcula la espiga promedio

$$\langle \tilde{f} \rangle = \frac{1}{N} \sum_{i=1}^N \langle f_i \rangle$$

y después cada espiga $\langle f_i \rangle$ es desplazada en el tiempo hasta que $\langle f_i, \tilde{f} \rangle$ sea máximo. Este proceso se repite hasta que no se requieran más desplazamientos. Sólo hasta entonces podemos calcular la función de correlación y obtener los valores y vectores propios.

Muchos analizadores de trenes multi-espigas, requieren realizar todo lo anterior en tiempo real, el hacerlo sólo en software, puede no ser óptimo pero esto depende de la plataforma sobre la cual esté corriendo, sin embargo existe actualmente, la electrónica necesaria para hacer todo esto en línea. Un ejemplo de esto se puede ver en [9], en donde Heetderks y Kwan presentan un sistema para detección de espigas, usando componentes electrónicos discretos y en [6] Kreiter, Aersten y Gerstein presentan otro sistema totalmente automatizado basado en el microprocesador MC68000.

Algunas técnicas en hardware.

En el artículo de Schmidt [2] se describen algunos instrumentos construidos específicamente para la detección y separación de espigas que utilizan diferentes técnicas, describirlas aquí nos llevaría algún tiempo y ese no es el objetivo del trabajo, sólo mencionaré brevemente en que consisten algunas de ellas.

Discriminadores.

Detectores de umbral. Es la primera técnica reportada en la literatura. Simplemente consistía en aceptar como actividad neuronal toda aquella señal que sobrepasara un umbral prefijado. Se podían fijar varios niveles de umbral para adquirir la actividad de varias neuronas en registro multiunitario, claro que esto podía provocar el tener muchas espigas espurias porque solamente se está tomando en cuenta la amplitud de la espiga.

Discriminadores de amplitud diferencial (DAD). Esta técnica utiliza básicamente el mismo principio que los detectores de umbral, con la diferencia de que éste utiliza dos niveles de referencia para encerrar entre ellos a la amplitud de la espiga a separar.

Cancelación de pulsos. Utiliza el mismo principio que el DAD. Pero aquí se utiliza un multivibrador de disparo que se activa cada vez que una espiga que sobrepase un cierto umbral inferior, y se vuelve a disparar cuando la señal rebasa un segundo umbral superior, el segundo pulso se resta al primero para su cancelación. Una cancelación indicaba la existencia de una espiga a separar.

Detección de transición principal. Este tipo de discriminadores utiliza una lógica de comparación analógica similar a la usada por un convertidor de A/D de rampa, en él se utilizan tres niveles de referencia. Cuando el instrumento detecta que la señal ha sobrepasado el nivel 1, genera un pulso de salida y espera a que esta sobrepase el nivel dos, y si la señal alcanza un máximo entre el nivel 2 y el 3, entonces se genera un segundo pulso que indica la detección de una espiga.

Detección de pico. Conceptualmente el método más poderoso de configuración de un DAD, es determinar cuando ocurre el pico de una espiga, para después determinar la ventana de tiempo que ocupa.

Discriminadores de ventana de tiempo simple. Este tipo de discriminadores utilizan un detector de umbral para eliminar el ruido basal y comenzar una secuencia tiempo. Aquí ya es importante tanto la amplitud de la espiga como su duración, la espiga debe tener la amplitud necesaria para caer entre dos niveles de amplitud prefijados, esto activa una ventana de tiempo, de tal forma que si el nivel de voltaje llega al potencial de reposo durante esa ventana de tiempo, entonces la espiga es clasificada en un grupo de acuerdo a los niveles de voltaje entre los que su amplitud máxima haya caído.

Discriminadores de ventana de tiempo dual. El principio elemental de funcionamiento es el mismo que el discriminador anterior, pero aquí la señal es pasada por dos ventanas de tiempo. Tiene un detector de umbral que inicia el proceso de reconocimiento. Se utilizan cuatro niveles de referencia, y la espiga debe alcanzar cada uno de los cuatro en un tiempo determinado por dos ventanas de tiempo que se activan cada vez que la espiga cae entre dos de ellos, de esta manera, aquí se está tomando en cuenta la forma de onda de la espiga. Es posible separar espigas utilizando diferencias en las formas de onda de registros simultáneos, en dos puntos del tiempo de la espiga.

Discriminadores multiparámetros. En la literatura (1976) se reportan algunos instrumentos [2] que utilizaban 4 parámetros de la forma de onda para clasificar espigas. Estos eran usados para discriminar mediante un dispositivo electrónico auxiliar a una computadora. Los parámetros usados eran: 7 niveles para el valor del pico positivo de la forma de onda, 7 niveles para el valor del pico negativo de la forma de onda, el intervalo de tiempo entre dos cruces de umbral preestablecidos y la secuencia de componentes positivos y negativos de la forma, cualquiera o todos estos parámetros pueden ser utilizados para la clasificación.

Todas las técnicas descritas eran técnicas que requerían electrónica analógica, con el auge de la electrónica digital otras técnicas se han ido desarrollando tanto en hardware como en software, algunas de ellas son:

Comparación contra plantillas. Con la velocidad que han alcanzado las nuevas computadoras, es posible diseñar programas que se encarguen de todo lo que antes se tenía que programar en forma alambrada. La detección y clasificación de espigas, en esta técnica requiere de un proceso de entrenamiento previo, durante el cual, el sistema es capaz de conformar una base de datos de las formas de onda promedio capturadas, que posteriormente se utilizarán como plantillas o patrones contra las que el software comparará para seguir con el proceso de captura. Esta idea, que en principio suena muy sencilla, tiene muchas variantes que durante los últimos años han sido reportadas en la literatura [4][7][10] y usadas por los paquetes comerciales.

Discriminadores por componentes principales. Combinada con la técnica anterior, ésta es una de las más usadas actualmente, tanto en software como hardware [1][4][5][6]. La definición y cálculo de componentes principales lo hice en el apartado de la página 15.

Algunos paquetes comerciales para la separación de potenciales de acción.

Discovery.

Es el paquete que actualmente estamos usando dentro del Laboratorio de Cibernética y utiliza el principio de componentes principales, parámetros de la forma de onda y comparación

contra plantilla para hacer la separación y clasificación de espigas. No es un paquete que se caracterice por ser amigable, por el contrario, requiere de muchas horas de trabajo llegar a entender su filosofía para poder explotarlo con resultados aceptables. Dentro del Laboratorio hemos trabajado ya con tres versiones del paquete, aunque existe ya una nueva que, según la propaganda supera en mucho a las anteriores, dice la compañía¹ haber superado muchos de los *bugs* de las versiones pasadas.

Cuenta con herramienta para hacer procesamiento de la señal fuera de línea, como puede ser transformada rápida de Fourier, extracción de parámetros de las formas de onda, histogramas de correlación y ajuste matemático de curvas. Puede muestrear hasta 16 canales multiplexados con una frecuencia máxima de 32 kHz. Usa un convertidor A/D D/A de 12 bits.

Existen otros paquetes comerciales, uno de Australia, otro de Israel y uno más de Estados Unidos.

MSD (Multi Spike Detector) Detector Multi-espigas.

Este es un paquete que vende Alpha Omega Engineering² de Israel basado en la técnica de comparación contra plantilla, que permite definir tres formas de plantilla para la separación y pueden ser modificadas en línea. Es capaz de manejar hasta 8 canales multiplexados. La propaganda indica que Abeles [1] y su grupo del departamento de fisiología de la Universidad Hebrea de Jerusalén lo probaron. El hardware consta de un DSP (no se especifica cual), un convertidor A/D de 12 bits y un convertidor D/A también de 12 bits. El costo aproximado del sistema completo era para 1994 de alrededor de \$12,000.00 DLS USA.

Waveform Discriminator (SPS-8701) Discriminador de forma de onda.

Este es un paquete que vende Signal Processing Systems³ de Australia, también basa su funcionamiento en la comparación de las espigas contra plantilla. Puede discriminar en tiempo real hasta tres unidades diferentes, funciona bajo la supervisión de un equipo de cómputo tipo PC-compatible. Puede muestrear a una frecuencia máxima de 90.9 kHz. Las plantillas son obtenidas por promediación. Las plantillas pueden adaptarse durante de la adquisición a criterio del operador. Sólo puede muestrear un canal para una señal de entre 10 mV y 2.5 V. para un ancho de banda de 64 kHz, pero no se especifica que convertidor usa. Este tiene un precio de alrededor de los \$4,000.00 DLS USA.

Multichannel Neuronal Acquisition Processor (NMAP). Procesador multicanal de adquisición neuronal.

Este es uno de los sistemas más completos pero por lo mismo más caro, dedicado a la adquisición y clasificación de señales neurofisiológicas. Lo distribuye Plexon, Inc⁴. de Texas, USA. Cuenta, en su configuración más elemental, con 16 canales de adquisición pero no se especifica si son o no multiplexados, ni que convertidor usa. Cuenta con DSP capaz de procesar hasta 32 canales para la captura y ordenamiento de espigas por comparación contra plantilla. El software corre bajo ambiente Microsoft Windows sobre una PC Pentium como

¹ Data Wave Technologies. E-Mail: datawave@usa.net

² Alpha Omega Engineering. E-Mail: alphaomg@netvision.net.il

³ Signal Processing Systems: Fax: +61 8 272 0802

⁴ Plexon, Inc. E-Mail: plexon@spsc.com

mínimo, también cuenta con una versión para Windows NT en estación de trabajo para versiones de adquisición de hasta 96 canales. Estos sistemas varían en precio desde \$26,000.00 DLS. para 16 canales, hasta \$68,600.00 DLS para 96 canales. ◆

Capítulo 2

El Simulador GESSIN.

Introducción.

El simulador GESSIN (Generador de Señales Sintéticas), fue escrito en el lenguaje de programación Pascal usando el compilador de Borland, Turbo Pascal 6.0, aunque es posible que algunos de sus módulos crezcan o se modifiquen en otro lenguaje como podría ser ANSI C. En las próximas páginas se hace una descripción tanto de su programación como de la forma en que tiene que ser utilizado y de cuáles son los formatos de los archivos de entrada y salida a cada uno de los módulos, hasta donde fue posible se respetó la compatibilidad con Neurored y con Discovery. Cabe hacer notar que son varios los archivos generados tanto por GESSIN como por Discovery, así que se intentó hacer manejable toda la información disponible. Para lograrlo es necesario cuidar que todos los archivos tengan el mismo nombre y que sea la extensión de los mismos la que nos indique el tipo de información que contiene cada uno de ellos. Más adelante se describen las extensiones que se sugieren como estándar. También es recomendable cuidar que todos los archivos queden concentrados en un mismo subdirectorio.

El uso completo del simulador está condicionado a contar con una tarjeta de conversión A/D D/A, porque de otra manera sería imposible tener la señal simulada en forma analógica. Sin una tarjeta de ese tipo sólo podríamos llegar a tener los histogramas de correlación cruzada que, dependiendo del objetivo, podrían ser suficientes.

Dependiendo del equipo sobre el cual sea ejecutado GESSIN, el tiempo de conversión D/A puede variar. Inicialmente se empezó a programar GESSIN usando una computadora DELL 333D, que tenía un procesador Intel 386DX a 33 MHz con coprocesador matemático y 10 Mb de memoria RAM, pero se terminó usando una computadora con un procesador AMD 586 a 133 MHz y con 24 Mb de memoria RAM. El cambio en los tiempos para tener disponible un dato en el DAC de la tarjeta DT2821 fue significativo pero no modificó la estructura del simulador. Inicialmente teníamos un tiempo de conversión de 23.4 ns y éste cambió a 7.8 ns al usar el procesador 586, lo cual se traduce a que el ancho de una espiga de 128 puntos de la base de datos usada, fue primero de 3 ms y después de 2 ms, el único cambio necesario para nuestros ejemplos fue cambiar el ancho de la ventana de captura para una espiga en Discovery.

A continuación se describen cada uno de los módulos, que en realidad, pueden verse y ejecutarse como programas separados porque son independientes uno del otro en cuanto a la ejecución se refiere. Todos se pueden acceder mediante un archivo de ejecución por lotes (GESSIN.BAT) en donde se cuenta con un menú que nos lleva a cada uno de los módulos.

Instalación de GESSIN.

Los requerimientos mínimos para ejecutar GESSIN son: procesador 80386 o superior con al menos 4 Mb de memoria RAM y monitor VGA color, al menos 2 Mb de espacio libre en disco duro aunque dependiendo de las simulaciones, se puede llegar a necesitar mucho más espacio libre, además es necesario contar con la tarjeta A/D D/A DT2821 para poder usar completamente GESSIN.

El simulador puede ser ejecutado módulo a módulo dado que cada uno de ellos es en realidad un programa ejecutable desde DOS, pero se cuenta con un archivo de ejecución por lotes desde el cual se tiene acceso a cada uno de ellos. Conviene crear subdirectorios para manejar adecuadamente la información. En el disquete de GESSIN se encuentra un archivo que permite copiar a disco duro todo el simulador creando un subdirectorio llamado GENERA a partir del cual se pueden ir creando subdirectorios para guardar los datos. Los archivos contenidos en ese disquete son:

INSTALAR.	BAT
DAC-ESP3.	EXE
PANTALLA.	BAT
UNE-VEC9.	EXE
XCOR-8.	EXE
ESPIGAS.	DAT
GESSIN.	BAT
EGAVGA.	BGI
LEEME.	TXT

Para instalar es necesario ejecutar el archivo INSTALAR.BAT, bajo el siguiente formato:

INSTALAR [drive fuente] [drive destino]

por ejemplo:

INSTALAR A: C:

y puede llamarse desde cualquier drive. Si no se utiliza el formato adecuado, el programa manda un mensaje mostrando la forma correcta y termina. Si ya existía una versión instalada de GESSIN, entonces el programa dará aviso de esto pero bajo la opción de continuar, sobrescribirá los archivos anteriores de la versión anterior con los nuevos, excepto los archivos de datos. Una vez completada la instalación con éxito, en el subdirectorio \GENERA deberán haber aparecido los archivos:

DAC-ESP3.	EXE
UNE-VEC9.	EXE
XCOR-8.	EXE
ESPIGAS.	DAT
GESSIN.	BAT
EGAVGA.	BGI

En caso de no aparecer todos estos, GESSIN no funcionará adecuadamente, por lo que en tal caso será necesario repetir la instalación o copiar manualmente todos o algunos de ellos.

Módulo de generación de la traza, UNE-VEC.

Todas las simulaciones que se discutirán más adelante en este trabajo, se generaron usando la base de datos de espigas de Moshe Abeles. Esta cuenta con 72 espigas (potenciales de acción) obtenidas en registros de la corteza auditiva primaria de gato. Cada espiga consta de 128 datos, que originalmente estaban en formato octal codificado en ASCII por lo que para poder usarlas fue necesario convertirlas a un formato decimal codificado en ASCII, y para poder llegar rápidamente a cada una de ellas se generó un archivo de registros en binario llamado ESPIGAS.DAT este contiene vectores de 129 elementos de tipo real. Se agregó un primer elemento a cada vector de espiga para contar en cualquier momento con el número de ella dentro de la base de datos y así identificarlas rápidamente.

El módulo UNE-VEC se usa para construir un vector de salida que contiene la información de la traza (señal sintética), construida a partir de un archivo de salida de Neurored que contiene los tiempos de disparo de las neuronas usadas en un circuito simulado en ese programa. A continuación se describe la información necesaria para ejecutar este módulo así como el formato del archivo de salida.

La primera pantalla de este módulo presenta una ruta de búsqueda por defecto, ahí se puede proporcionar una ruta nueva válida a donde el programa irá a buscar y a escribir todos los archivos de datos de entrada y/o salida.

La segunda pantalla se muestra en la Figura 11 y se describe a continuación línea por línea.

```

1 Nombre del archivo de las espigas patrón : espigas.dat
2 Nombre del archivo de tiempos de disparo : inhibel.sal
3 El archivo \genera\datos\inhibel\inhibel.sal tiene 4 columnas de tiempos.
4 Cuantas neuronas se usaran ? : 2
5 Posición de las neuronas en el archivo ? : 1
6 Posición de las neuronas en el archivo ? : 2
  
```

Figura 11 Segunda pantalla de UNE-VEC.

Línea 1: El archivo patrón por defecto es ESPIGAS.DAT descrito al inicio de este capítulo, en el tenemos 72 espigas. Si se quiere usar otra base de datos, en esta línea en vez de dar ENTER, se debe proporcionar la ruta y nombre de la nueva base de datos.

Línea 2: El archivo de disparos es el archivo de salida de una simulación de Neurored, pero dado el formato de ese archivo este puede ser generado o editado en cualquier procesador de texto salvándolo en formato "sólo texto" (ASCII) con CR y LF.

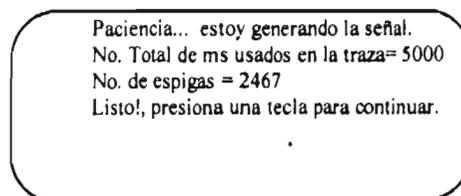
Línea 3: En esta línea el programa proporciona al usuario, la información concerniente al archivo de disparos. El número de columnas (neuronas) que contiene. Todas pueden ser usadas como series de disparos.

Línea 4: Aquí el usuario especifica cuantas de las columnas anteriores van a ser usadas en la generación de la traza de señal simulada, pueden ser usadas una o todas ellas.

Línea 5 y Línea 6 y las líneas que sigan dependiendo del número de neuronas usadas. Aquí se indica que columna del archivo le corresponde a que neurona, el número dado se le asignará consecutivamente a cada neurona. Las columnas pueden ser usadas en cualquier orden.

Después de la pantalla descrita anteriormente, el programa pide el nombre del archivo para la señal compuesta, este es un archivo de salida cuya extensión, en el nombre, se recomienda poner como .SEN, el formato de este archivo de salida se describe a continuación. Se tienen tres arreglos de una longitud de 30,000 elementos de tipo real. Los dos primeros contienen información necesaria para su recuperación y uso en otro módulo y el tercero contiene los datos propios de la señal simulada. El 1er. arreglo contiene en el primer elemento el número de milisegundos usados, el 2o. arreglo tiene en el primer elemento el número de neuronas usadas, así como, en los elementos siguientes, las posiciones (números de columnas) de cada una de ellas en archivo de disparos. Finalmente el 3er. arreglo tiene la señal sintética, cada elemento del arreglo corresponde a un milisegundo y contiene el número de la espiga del archivo de la base de datos correspondiente, en caso de no haber disparo en ninguna de las neuronas de ese milisegundo, ese elemento contiene un cero.

A continuación existe la posibilidad de ver las espigas del archivo de la base de datos llamadas espigas patrón Figura 13, o se puede continuar directamente la ejecución del programa. Después se pide al usuario las formas de los potenciales de acción a ser usados para cada neurona, los números corresponden a los asignados a cada forma en la base de datos y por lo tanto deben estar entre 1 y 72. En este punto el programa genera la señal sintética y se presenta la información de la señal resultante: número de milisegundos de la traza y de ella el número total de espigas resultantes distintas de cero. Este último dato servirá como referencia cuando la señal sea adquirida usando el Discovery para saber el grado de precisión con el que éste logra detectar y separar las espigas. Un ejemplo de esa información está en la Figura 12



Paciencia... estoy generando la señal.
No. Total de ms usados en la traza= 5000
No. de espigas = 2467
Listo!, presiona una tecla para continuar.

Figura 12 Información de UNE-VEC al generar la señal.

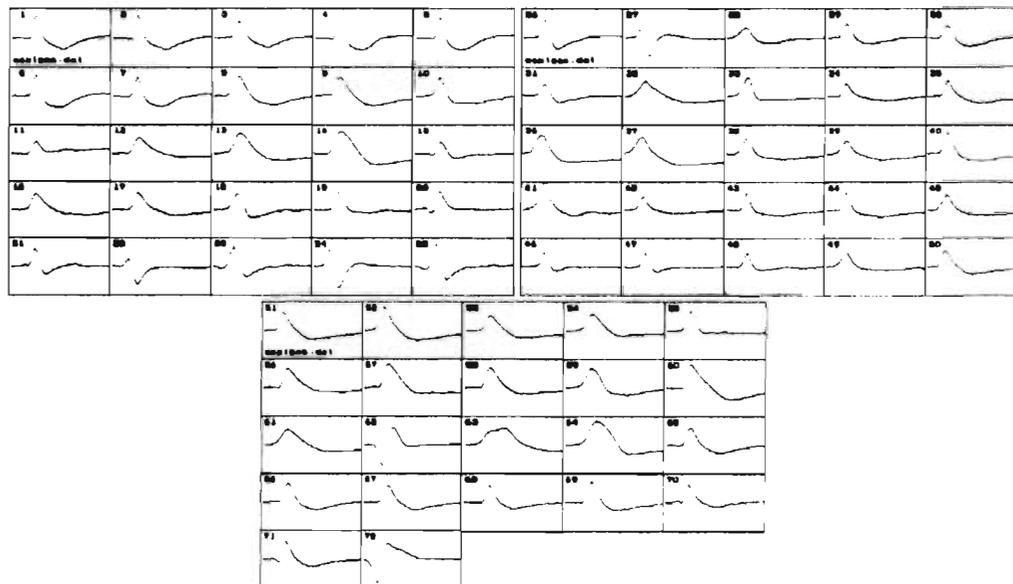


Figura 13 Formas de onda de las espigas de la base de datos usada.

Una vez que el programa ha generado la traza de la señal, esta puede ser vista en pantalla para lo cual el programa cuestiona al usuario acerca de su deseo de verla, la señal es presentada por páginas de 240 milisegundos, ver Figura 14, teniendo siempre visible una referencia del tiempo. Para una traza muy grande puede ser muy engorroso tener que verla toda, así que si el usuario ya entró a esta opción puede dejarla simplemente presionando la tecla "s" en cualquier momento para continuar con la ejecución del programa.

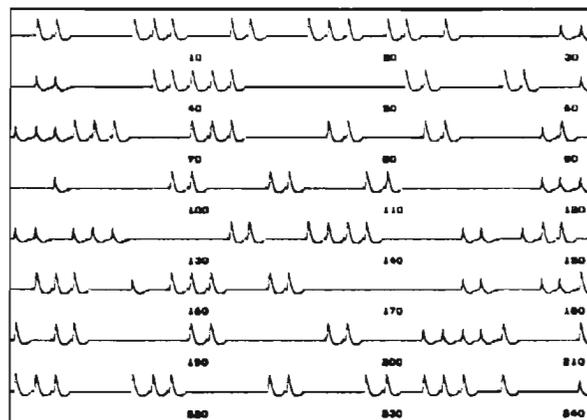


Figura 14 Un tramo de una señal sintética.

Tenemos algunas opciones más antes de terminar este módulo. La siguiente es la opción de tener la actividad de la señal generada representada por puntos (píxeles) en pantalla, se le pregunta al usuario si quiere ver el "raster" de la señal, éste tiene la finalidad de dar una idea global de cual es la actividad de las neuronas disparando. Aquí no es posible diferenciar las neuronas de las cuales provienen los disparos. Después existe la posibilidad de generar un archivo de salida que tenga el formato original de Neurored, con éste y el módulo de correlación XCOR, es posible analizar el efecto haber quitado los traslapes del archivo original de disparos, (entrada de UNE-VEC). En el Capítulo 3 se discute como se eliminaron esos traslapes y cual fue el efecto en la dependencia en el tiempo que presentaban originalmente las neuronas.

Por último, se tiene la opción de poder ver las formas de los potenciales de acción usados, superpuestos (Figura 15) en la misma pantalla para tener una forma directa de compararlos y poder distinguirlos una vez que en Discovery se hace una separación de ellos.

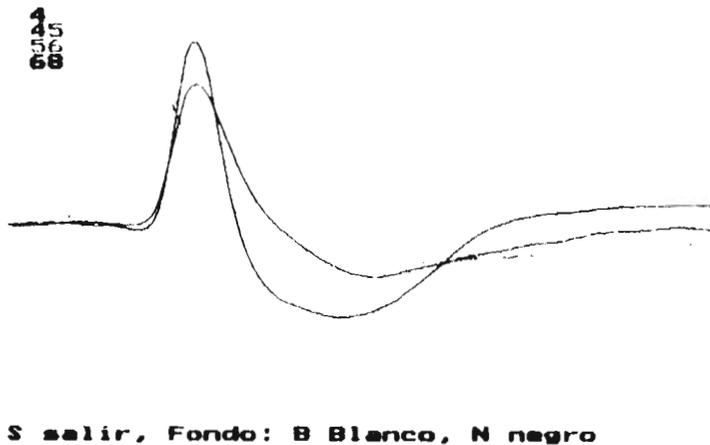


Figura 15 Superposición de las formas de onda usadas en una señal sintética.

Módulo de conversión D/A, DAC-ESP.

Una vez que la señal ha sido generada (UNE-VEC), ya estamos en condiciones de hacer análisis de correlación pero, nuestro interés reside en poder tener la traza en forma analógica y así usar este simulador como herramienta en la adquisición de señales reales. Así pues, este módulo sirve para obtener la traza de la señal sintética en su forma analógica. El módulo usa el DAC 1 de la tarjeta DT2821 (misma que utiliza Discovery para hacer adquisición A/D), el programa le pide al usuario el nombre del archivo que contiene a la señal sintética, y valida la entrada de tal forma que si el archivo requerido no existe le avisa y suspende la ejecución, de otra manera al indicarle el programa empieza a poner los datos en el DAC. La señal generada es grabada en cinta de vídeo usando la grabadora digital VR-100A para poder ser, fuera de línea, recuperada y analizada por Discovery.

Cada vector que contiene los puntos de una espiga es de 129 datos, por lo que para una traza de 30,000 milisegundos era necesario realizar muchas lecturas a disco duro o simplemente la memoria de datos asignada por Pascal era insuficiente para toda la traza. El dejar todo en disco duro provocaba retrasos en la salida al DAC, por lo que en el osciloscopio se veían saltos

no deseados, que cambiaban también, la relación temporal de la señal. Fue necesario usar apuntadores para utilizar óptimamente la memoria y no recurrir al uso del disco duro. Así los vectores de las formas son leídos y almacenados en una lista ligada unidireccional. Con esto quedó salvado el problema de los retrasos en el DAC.

Módulo de Correlación, XCOR.

Este módulo del programa sirve para calcular los histogramas de autocorrelación y correlación cruzada de dos series de tiempo. Las series de tiempo se dan dentro de un archivo de texto y la información está organizada según se muestra en la Figura 16 siguiendo el formato

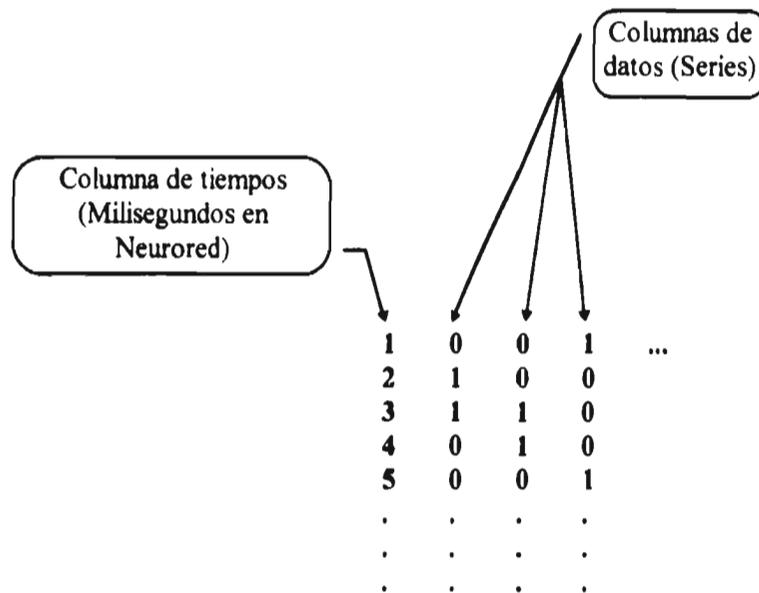


Figura 16 Formato para el archivo de entrada a XCOR.

de los archivos de salida del simulador Neurored

El número máximo de líneas en el archivo (milisegundos) es de 30000, esto debido a las limitantes del compilador Turbo Pascal 6.0, que no permite segmentos de datos mayores de 64 Kb, pero contamos también con una versión en C que corre bajo plataforma UNIX en una estación de trabajo Sparc 10.

Al iniciar la ejecución de este módulo se necesita proporcionar la información que se muestra en la Figura 17 y que más abajo se describe línea por línea.

- 1 Nombre del archivo de disparos : muestra1.sal
- 2 En el archivo muestra1.sal se tienen 6 columnas de datos.
- 3 Neuronas a correlacionar (NEU1,NEU2): 1 2
- 4 Espera estoy leyendo los datos...
- 5 Este archivo tiene 200 milisegundos.
- 6 Intervalo en milisegundos a usar [T1,T2]: 0 200
- 7 Ancho de la ventana de despliegue [A]: 200

Figura 17 Información necesaria para el módulo XCOR.

Inicialmente, y al igual que en el módulo UNE-VEC, se presenta la ruta de búsqueda para los archivos de entrada/salida, puede ser la misma dando solo ENTER o el usuario puede dar una ruta válida propia.

Línea 1: El archivo de disparos (entrada) debe estar en el mismo directorio que el XCOR, en caso contrario es necesario dar toda la ruta de búsqueda. Si el programa no puede localizar el archivo de entrada mandará un mensaje y terminará.

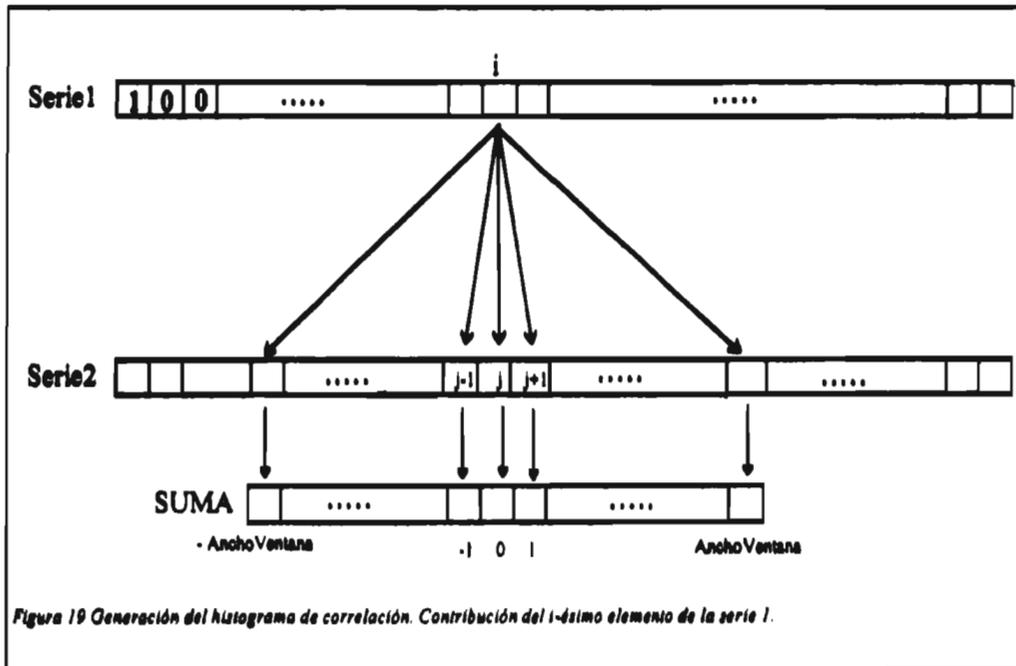
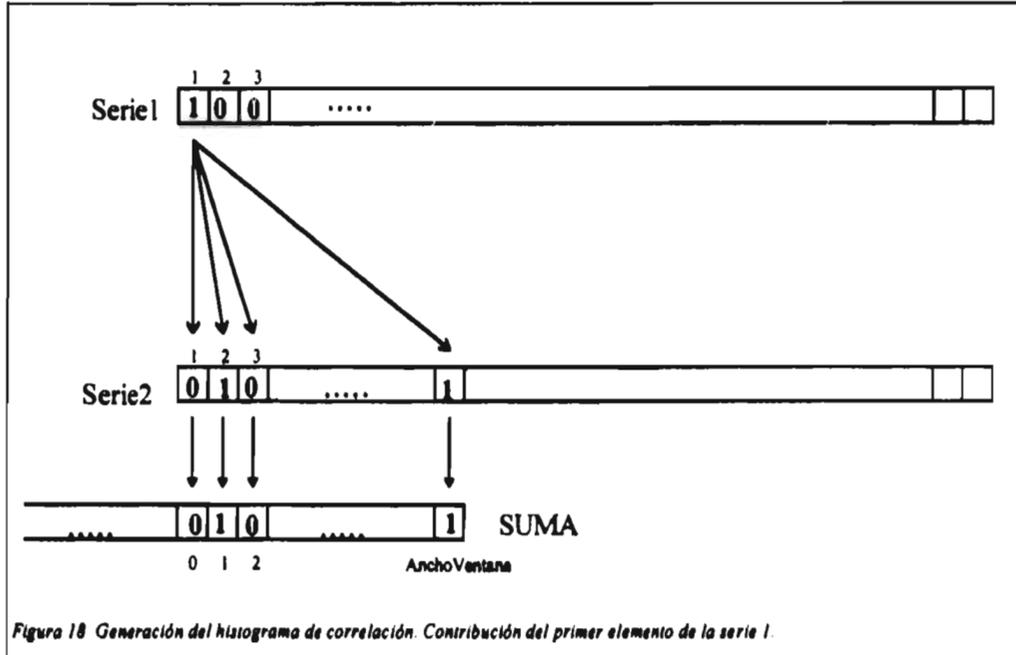
Línea 3: Las neuronas a correlacionar se refiere a las columnas de datos a usar, y se toman en el orden en el que se proporcionan. Se pueden dar los dos valores separados por un espacio o un ENTER para cada uno. Cabe hacer notar que en la línea 2 se presenta la información del número de columnas disponibles en el archivo de entrada que el programa obtiene automáticamente.

Línea 6: En esta línea se especifica el intervalo de la columna de tiempos que se va a utilizar en el análisis. Como Neurored comienza a numerar los milisegundos a partir de 1, el programa convierte a 1 una entrada cuyo extremo izquierdo fuese proporcionado como cero.

Línea 7: El ancho de ventana para el despliegue proporciona el intervalo que usará el programa para calcular el histograma y mostrarlo en su gráfica. Esta ventana se desplazará a lo largo de todo el intervalo de análisis (línea 6). Obviamente el ancho de ventana debe ser menor o igual que el intervalo de análisis.

Descripción del algoritmo de XCOR.

Supongamos dos series de tiempo: Serie 1, Serie 2. Cada elemento de la serie 1 que sea un 1 será tomado como referencia (como cero del histograma) y se tomarán distancias relativas a este dato, si en la serie 2 hubo disparos es decir 1's, entonces tenemos una contribución a una columna del histograma. Las contribuciones de cada elemento a cada distancia relativa, se van guardando en un vector llamado **SUMA** cuyo tamaño es [- AnchoVentana, AnchoVentana], y cuyo centro, como ya se mencionó arriba, es en cada iteración algún elemento distinto de cero de la serie 1. En la Figura 18 y en la Figura 19 se muestra la forma en la que se genera el histograma de correlación.



Al final de los cálculos el vector **SUMA** tiene los valores de las contribuciones hacia atrás y hacia adelante del elemento de referencia, obteniendo un histograma del estilo del presentado en la Figura 20.

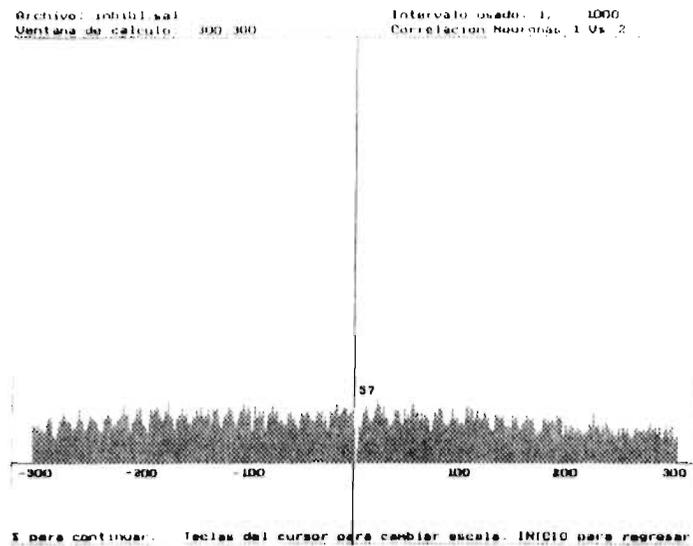


Figura 20 Ejemplo de correlación. En el recuadro superior aparece la información del archivo usado

En la gráfica que se muestra en la Figura 20 se puede hacer un ZOOM para ver en detalle alguna zona del histograma, este siempre se mantendrá centrado y es posible regresar a la resolución original presionando la tecla INICIO. Para continuar con la ejecución del programa se tiene que presionar la tecla "S". Si se está trabajando con autocorrelación entonces sólo se observará la parte derecha de la gráfica dado que ese histograma es siempre simétrico respecto al origen. La ventana de cálculo se puede escoger sobre cualquier valor siempre y cuando no se exceda la longitud de las series, pero en la práctica no se requieren ventanas muy grandes dado que no nos interesan análisis de más de 200 ms, porque en neuronas reales los tiempos de dependencia no llegan a ser tan grandes. En pantalla sólo tenemos capacidad, en este módulo, para ver los resultados en una ventana de ± 300 ms.

Al salir de la pantalla anterior, el programa pregunta si se quiere seguir trabajando con el mismo archivo de datos, si es así (opción S) entonces se vuelve a pedir el número de las columnas de neuronas a usar. Los datos tienen que ser nuevamente leídos lo único que se ahorra es el abrir nuevamente el archivo de datos.

En la Figura 21 se muestra el algoritmo en pseudocódigo para el cálculo del histograma.

Módulo XCOR

ENTRADAS: Archivo de datos (salida de Neurored), Intervalo a usar [**LimInf**, **LimSup**], Neuronas a correlacionar **Neu1**, **Neu2**, Ventana de despliegue **AnchoVentana**.

SALIDAS: Gráfico del **Histograma** de correlación, Vector de correlación **SUMA**, mensaje de error si el archivo de datos no existe.

Paso 1: Se leen las series **Serie1** y **Serie2** correspondientes a las columnas de las neuronas seleccionadas.

Paso 2: **Suma** ← 0

{ Se calcula la parte del histograma del lado positivo }

Paso 3: Para **i** desde **LimInf** hasta **LimSup** realiza

Paso 4: Si $i \leq \text{LimSup} - \text{AnchoVentana}$ entonces

Paso 5: Si **Serie1[i]=1** entonces

Paso 6: Para **j** desde **i** hasta **AnchoVentana** realiza

Paso 7: Si **Serie2[j]=1** entonces

Paso 8: Incrementa **Suma[j-i]**

Paso 9: Si no se cumple **paso 4** entonces

Paso 10: Si **Serie1[i]=1** entonces

Paso 6: Para **j** desde **i** hasta **LimSup** realiza

Paso 7: Si **Serie2[j]=1** entonces

Paso 8: Incrementa **Suma[j-i]**

{ Se calcula la parte del histograma del lado negativo }

Paso 9: Para **i** desde **LimSup** decrementando hasta **LimInf** realiza

Paso 10: Si $i > \text{LimSup} + \text{AnchoVentana}$ entonces

Paso 11: Si **Serie1[i]=1** entonces

Paso 12: Para **j** desde **i-1** decrementando hasta **i-AnchoVentana** realiza

Paso 13: Si **Serie2[j]=1** entonces

Paso 14: Incrementa **Suma[j-i]**

Paso 15: Si no se cumple **paso 9** entonces

Paso 16: Si **Serie1[i]=1** entonces

Paso 17: Para **j** desde **i-1** decrementando hasta **LimInf** realiza

Paso 18: Si **Serie2[j]=1** entonces

Paso 19: Incrementa **Suma[j-i]**

Paso 20: **FIN**

Figura 21 Pseudocódigo del módulo XCOR que sirve para obtener correlación cruzada.

Capítulo 3

Resultados

Planteamiento de las simulaciones.

Las simulaciones que a continuación se muestran corresponden a problemas típicos de conexión: Excitación, Inhibición, Independencia y Oscilación. La idea principal con estas simulaciones es la de presentar un panorama general del simulador GESSIN, evaluar su alcance y discutir sus limitaciones. El simulador puede ser usado para entrenar al investigador en la identificación del tipo de actividad que dos o más neuronas pueden tener al comunicarse entre sí. El poder controlar el tipo de conexión y los tiempos de disparos, permite estar plenamente seguros del tipo de histogramas de correlación que se deben obtener al analizar dichas conexiones.

Aunque GESSIN puede recibir como entrada cualquier archivo de tipo ASCII con el formato descrito en la página 25, en estas simulaciones usamos las salidas del simulador Neurored [24], porque con él tenemos archivos de actividad simulada de larga duración y perfectamente controlados. Neurored también cuenta con un módulo para obtener histogramas de correlación cruzada llamado Correlac, pero está limitado a 500 milisegundos de análisis, XCOR de GESSIN, en cambio tiene capacidad de análisis para 30,000 milisegundos. Otra limitante de Correlac es que no puede obtener la autocorrelación de una neurona.

El mecanismo que usamos para las simulaciones implica el uso de tres programas de correlación: Correlac, XCOR y Correlación con Discovery. Los dos primeros utilizan como entrada archivos de Neurored mientras que el tercero toma los datos de archivos adquiridos de una señal en forma analógica.

Inicialmente se diseña un circuito con Neurored, con un número de neuronas que no exceda de 10, aunque hacerlo con este máximo, implica una pérdida muy grande de información por los traslapes (decimos que ocurre un traslape cuando dos o más neuronas presentan actividad en el mismo milisegundo), con lo que quizá se haga imposible el recuperar la información de la conectividad. En esta tesis se trabajó con un número máximo de 5 neuronas y la pérdida más grande de información en los disparos de una neurona fue de alrededor de un 40%, pero en todos los casos se recuperó la conectividad del circuito. Más adelante en este mismo capítulo se muestra la relación en el número de disparos que tienen cada una de las neuronas en la simulación de cinco neuronas y como se modifican estas relaciones una vez que los archivos son procesados primero por GESSIN y luego por Discovery. Este circuito en Neurored genera un archivo de salida en formato ASCII. El programa UNE-VEC toma este archivo y con los disparos que hay en él genera un vector que contiene el tipo de espigas que corresponde a la neurona que haya disparado en ese milisegundo. El barrido del archivo se hace por renglones y se toma de cada uno de ellos la neurona que haya disparado, si hay varias que disparen en el mismo renglón, entonces se le da prioridad a la que no haya presentado disparo recientemente, con esto se le está dando la misma oportunidad de disparo a todas las neuronas. En caso de sólo usar dos, en los traslapes los disparos se van alternando.

Para una mejor descripción del funcionamiento de UNE-VEC ir a la página 25. El resultado final de ejecutar este módulo, es un archivo en donde tenemos la traza de una señal sintética con información mezclada de varias neuronas con una relación conocida de disparos en el tiempo dada por el simulador Neurored. Adicionalmente a esta salida tenemos otro archivo opcional de salida, que tiene el formato de Neurored, este tiene el fin de poder usar Correlac para analizar el archivo original modificado en sus traslapes por UNE-VEC. De esta manera podemos medir la pérdida de información por traslapes. En este punto tenemos la posibilidad de hacer análisis de correlación sobre el archivo de salida con formato de Neurored usando XCOR de GESSIN o Correlac de Neurored. Aquí todavía todo es digital.

El archivo de salida de UNE-VEC pasa a ser de entrada para DAC-ESP que se encarga de llevar la traza de su forma digital a su forma analógica usando el DAC No. 1 de la tarjeta DT2821 (ver Apéndice B). Fue necesario calibrar esta salida y medir la duración de cada espiga para tener controlada la señal de salida analógica. Para medir los tiempos de duración de cada espiga, se usaron archivos de disparos consecutivos, primero con separación cero entre cada una después uno, dos, tres y cuatro milisegundos. En la Figura 22 y en la Figura 23 se muestran los primeros 240 milisegundos de la traza de las señales usadas para calibrar, primero tenemos espigas sin separación y en seguida con separación de un milisegundo, los archivos de dos, tres

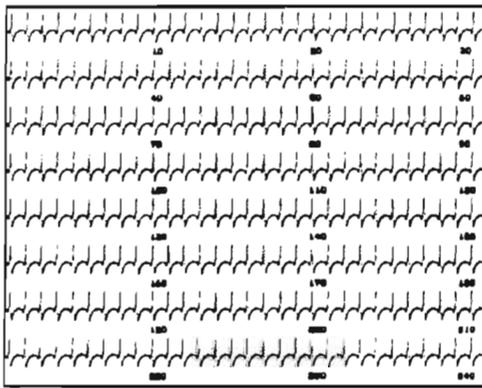


Figura 22 Archivo para calibración. Espigas con distancia 0 ms.

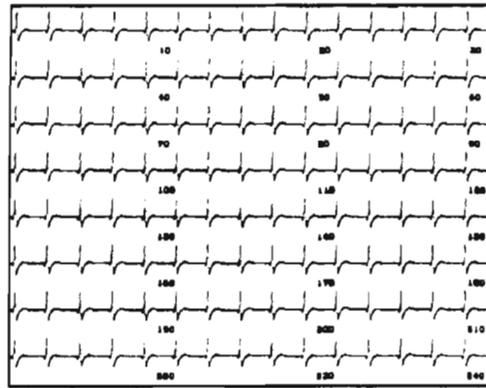


Figura 23 Archivo para calibración. Espigas con distancia 1 ms.

y cuatro milisegundos de separación son similares a los aquí mostrados. Cada disparo mostrado en estas figuras corresponden a un renglón de los archivos de Neurored, es decir, un “un milisegundo”, aunque ya en el DAC la duración de cada espiga, medida a partir de ellos, fue de 1.98 ms lo cual es más realista.

Una vez puesta la señal en el DAC, se pasa a la grabadora digital VR100A para grabarla en una cinta de videocasete y posteriormente analizarla.

A partir de aquí se usa el software comercial Discovery con el cual volvemos a convertir la traza analógica a su forma digital y volvemos a realizar el análisis de correlación pero ahora usando la herramienta con la que cuenta ese paquete. A continuación se muestra un esquema en el que podemos ver cuál es la ruta seguida en el proceso descrito anteriormente.

Los pasos seguidos en cada simulación se pueden enumerar así:

Paso 1: Generar traza (UNE-VEC).

Paso 2: Analizar archivo original y el modificado por GESSIN.

Paso 3: Convertir traza digital a la traza analógica y grabarla a cinta.

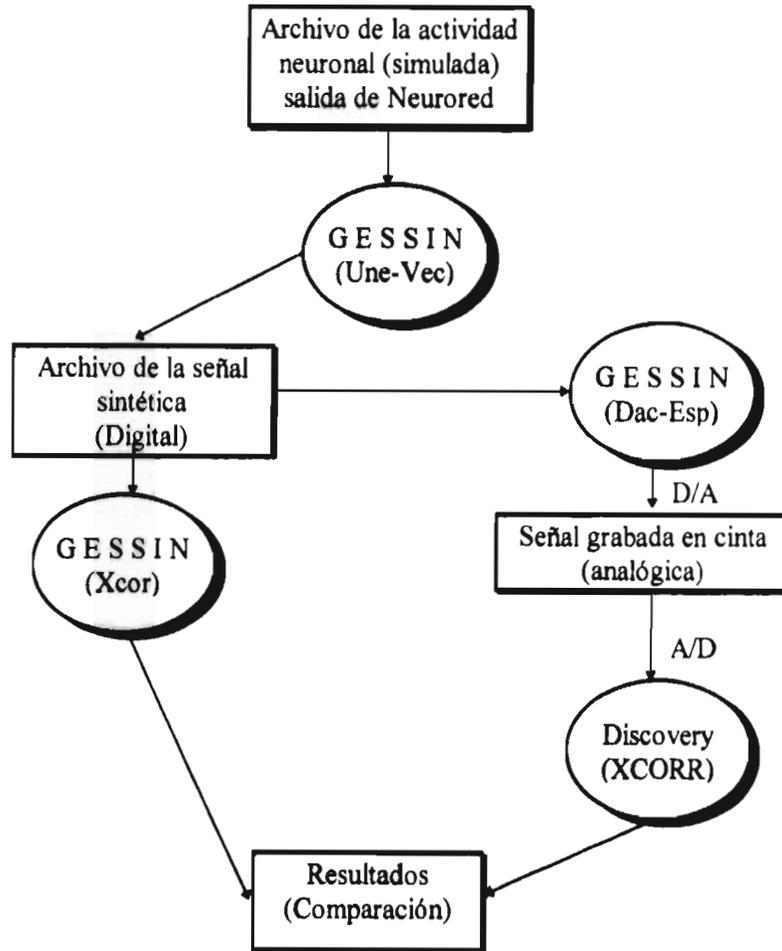


Figura 24 Esquema de tareas para las simulaciones.

Paso 4: Adquirir la traza analógica con Discovery.

Paso 5: Analizar la traza con herramienta de Discovery.

Paso 6: Comparar resultados.

En la Figura 24 se esquematizan los pasos anteriormente listados, y que se aplicaron a cada una de las simulaciones que a continuación se presentan.

Ejemplo de excitación.

En neuronas reales, la excitación es la actividad que se puede detectar con más facilidad que las otras, y en las simulaciones no fue diferente. Sin embargo, cuando se usan muchas neuronas se debe tener mucho cuidado dado que, debido a la pérdida de información por los traslapes, los histogramas de correlación cruzada pierden cuentas y por lo tanto el pico usado para interpretar excitación puede perderse en la actividad de fondo. Sin embargo, lo que pudimos constatar en esta simulación es que se necesitaría tener una pérdida de más del 60% para perder completamente la actividad original, pero en el peor de los casos tuvimos una pérdida del 40%. A partir de la Figura 28 y hasta la Figura 36 se muestran los resultados de

simulación concerniente a la excitación entre dos neuronas conectadas como se muestra en la Figura 25. Aquí la neurona 1 excita a la neurona 2 y las fibras 3 y 4 alimentan a las neuronas, respectivamente. Este esquema de conexión es el que utiliza Neurored para simular la red neuronal. Así pues, en el archivo de disparos tenemos 4 columnas que corresponden, respectivamente a la neurona 1, neurona 2, a la fibra 3 y la última a la fibra 4.

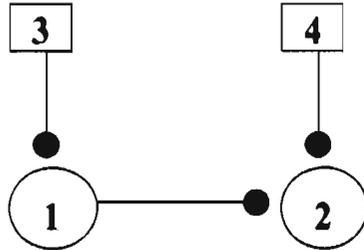


Figura 25 Diagrama de conexionamiento para las neuronas usadas para la simulación de excitación.

El uso de Neurored para poder obtener un archivo de salida para ser usado en GESSIN, se puede consultar en la tesis [24]. Sin embargo, y como ya había mencionado antes, el origen de este archivo no es trascendente dado que cualquier archivo que cumpla con el formato del capítulo 2 podría ser usado.

A continuación se describen cada una de las simulaciones y se presentan los resultados obtenidos gráficamente, tratando con esto, de ser lo más explícito posible.

Paso 1: Al generar la traza usando UNE-VEC, un análisis sobre los archivos usados usando un programa en Pascal, arroja los siguientes resultados:

Nombre del archivo a revisar: excit4.sal
1 0 0 0
Este archivo tiene 4 columnas de datos
Este archivo tiene 5000 líneas de datos
Datos por columna:
Columna # 1= 1049
Columna # 2= 2159
Columna # 3= 739
Columna # 4= 522

Nombre del archivo a revisar: excit4.nrd
1 0 0
Este archivo tiene 2 columnas de datos
Este archivo tiene 5000 líneas de datos
Datos por columna:
Columna # 1= 733
Columna # 2= 1843

Tabla 2 Comparación de los archivos de disparos antes y después de GESSIN.

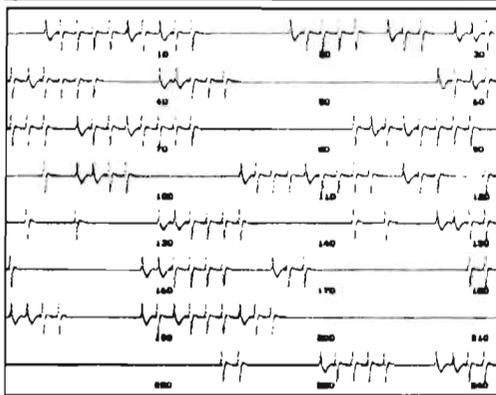
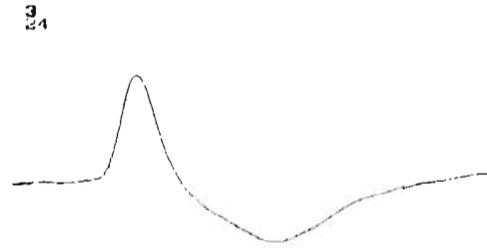


Figura 26 Excitación. Un tramo de la señal sintética para dos neuronas.



S salir, Fondo: B Blanco, N negro

Figura 27 Excitación. Formas de onda usadas para el ejemplo de excitación.

En esa tabla se puede ver que la neurona uno (columna uno), tuvo una pérdida del 30% mientras que la neurona dos perdió tan sólo el 15% de la información inicial. Como no nos interesa la evolución temporal de las fibras, las columnas correspondientes a ellas desaparecen del archivo modificado por GESSIN. Por convención, las extensiones de los nombres de los archivos las usamos para diferenciar el tipo de archivo que estamos usando en cada paso, por ejemplo, SAL se utiliza para los archivos de salida de Neurored mientras que NRD son los archivos de salida de GESSIN con el formato de Neurored. La extensión SEN es usada para los archivos que contienen la información correspondiente a la traza de la señal sintética. Para este ejemplo usé las formas de onda 3 y 24 que se muestran, traslapadas en la Figura 27, y un tramo de la señal, todavía en forma digital se puede ver en la Figura 26.

En la Figura 28 podemos ver la correlación cruzada entre las dos neuronas de la simulación, antes de que la información fuese procesada por GESSIN, el pico del histograma se encuentra a la derecha de la referencia cero, lo cual se puede interpretar como una dependencia temporal excitadora, en este caso la neurona numerada como uno sería la que tomada como serie 1 en análisis de correlación, provoca que la neurona 2, tomada como serie 2, responda a

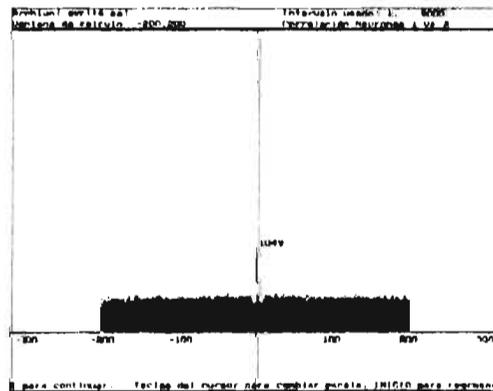


Figura 28 Excitación. Correlación cruzada entre las dos neuronas del archivo EXCIT4.SAL. Antes de GESSIN.

los disparos de uno. En las Figura 29 y Figura 30 se muestran respectivamente las autocorrelaciones de las neuronas 1 y 2 y ambas se observan con un comportamiento normal, no oscilatorio.

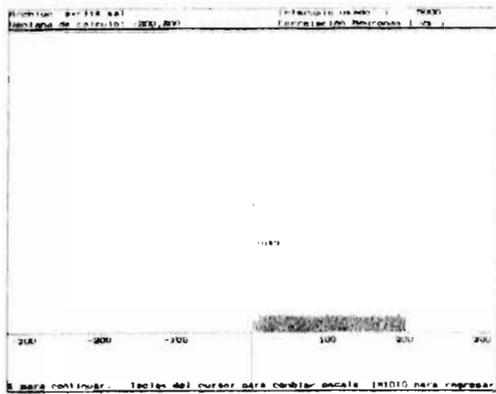


Figura 29 Excitación. Autocorrelación de la neurona 1. Antes de GESSIN

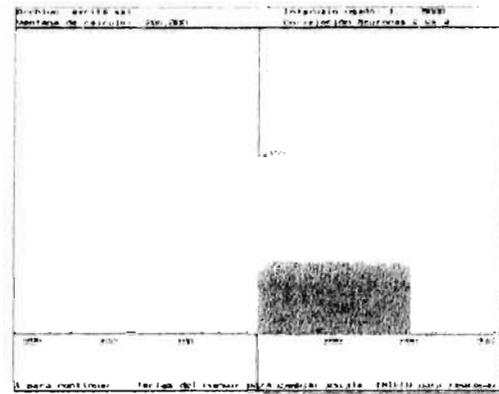


Figura 30 Excitación. Autocorrelación de la neurona 2. Antes de GESSIN.

Una vez obtenido el archivo de la señal sintética tenemos también, en forma opcional, un archivo de esa señal con el formato de Neurored, con el cual podemos hacer nuevamente correlación cruzada para ver el efecto de la pérdida de información por traslapes en la relación

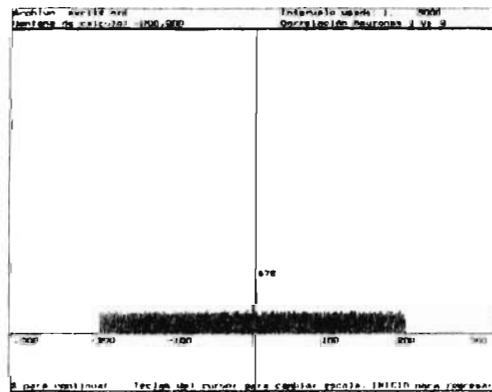


Figura 31 Excitación. Correlación cruzada entre las dos neuronas del archivo EXCIT4.NRD. Después de GESSIN.

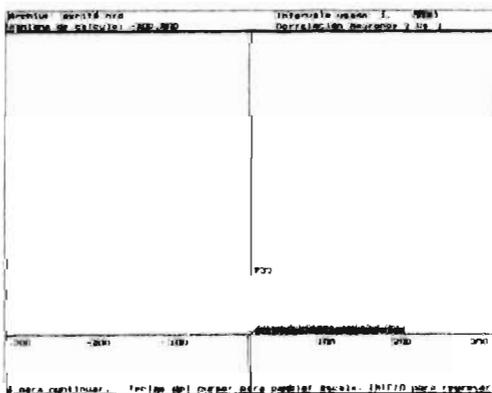


Figura 32 Excitación. Autocorrelación de la neurona 1. Después de GESSIN.

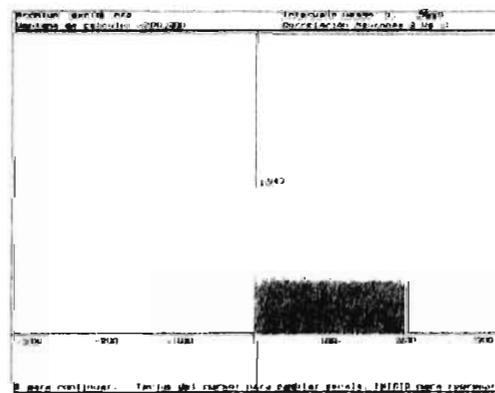


Figura 33 Excitación. Autocorrelación de la neurona 2. Después de GESSIN.

temporal de dichas neuronas. En las Figura 31, 32 y 33 tenemos la correlación cruzada y las autocorrelaciones después de haber usado GESSIN.

Como se puede apreciar de la Figura 28 a la Figura 33, la pérdida de información sólo provocó que las columnas del histograma bajaran, es decir, sólo disminuyeron el número de ocurrencias, pero la forma de las gráficas no se modificaron, lo cual implica que no se ve afectada su relación de disparos en el tiempo.

Pero este análisis no es suficiente, necesitamos asegurar que esa dependencia se siga manteniendo aún cuando la señal sea pasada a su forma analógica y nuevamente recuperada a su forma digital usando el paquete comercial Discovery.

Paso 3: Al usar el módulo DAC-ESP de GESSIN, tenemos disponible la señal en forma analógica para grabarla en cinta como se describió en los antecedentes. Este programa al terminar, le informa al usuario acerca del número de milisegundos de Neurored, y cuanto tiempo tardó sacando la traza por el DAC. Discovery al adquirir debe de ser congruente con esta información, y efectivamente lo fue. Para esta simulación DAC-ESP reporta haber encontrado 5000 milisegundos de Neurored con 2576 espigas y un tiempo real de señal de 9.89 segundos.

Paso 4: Al reproducir y adquirir la señal con Discovery, nuestros datos son perfectamente reproducibles, tenemos que Discovery detecta y separa 2576 espigas



Figura 34 Excitación. Edición de cúmulos para la separación de las espigas

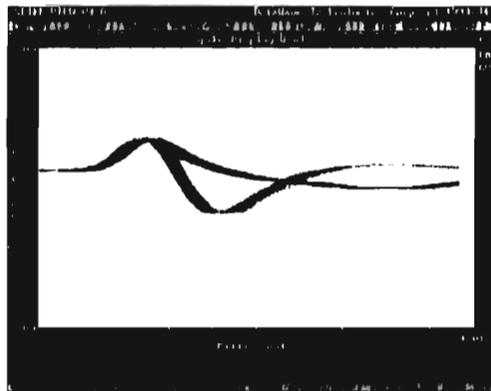


Figura 35 Excitación. Formas de onda de las espigas separadas, (comparar con las originales en la Figura 27)

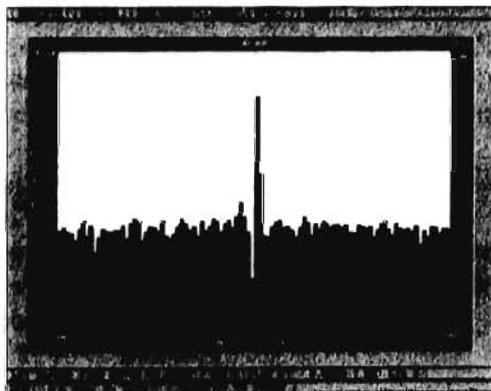


Figura 36 Excitación. Correlación cruzada para las neuronas del archivo EXCIT4.UFF y representadas por los cúmulos de la Figura 34.

exactamente el número original, en tiempo de 10 segundos. Esto fue totalmente verificado en todas las simulaciones siguientes. Por lo tanto, podemos decir que nuestro dispositivo experimental está funcionando, ahora falta hacer el análisis de correlación para compararlo con lo ya antes discutido. En este punto los formatos de los archivos ya no son compatibles, y los tiempos han cambiado de significado, por lo tanto no estamos en posibilidades de usar XCOR nuevamente, así que no tenemos más que usar la herramienta propia de Discovery para la correlación cruzada.

El histograma de correlación cruzada obtenido con Discovery se presenta en la Figura 36, en él se puede observar el mismo tipo de pico obtenido en los otros dos histogramas calculados antes y después de haber usado GESSIN. Aquí se observa un hueco en el origen cuya pendiente es infinita, es decir que ocurre sólo para un instante. Después de haber hecho algunas pruebas con GESSIN llegamos a la conclusión de que esos huecos eran debidos a la pérdida de información por los traslapes, cabe hacer mención de que ese tipo de comportamiento lo tuvimos sistemáticamente, por lo que podemos decir que no incluye información nueva respecto a la relación temporal entre las neuronas.

Ejemplo de inhibición.

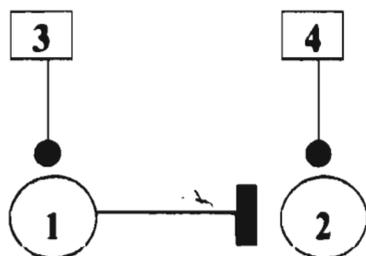


Figura 37 Diagrama de conexionamiento para las neuronas usadas para la simulación de inhibición.

Al igual que en el ejemplo anterior, aquí tenemos dos neuronas conectadas de tal forma que su relación temporal sea la de inhibición, en la Figura 37 podemos ver el diagrama de conexión. Aquí la neurona 1 inhibe a la neurona 2 (formas de onda como en la Figura 38), las fibras 3 y 4, alimentan a las neuronas, sin que eso modifique la dependencia en el tiempo entre las neuronas. El mecanismo de análisis se repite y a continuación muestro, los resultados gráficos de él. Cabe hacer notar que la inhibición es un estado de conectividad que se

detecta en señales reales, con más dificultad que los otros y en las simulación no fue diferente. Se tuvieron algunos problemas, sobre todo en cuanto al ancho de la ventana de análisis, si se toma todo el tiempo de la señal, la inhibición se oculta, así que tuve que tomar un tramo menor que en términos de la señal total, el tramo usado fue del orden de un 20% del total.

Paso 1 : Al usar UNE-VEC, tenemos una pérdida de tan sólo el 3.5% de la información de la neurona 1 y del 13% de la neurona 2, como se puede ver en la Tabla 3, esto es una pérdida relativamente pequeña considerando que podríamos perder hasta un 40 o 50% sin

Nombre del archivo a revisar: inhbel.sal

1 0 0 1 0

Este archivo tiene 4 columnas de datos

Este archivo tiene 5000 líneas de datos

Datos por columna:

Columna # 1= 2047

Columna # 2= 565

Columna # 3= 1275

Columna # 4= 979

Nombre del archivo a revisar: inhbel.nrd

1 0 0

Este archivo tiene 2 columnas de datos

Este archivo tiene 5000 líneas de datos

Datos por columna:

Columna # 1= 1975

Columna # 2= 492

Tabla 3 Comparación de los archivos de disparos antes y después de GESSIN

exactamente el número original, en tiempo de 10 segundos. Esto fue totalmente verificado en todas las simulaciones siguientes. Por lo tanto, podemos decir que nuestro dispositivo experimental está funcionando, ahora falta hacer el análisis de correlación para compararlo con lo ya antes discutido. En este punto los formatos de los archivos ya no son compatibles, y los tiempos han cambiado de significado, por lo tanto no estamos en posibilidades de usar XCOR nuevamente, así que no tenemos más que usar la herramienta propia de Discovery para la correlación cruzada.

El histograma de correlación cruzada obtenido con Discovery se presenta en la Figura 36, en él se puede observar el mismo tipo de pico obtenido en los otros dos histogramas calculados antes y después de haber usado GESSIN. Aquí se observa un hueco en el origen cuya pendiente es infinita, es decir que ocurre sólo para un instante. Después de haber hecho algunas pruebas con GESSIN llegamos a la conclusión de que esos huecos eran debidos a la pérdida de información por los traslapes, cabe hacer mención de que ese tipo de comportamiento lo tuvimos sistemáticamente, por lo que podemos decir que no incluye información nueva respecto a la relación temporal entre las neuronas.

Ejemplo de inhibición.

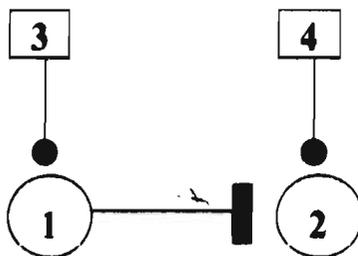


Figura 37 Diagrama de conexionamiento para las neuronas usadas para la simulación de inhibición.

Al igual que en el ejemplo anterior, aquí tenemos dos neuronas conectadas de tal forma que su relación temporal sea la de inhibición, en la Figura 37 podemos ver el diagrama de conexión. Aquí la neurona 1 inhibe a la neurona 2 (formas de onda como en la Figura 38), las fibras 3 y 4, alimentan a las neuronas, sin que eso modifique la dependencia en el tiempo entre las neuronas. El mecanismo de análisis se repite y a continuación muestro, los resultados gráficos de él. Cabe hacer notar que la inhibición es un estado de conectividad que se

detecta en señales reales, con más dificultad que los otros y en las simulación no fue diferente. Se tuvieron algunos problemas, sobre todo en cuanto al ancho de la ventana de análisis, si se toma todo el tiempo de la señal, la inhibición se oculta, así que tuve que tomar un tramo menor que en términos de la señal total, el tramo usado fue del orden de un 20% del total.

Paso 1 : Al usar UNE-VEC, tenemos una pérdida de tan sólo el 3.5% de la información de la neurona 1 y del 13% del la neurona 2, como se puede ver en la Tabla 3, esto es una pérdida relativamente pequeña considerando que podríamos perder hasta un 40 o 50% sin

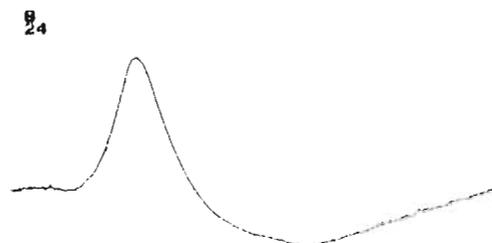
Nombre del archivo a revisar: inhibel.sal
1 0 0 1 0
Este archivo tiene 4 columnas de datos
Este archivo tiene 5000 líneas de datos
Datos por columna:
Columna # 1= 2047
Columna # 2= 565
Columna # 3= 1275
Columna # 4= 979

Nombre del archivo a revisar: inhibel.nrd
1 0 0
Este archivo tiene 2 columnas de datos
Este archivo tiene 5000 líneas de datos
Datos por columna:
Columna # 1= 1975
Columna # 2= 492

Tabla 3 Comparación de los archivos de disparos antes y después de GESSIN

perder la dependencia en el tiempo de dichas neuronas.

Paso 2 : En las Figura 39, Figura 40 y Figura 41 se presenta el análisis con XCOR



S salir, Fondo: B Blanco, N negro

Figura 38 Inhibición. Formas de onda usadas para el ejemplo de Inhibición.

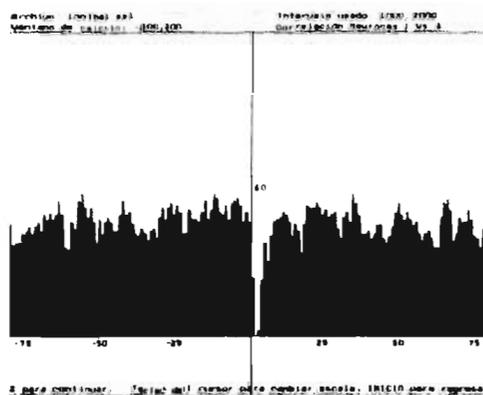


Figura 39 Inhibición Correlación cruzada entre las dos neuronas del archivo INHIBI.SAL Antes de GESSIN

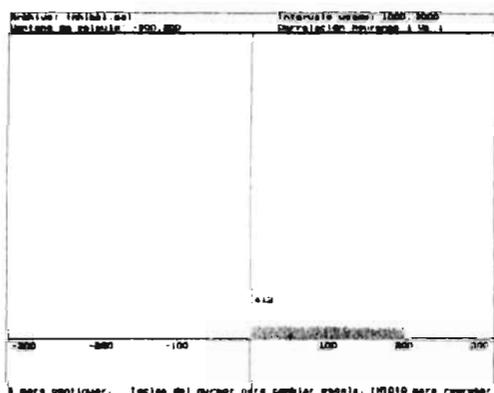


Figura 40 Inhibición. Autocorrelación de la neurona 1. Antes de GESSIN.

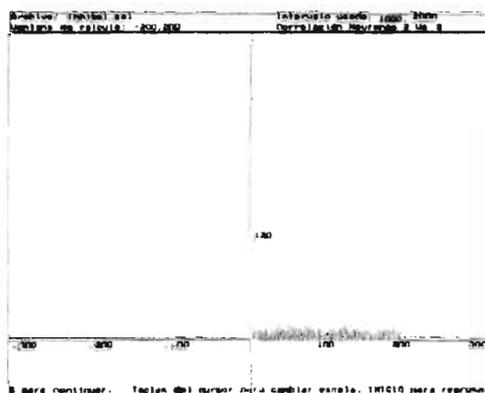


Figura 41 Inhibición. Autocorrelación de la neurona 2. Antes de GESSIN.

antes de haber usado GESSIN. En el histograma de correlación (Figura 39) se puede ver un hueco a la derecha del origen, lo cual denota un comportamiento sistemático alrededor de dos o tres milisegundos. Como el hueco se encuentra a la derecha podemos inferir que la neurona 1 inhibe a la 2 porque la neurona 1 corresponde a la serie 1 y la neurona 2 a la serie 2, por lo que un disparo de la neurona 1 se refleja como la ausencia de disparos de la neurona 2 alrededor de unos dos o tres milisegundos después.

En las Figura 42, Figura 43 y Figura 44 se muestra el mismo análisis que el descrito en el párrafo anterior pero aquí se hace después de haber procesado la señal con GESSIN.

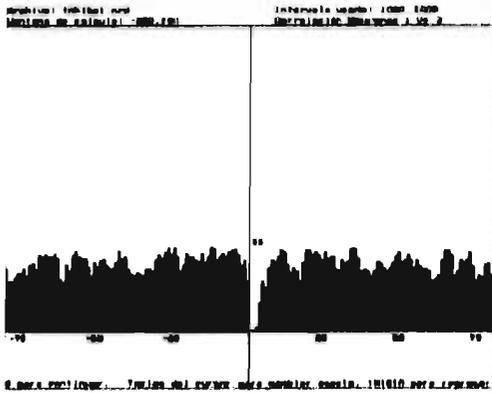


Figura 42 Inhibición. Correlación cruzada entre las dos neuronas del archivo INIBIBI.NRD. Después de GESSIN

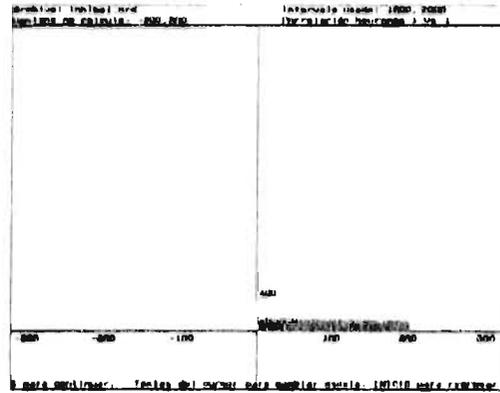


Figura 43 Inhibición. Autocorrelación de la neurona 1. Después de GESSIN.

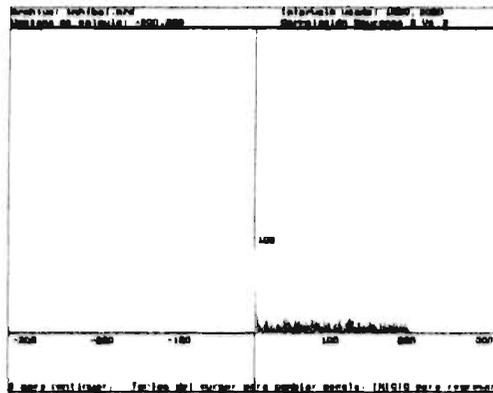


Figura 44 Inhibición. Autocorrelación de la neurona 2. Después de GESSIN.

Paso 3 : El uso del programa DAC-ESP para la conversión de la señal a su forma analógica reporta al final un total de 5000 milisegundos de Neurored con 2467 espigas en un tiempo total de 9.94 segundos. Se graba a cinta para su posterior captura y análisis con Discovery.

Paso 4 : Al adquirir la señal usando Discovery, este logra detectar y separar exactamente todas las espigas de la traza generada por GESSIN, 2467, en un tiempo total de 10 segundos. En la Figura 45, podemos ver la edición de cúmulos para separar las espigas y en la Figura 46 se aprecian las formas de onda de las espigas utilizadas en la simulación. Como se puede ver en esas figuras, las separación lograda es la idealmente esperada.

Paso 5 : El análisis con el histogramas de correlación cruzada se presenta en la Figura 47 en la cual se puede apreciar un hueco igual al obtenido en el análisis con XCOR, y alrededor de los mismo tiempos, la autocorrelación de la neurona 1 se puede apreciar en la Figura 48.

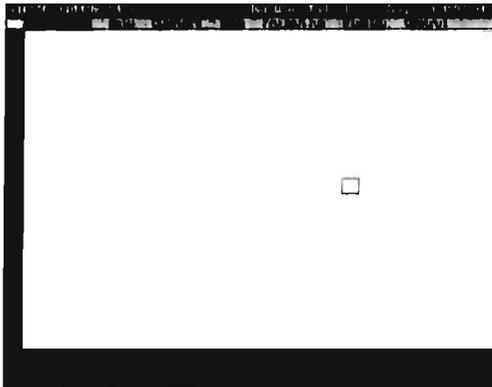


Figura 45 Inhibición. Edición de cúmulos para la separación de las espigas.

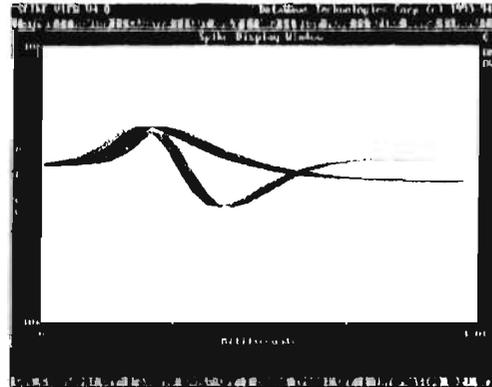


Figura 46 Inhibición. Formas de onda de las espigas separadas

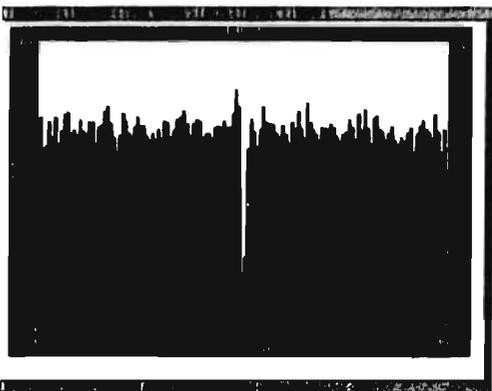


Figura 47 Inhibición. Correlación cruzada para las neuronas del archivo INHIBEL.UFF

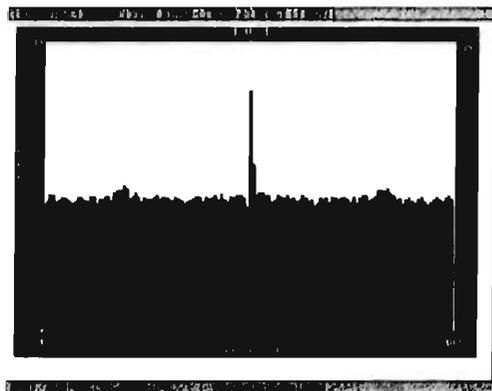


Figura 48 Inhibición. Autocorrelación de la neurona 1 del archivo INHIBEL.UFF

Ejemplo de independencia.

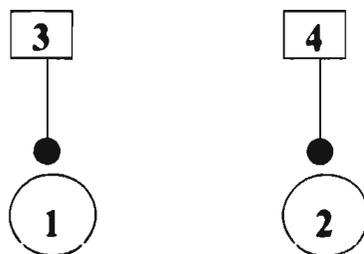


Figura 49 Diagrama de conexonamiento para las neuronas usadas para la simulación de independencia.

A igual que en los dos ejemplos anteriores aquí se usa un circuito de dos neuronas pero que disparan independientemente y no tienen ningún tipo de conexión. En la Figura 49 podemos ver de que manera las neuronas van a disparar. las fibras 3 y 4 proporcionan la actividad para los disparos y ambas neuronas disparan independientemente. El análisis de este ejemplo es igual que los anteriores por lo que me remitiré a presentar los resultados gráficos y al final de ellos haré una breve discusión de ellos, siguiendo como siempre los cinco pasos descritos en

el último párrafo de la página 36, llegamos a obtener los histogramas de correlación y autocorrelación para realizar la comparación de resultados. En la Figura 39, Figura 40 y Figura 41, tenemos los histogramas de la señal antes de ser procesada por GESSIN; en la Figura 42, Figura 43 y Figura 44 los histogramas después de haber usado GESSIN; en la Figura 47 y Figura 48 los histogramas obtenidos con Discovery.

Nombre del archivo a revisar: indep2.sal
 1 0 0 1
 Este archivo tiene 4 columnas de datos
 Este archivo tiene 5000 líneas de datos
 Datos por columna:
 Columna # 1= 1501
 Columna # 2= 1491
 Columna # 3= 1272
 Columna # 4= 1260

Nombre del archivo a revisar: indep2.nrd
 1 0 0
 Este archivo tiene 2 columnas de datos
 Este archivo tiene 5000 líneas de datos
 Datos por columna:
 Columna # 1= 1252
 Columna # 2= 1242

Tabla 4 Comparación de los archivos de disparos antes y después de GESSIN

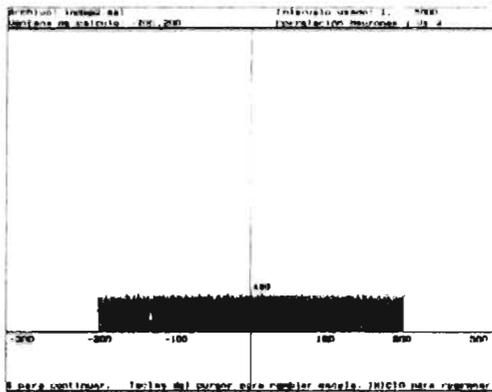


Figura 50 Independencia. Correlación cruzada entre las dos neuronas del archivo INDEP2.SAL. Antes de GESSIN

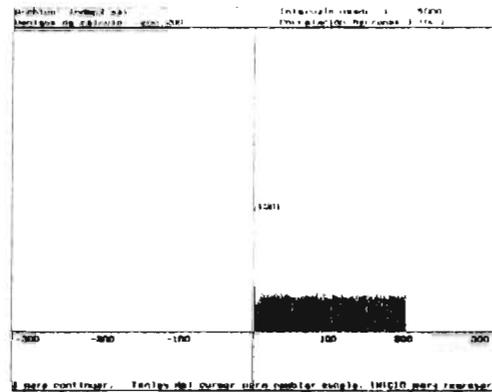


Figura 51 Independencia. Autocorrelación de la neurona 1. Antes de GESSIN.

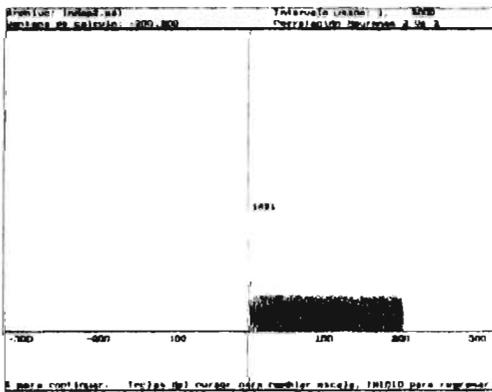


Figura 52 Independencia. Autocorrelación de la neurona 2. Antes de GESSIN.

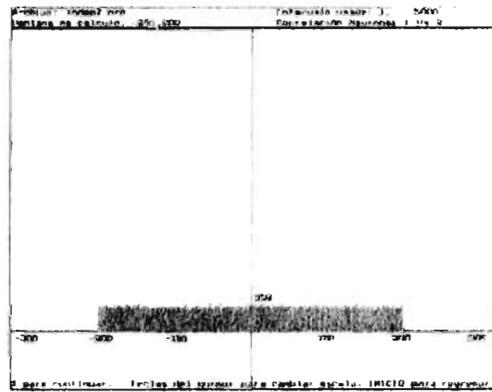


Figura 53 Independencia. Correlación cruzada entre las dos neuronas del archivo INDEP2.NRD. Después de GESSIN

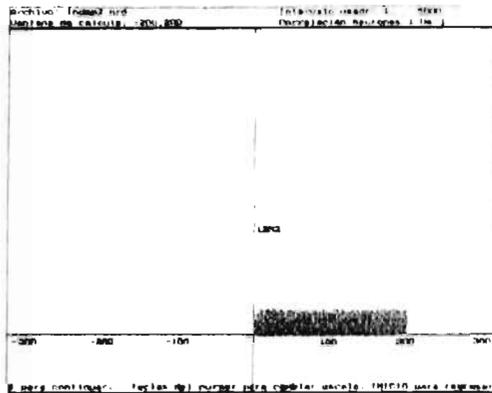


Figura 54 Independencia. Autocorrelación de la neurona 1 Después de GESSIN.

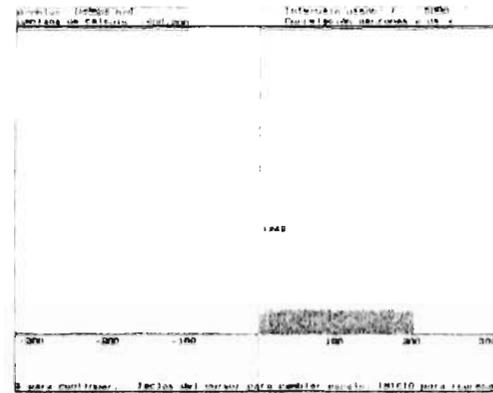


Figura 55 Independencia Autocorrelación de la neurona 2 Después de GESSIN.

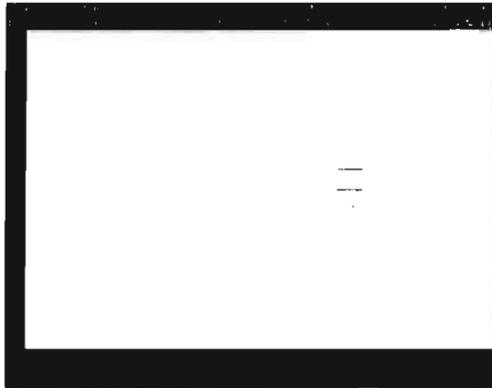


Figura 56 Independencia Edición de cúmulos para la separación de las espigas.

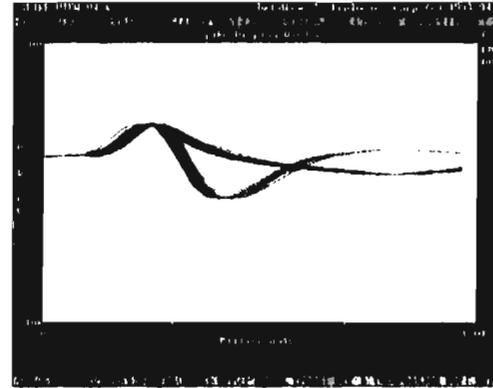


Figura 57 Independencia Formas de onda de las espigas separadas.



Figura 58 Independencia. Correlación cruzada para las neuronas del archivo INDEP2.UFF.

Paso 1 : En la Tabla 4 vemos la información proporcionada por UNE-VEC, aquí la pérdida de información fue del 16.5% para la neurona 1 y del 17% para neurona 2, dado que no hay conexión ambas neuronas pierden aproximadamente el mismo número de disparos.

Paso 2 : Comparando las gráficas de correlación y autocorrelación antes y después de haber usado GESSIN, vemos exactamente la misma dependencia en el tiempo entre neuronas antes y después del procesamiento.

Paso 3 : Al adquirir la señal con Discovery se recuperan totalmente en número original de espigas y las formas de onda adquiridas se pueden ver en la Figura 57.

Paso 5 : El histograma de correlación cruzada entre la neurona y dos, calculado con Discovery se presenta en la Figura 58. Aquí aparece un pico pero de altura relativa muy pequeña por lo que no puede ser tomado como excitación. La diferencia con uno de los picos que se ve a la izquierda en la misma gráfica, es muy pequeña, este comportamiento difiere de los otros histogramas porque al quitar los traslapes se crea una relación espuria que es fácilmente detectada por la altura de las columnas en el histograma.

Ejemplo de oscilación.

En la Figura 59 tenemos el circuito utilizado para la simulación de la oscilación, aquí en realidad tenemos dos circuitos que debido al funcionamiento interno de Neurored son necesarios para lograr el efecto deseado de la oscilación, ambos circuitos responden igual como

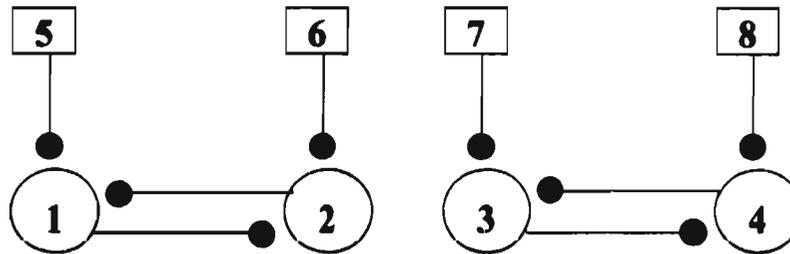


Figura 59 Diagrama de conexionamiento para las neuronas usadas para la simulación de oscilación.

se podrá constatar más adelante durante el análisis. La neurona 1 excita a la 2 y viceversa y lo mismo sucede para las neuronas 3 y 4, pero el circuito 1-2 es independiente del circuito 3-4, las fibras, como siempre, sólo provocan que las neuronas tengan actividad.

Paso 1 : La generación de la traza arroja los resultados presentados en la Tabla 5, las

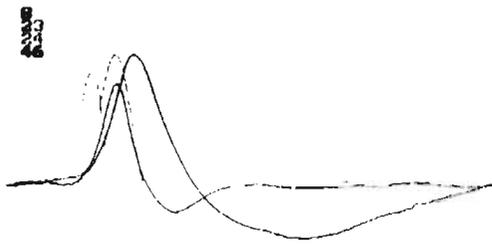
Nombre del archivo a revisar: oscil.sal
1 0 0 0 0 0 1 0 1
Este archivo tiene 8 columnas de datos
Este archivo tiene 30000 líneas de datos
Datos por columna:
Columna # 1= 5076
Columna # 2= 5036
Columna # 3= 5076
Columna # 4= 5036
Columna # 5= 8982
Columna # 6= 8887
Columna # 7= 8982
Columna # 8= 8887

Nombre del archivo a revisar: oscil.nrd
1 0 0 0 0
Este archivo tiene 4 columnas de datos
Este archivo tiene 30000 líneas de datos
Datos por columna:
Columna # 1= 2593
Columna # 2= 2539
Columna # 3= 2313
Columna # 4= 2266

Tabla 5 Oscilación. Comparación de los archivos de disparos antes y después de GESSIN

pérdidas de información de las neuronas 1,2,3 y 4 son respectivamente del 49%, 49%, 54% y 55%. La disminución en los disparos de cada neurona es grande, pero como se verá más adelante los resultados obtenidos no difieren de los esperados. El porcentaje necesario para perder la dependencia en el tiempo entre neuronas conectadas, es de más del 60%, calculado a partir de varias simulaciones.

Paso 2 : Presentaré los resultados de las correlaciones de uno sólo de los circuitos que conforman el ejemplo de oscilación, dado que ambos (1-2 y 3-4) arrojan exactamente los



8 salir, Fondo: B Blanco, N negro

Figura 60 Oscilación. Formas de onda usadas para el ejemplo de oscilación

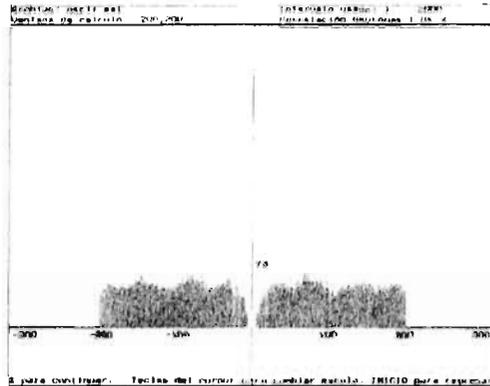


Figura 61 Oscilación. Correlación cruzada entre las neuronas 1 y 2 del archivo OSCII.SAL. Antes de GESSIN

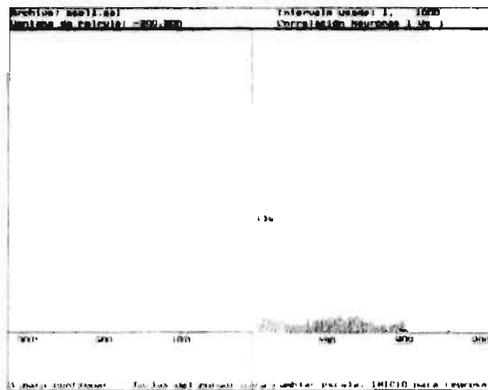


Figura 62 Oscilación. Autocorrelación de la neurona 1. Antes de GESSIN.

mismos histogramas. En la Figura 61 vemos la correlación cruzada para las neuronas uno y dos. Vemos que cerca del origen no hay actividad, lo cual querría decir que las neuronas empiezan a responder pasados algunos milisegundos, pero además vemos unas ondulaciones simétricas respecto al origen, lo cual nos indica que la segunda neurona está respondiendo en forma oscilatoria. La Figura 62 muestra la autocorrelación de la neurona 1, sólo presento ésta porque las otras tres son iguales, aquí podemos ver que la actividad de todas las neuronas del circuito también presentan un comportamiento oscilatorio.

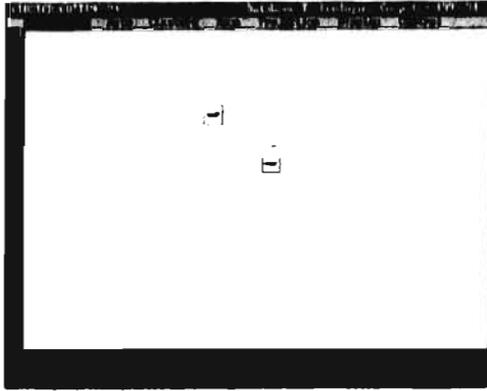


Figura 63 Oscilación. Edición de cúmulos para la separación de las espigas

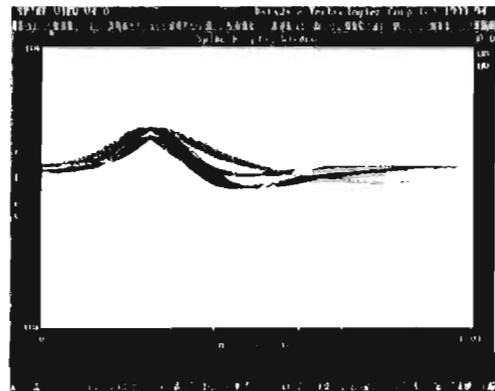


Figura 64 Oscilación. Formas de onda de las espigas separadas

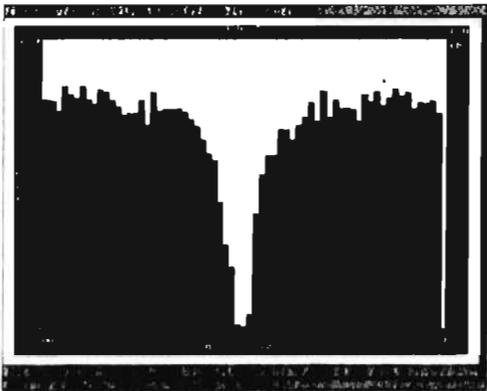


Figura 65 Oscilación. Correlación cruzada para las neuronas 1 Vs 2 del archivo OSC11.UFF (ventana -400, 400 ms)

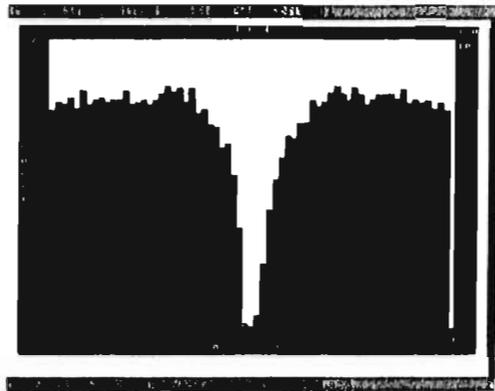


Figura 66 Oscilación. Correlación cruzada para las neuronas 3 Vs 4 del archivo OSC11.UFF (ventana -400, 400 ms)

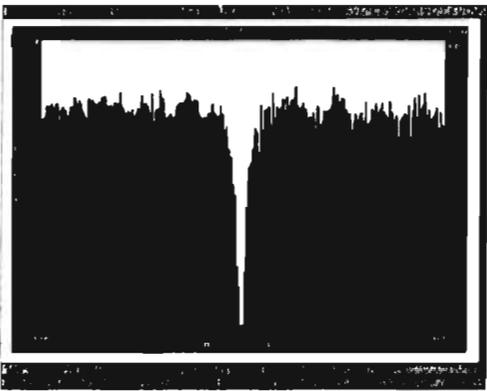


Figura 67 Oscilación. Correlación cruzada para las neuronas 1 Vs 2 del archivo OSC11.UFF (ventana -600, 600 ms)

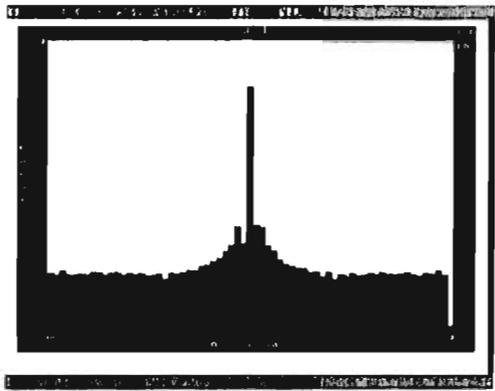


Figura 68 Oscilación. Autocorrelación neurona 1

Paso 3 : Al llevar la traza a su forma analógica, el programa DAC-ESP, reporta haber generado 30,000 milisegundos de Neurored con 9711 espigas y un tiempo real de señal de 59.38 segundos.

Paso 4 :La separación de las espigas adquiridas por Discovery, no presentó mayor problema, el haber escogido formas de onda para cada neurona, de manera de tener diferencias importantes, hace que en la edición de cúmulos, los cúmulos aparezcan perfectamente definidos (Figura 63), y por lo tanto, la separación se pueda llevar a efecto sin ningún contratiempo, (Figura 64). Discovery logra detectar 9710 espigas, la diferencia fue de sólo una espiga no detectada, y un tiempo total de 60 segundos.

Paso 5 : La oscilación presentó algunos problemas para poder ser vista desde los histogramas de correlación obtenidos con Discovery. La pérdida de información debida a los traslapes es la causante de que los efectos de la oscilación se diluyan un poco cuando la señal es pasada a su forma analógica y después otra vez a su forma digital. Observando el primer histograma de correlación de Discovery (Figura 65 y Figura 66) vemos algo similar a lo obtenido antes de procesar la señal, pero las subidas y bajadas, aunque todavía se pueden ver, no son del todo definidas. Ampliando un poco el ancho de ventana de análisis (Figura 67) la oscilación se puede apreciar un poco mejor al definirse más los máximos y mínimos. En la Figura 68 muestro la autocorrelación de la neurona 1, pero las demás neuronas tienen un histograma similar. Aquí vemos que por efectos de la oscilación la base de la espiga central se amplía un poco. Aún con todo podemos decir que los resultados de esta simulación son aceptables.

Ejemplo de un circuito con 5 neuronas.

A continuación se presenta una simulación en la que se usó un circuito de 5 neuronas conectadas entre sí de acuerdo al diagrama de la Figura 69. Las neuronas 1 y 2 tienen entrada

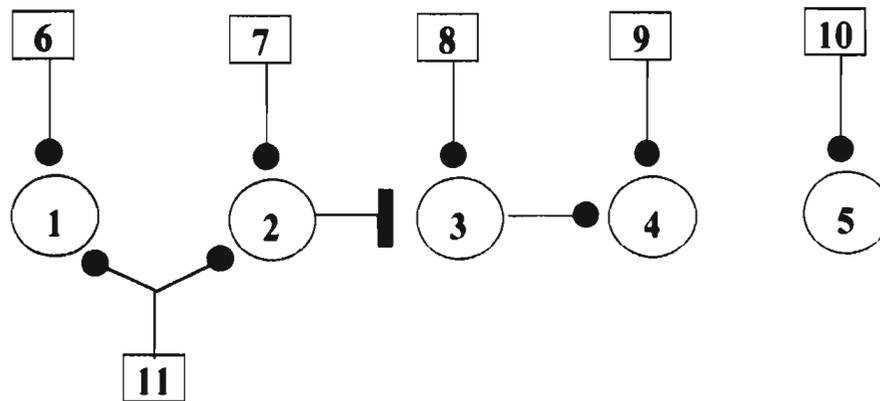


Figura 69 Diagrama de conexiónamiento de una red con cinco neuronas

compartida (fibra 11) pero no están conectadas entre sí; la neurona 2 inhibe a la 3, ésta excita a la 4 y la 4 y 5 no están conectadas. La idea con un circuito así es analizar todas las combinaciones posibles entre las neuronas para tratar de identificar, en caso de no conocerse a priori, la conectividad entre ellas. En este ejemplo presento los resultados como si no hubiese conocido de antemano la conexión.

Paso 1 : Al generar la señal se obtiene la Tabla 6, en donde se muestran las pérdidas de información por traslapes. Respectivamente los porcentajes de pérdida para las neuronas 1,2,3,4 y 5 son respectivamente 59%, 61%, 51%, 57% y 62%. UNE-VEC reporta al final de su

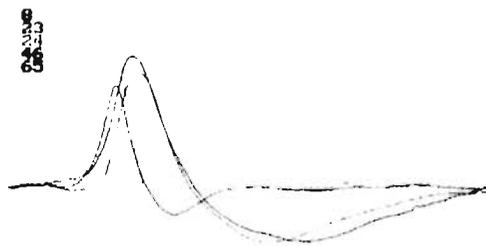
Nombre del archivo a revisar: red52f.sal
1 0 0 0 0 0 0 0 1 0 0
Este archivo tiene 11 columnas de datos
Este archivo tiene 30000 líneas de datos
Datos por columna:
Columna # 1= 19400
Columna # 2= 20873
Columna # 3= 5296
Columna # 4= 15598
Columna # 5= 7664
Columna # 6= 5942
Columna # 7= 9089
Columna # 8= 10479
Columna # 9= 10465
Columna #10= 5945
Columna #11= 6087

Nombre del archivo a revisar: red52f.nrd
1 0 0 0 0 0
Este archivo tiene 5 columnas de datos
Este archivo tiene 30000 líneas de datos
Datos por columna:
Columna # 1= 7769
Columna # 2= 8134
Columna # 3= 2613
Columna # 4= 6636
Columna # 5= 2907

Tabla 6 (5 Neuronas) Comparación de los archivos de disparos antes y después de GESSIN

ejecución una señal de 30,000 milisegundos de Neurored con 28,059 espigas. Para este ejemplo se utilizaron las formas 8, 23, 24, 46 y 65 que tienen características que permitieron una rápida separación en Discovery (Figura 70).

Paso 2 : Los histogramas de correlación entre las parejas de neuronas 1-2, 1-3, 1-4, 1-5, 2-3, 2-4, 2-5, 3-4, 3-5 y 4-5 antes (figuras de la columna izquierda) y después (figuras de la columna derecha) del procesamiento con GESSIN se presentan en el siguiente conjunto de



S salir, Fondo: B Blanco, N negro

Figura 70 (5 Neuronas) Formas de onda usadas para el ejemplo de 5 neuronas.

figuras (Figura 71 a la Figura 90).

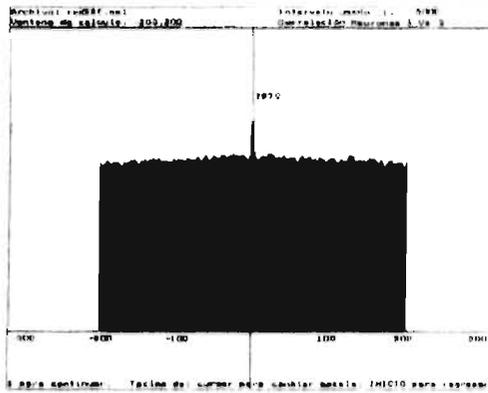


Figura 71 (5 Neuronas) Correlación cruzada 1 Vs 2. Antes de usar GESSIN.

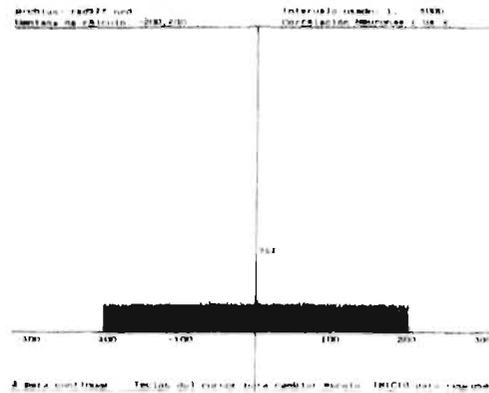


Figura 72 (5 Neuronas) Correlación cruzada 1 Vs 2 Después de usar GESSIN.

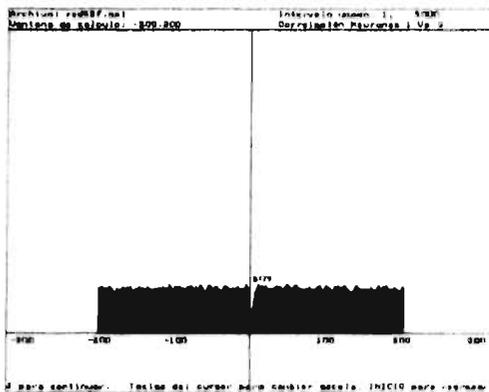


Figura 73 (5 Neuronas) Correlación cruzada 1 Vs 3 Antes de usar GESSIN.

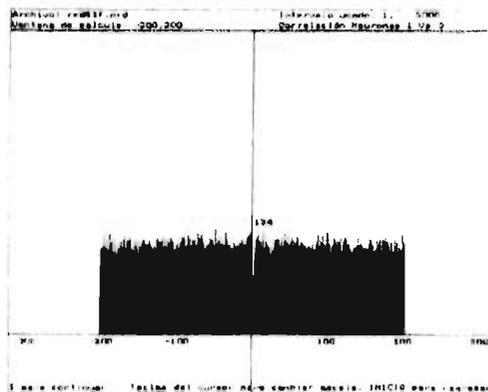


Figura 74 (5 Neuronas) Correlación cruzada 1 Vs 3 Después de usar GESSIN.

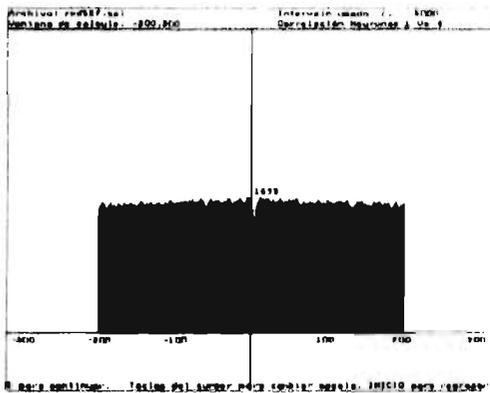


Figura 75 (5 Neuronas) Correlación cruzada 1 Vs 4. Antes de usar GESSIN.

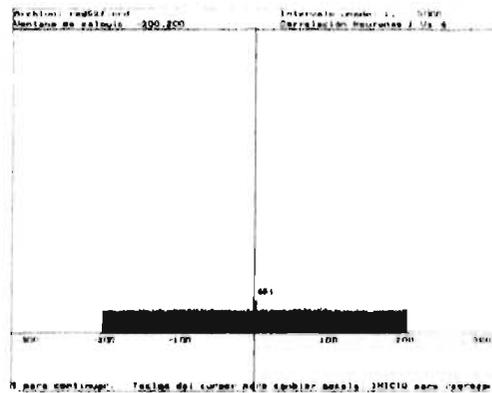


Figura 76 (5 Neuronas) Correlación cruzada 1 Vs 4. Después de usar GESSIN.

Como se muestra en el diagrama de conexión, la neurona 1 no tiene conexión con la 2 pero si tienen entradas compartidas. Dado que ambas neuronas están recibiendo actividad de la misma fibra, es de esperar entonces que en el histograma aparezca un pico centrado en el origen y que crece simétricamente desde la base. Viendo la Figura 71 con un poco de cuidado se puede apreciar ese comportamiento. En la Figura 72 aparece el pico correspondiente pero

la base es más angosta y se ve a la derecha del origen, lo que puede cambiar la inferencia de conectividad.

La Figura 73 tiene el histograma 1 Vs 3, la neurona 1 tiene dos fibras mientras que la neurona 3 sólo tiene una, esto puede provocar que la neurona 1 dispare más frecuentemente que la tres y parezca como si la 1 inhibiera a la 3. El histograma después de GESSIN (Figura 74) muestra, aunque en diferente escala, exactamente el mismo comportamiento.

Las neuronas 1 y 4 tampoco presentan ninguna conexión, pero la presencia de inhibición intermedia se refleja como un pequeño hueco a la derecha del origen (Figura 75). Después de GESSIN se ve una espiga a la derecha del origen, por lo que se altera la inferencia de conectividad.

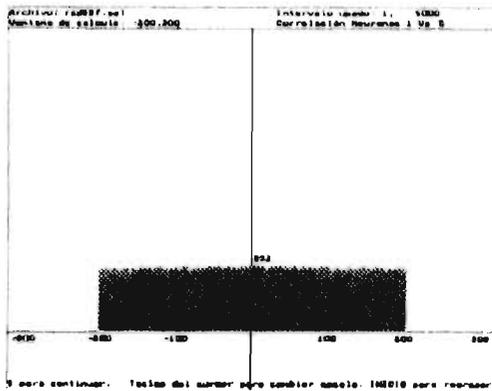


Figura 77 (5 Neuronas) Correlación cruzada 1 Vs 5. Antes de usar GESSIN.

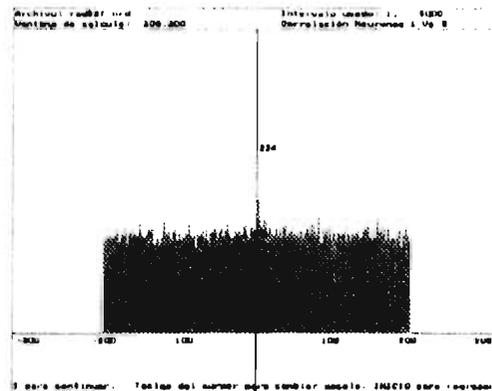


Figura 78 (5 Neuronas) Correlación cruzada 1 Vs 5. Después de usar GESSIN.

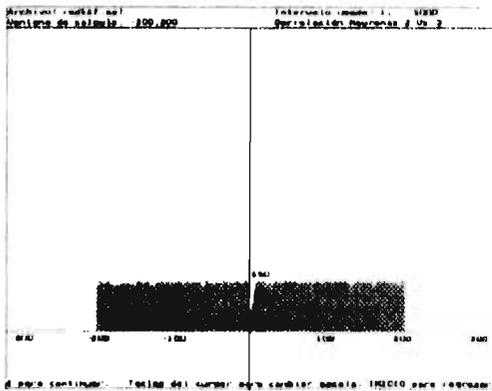


Figura 79 (5 Neuronas) Correlación cruzada 2 Vs 3. Antes de usar GESSIN.

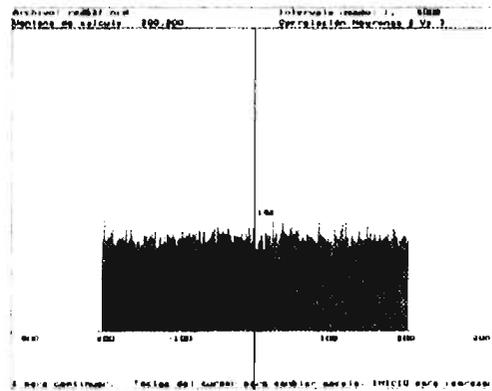


Figura 80 (5 Neuronas) Correlación cruzada 2 Vs 3. Después de usar GESSIN.

El par 1-5 presenta un histograma (Figura 77) perfectamente plano en análisis antes de GESSIN, sin embargo después de haber utilizado GESSIN (Figura 78), aparece un pico centrado en el origen, pero que comparado con los otros picos circundantes, no es significativo. Es un caso similar al que ocurrió con el par 1-2, nada más que aquí aparece una columna en blanco dentro del pico, esto se debe a la pérdida de información por los traslapes.

El par 2-3, de acuerdo al diagrama de conexionamiento, presenta inhibición, como vemos en el histograma antes de GESSIN, el hueco aparece suficientemente marcado para representar una inhibición (Figura 79). Vemos que en el histograma después de GESSIN (Figura 80), se pierde el hueco y ya no se puede detectar la inhibición.

El histograma del par 2-4 también presenta un pequeño hueco desplazado a la derecha del origen (Figura 81), igual al que ha aparecido en los pares en los que intervienen las neuronas 1 y 2. Pero en el histograma después de GESSIN (Figura 82), vemos que aparece con un pico de altura muy pequeña, pero significativa. Entonces la inferencia de conectividad se modifica.

En el histograma de la Figura 83 podemos apreciar una total independencia entre las

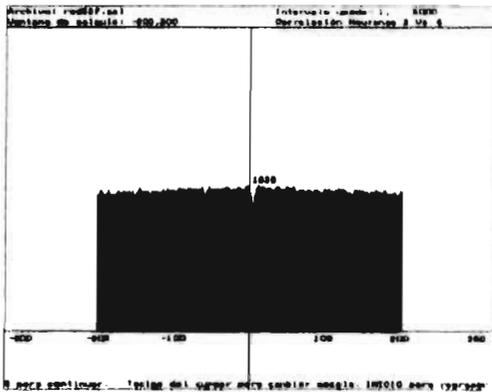


Figura 81 (5 Neuronas) Correlación cruzada 2 Vs 4. Antes de usar GESSIN.

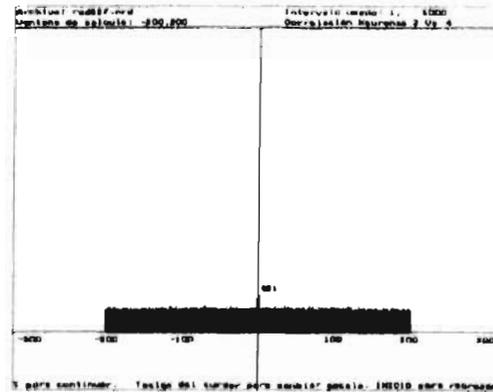


Figura 82 (5 Neuronas) Correlación cruzada 2 Vs 4. Después de usar GESSIN.

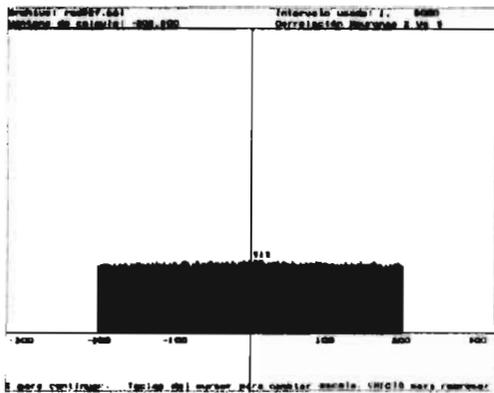


Figura 83 (5 Neuronas) Correlación cruzada 2 Vs 5. Antes de usar GESSIN.

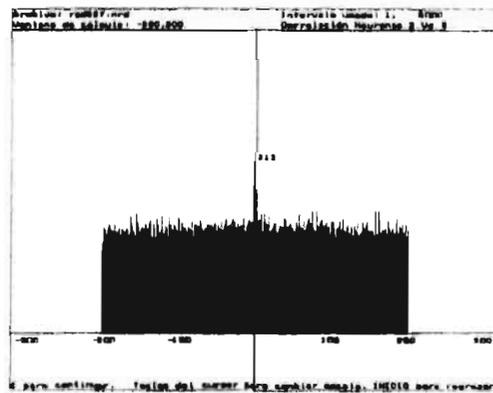


Figura 84 (5 Neuronas) Correlación cruzada 2 Vs 5. Después de usar GESSIN.

neuronas que forman el par 2-5. Sin embargo en el análisis después de GESSIN, vemos un pico centrado en el origen y simétrico respecto a él (Figura 84). Aquí también se modifica la inferencia de conectividad.

Las neuronas 3 y 4 están conectadas mediante una excitación, al estudiar el primer histograma (Figura 85), antes de GESSIN, vemos un pico que se encuentra ligeramente desplazado a la derecha del origen, lo cual denota la excitación de la cual se habla. En la Figura 86 se observa el mismo pico, después de haber usado GESSIN. La diferencia en la altura se debe a que los histogramas, están presentados a escalas diferentes, ver el número arriba de cada pico que es el máximo alcanzado por él. Aquí la inferencia de conectividad es correcta en

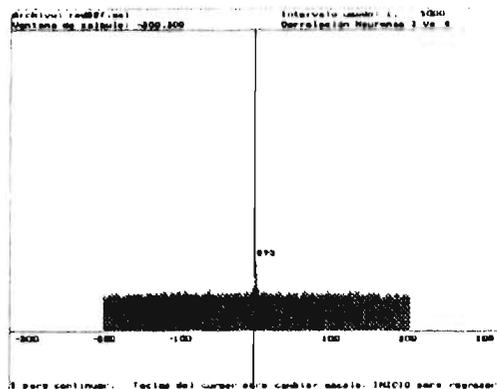


Figura 85 (5 Neuronas) Correlación cruzada 3 Vs 4. Antes de usar GESSIN.

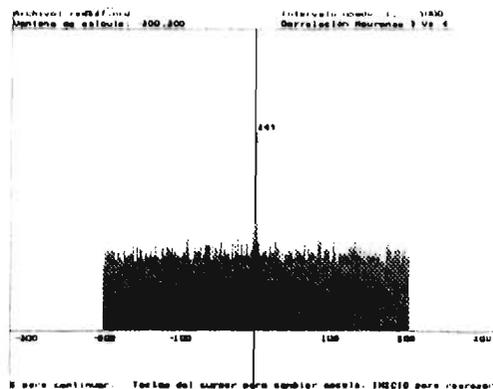


Figura 86 (5 Neuronas) Correlación cruzada 3 Vs 4. Después de usar GESSIN.

ambos casos.

Con el par 3-5 no tenemos ningún tipo de entrada compartida ni conexión alguna por la cual, tanto antes (Figura 87) como después (Figura 88) de GESSIN, tenemos histogramas planos que indican independencia entre esas neuronas.

Entre 4 y 5 tampoco hay conexión y el primer histograma se presenta plano como debe ser (Figura 89), pero el segundo tiene una espiga a la derecha del origen (Figura 90). Aquí se modifica la inferencia de conectividad.

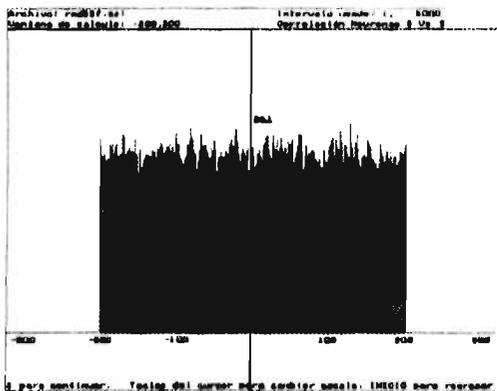


Figura 87 (5 Neuronas) Correlación cruzada 3 Vs 5. Antes de usar GESSIN.

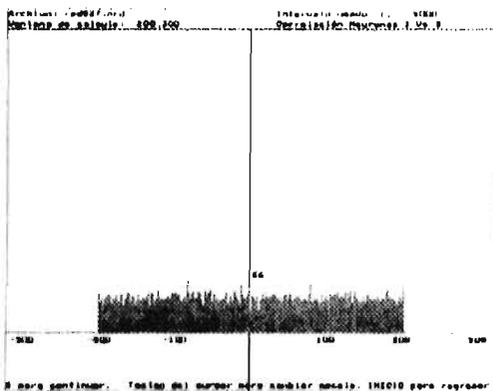


Figura 88 (5 Neuronas) Correlación cruzada 3 Vs 5. Después de usar GESSIN.

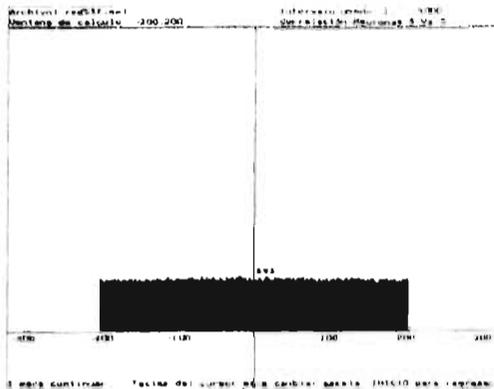


Figura 89 (5 Neuronas) Correlación cruzada 4 Vs 5. Antes de usar GESSIN.

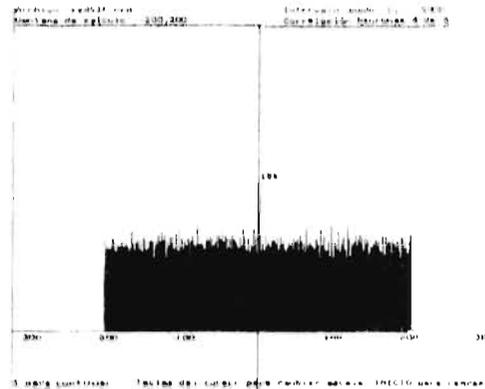


Figura 90 (5 Neuronas) Correlación cruzada 4 Vs 5 Después de usar GESSIN.

Paso 3 : Al convertir la traza de la señal a su forma analógica tenemos que DAC-ESP reporta al final haber procesado 30,000 milisegundos de Neurored con un total de 28059 espigas y un tiempo real de señal de 59.54 segundos.

Paso 4 : Al adquirir la señal con Discovery, este reporta haber separado las 28059 espigas del archivo original en tiempo total de 60 segundos. La edición utilizada para la separación de los cúmulos se muestra en la Figura 91 y las formas de onda de los grupos separados en la Figura 92.



Figura 91 (5 Neuronas) Edición de cúmulos para la separación de espigas, archivo RED32F.UFF

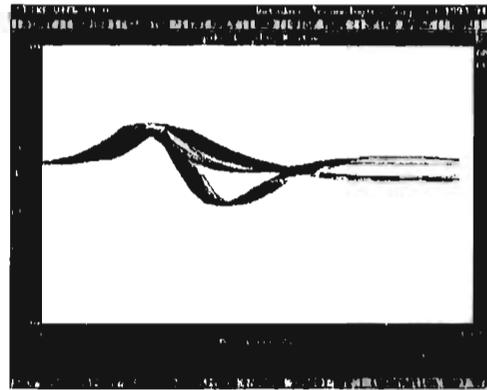


Figura 92 (5 Neuronas) Formas de onda de las espigas separadas. Archivo RED32F.UFF.

Paso 5 : A continuación se presenta el análisis con histogramas de correlación cruzada usando la herramienta de Discovery. Es importante hacer notar que al momento de convertir la señal a su forma analógica se pierde la correspondencia con los nombres asociados a cada neurona desde el principio, es decir, la neurona 1 puede ser vista por Discovery desde la señal analógica con cualquier otro número, sin embargo podemos establecer una correspondencia entre los números inicialmente asignados a cada neurona y los que, después de la adquisición, les ha asignado Discovery. Esto lo podemos lograr comparando la Figura 70 con la Figura 92. La forma de onda asignada a cada neurona determina de que neurona se trata, así pues basta con identificar su forma de onda para identificarla. Comparando tenemos las siguientes correspondencias antes y después de Discovery : 1 -> 3, 2 -> 4, 3 ->5, 4 -> 1 y 5 -> 2. A lo largo de la descripción de las histogramas de Discovery, seguiré usando la nomenclatura inicial

pero el lector debe tener cuidado con los encabezados de las gráficas dado que estos, por lo tanto, no corresponderán al texto.

En la Figura 93 tenemos la correlación cruzada de la neurona 1 contra la 2 en su nomenclatura original. Ahí podemos apreciar un pico central que denota una entrada compartida, lo cual concuerda con lo obtenido antes de la conversión. En el histograma de XCOR para el par 1-3 vimos (Figura 74) un hueco desplazado a la derecha mismo que se presenta nuevamente en la Figura 94, la forma cambia porque lo estamos viendo a diferente

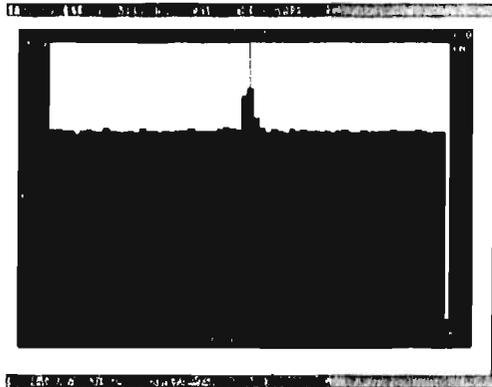


Figura 93 (5 Neuronas) Correlación cruzada 3 vs 4 correspondiente a la 1 Vs 2 original

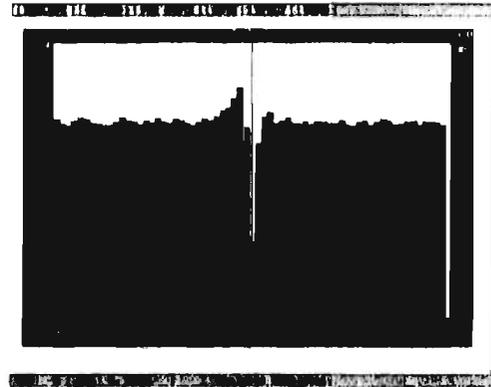


Figura 94 (5 Neuronas) Correlación cruzada 3 vs 5 correspondiente a la 1 Vs 3 original.

escala y con un ancho de columna también distinto, pero el comportamiento se mantiene igual en ambos.

Las correlaciones siguientes corresponden a los pares 1-4 y 1-5, pero debido a la correspondencia de los nombres nuevos, lo que se presenta en las gráficas de las correlaciones son los pares 4-1 y 5-1 (Figura 95 y Figura 96). Los resultados son similares a los obtenidos con XCOR después de GESSIN, es decir, hay ambigüedad en la inferencia de la conectividad.

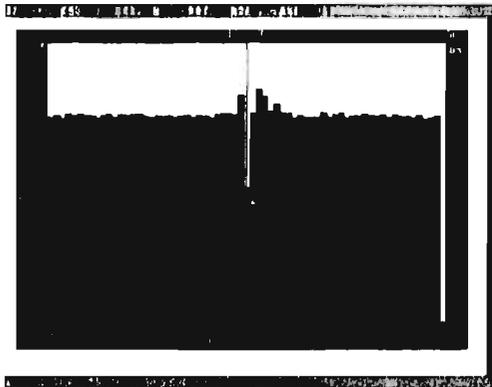


Figura 95 (5 Neuronas) Correlación cruzada 1 vs 3 correspondiente a la 4 Vs 1 original.

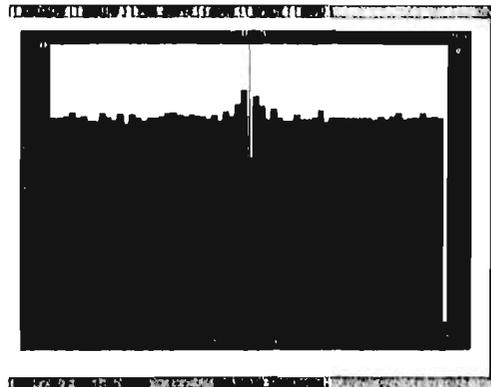


Figura 96 (5 Neuronas) Correlación cruzada 2 vs 3 correspondiente a la 5 Vs 1 original.

La correlación para el par 2-3 muestra el hueco a la izquierda del origen (Figura 97) como el original, hay que recordar que el diagrama de conexionamiento indica una inhibición entre estas dos neuronas. En la Figura 98 aparece la correlación del par 2-4 presentada como 4-2, si regresamos a observar la Figura 82 veremos un pico desplazado a la derecha, aquí

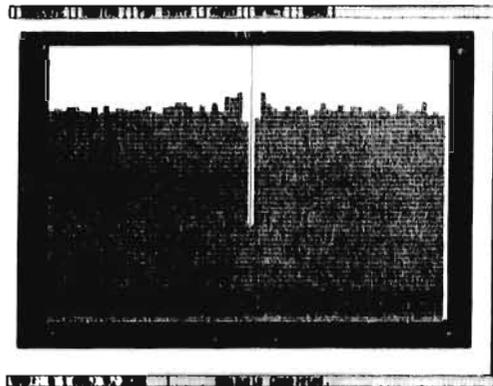


Figura 97 (5 Neuronas) Correlación cruzada 4 vs 5 correspondiente a la 2 Vs 3 original.

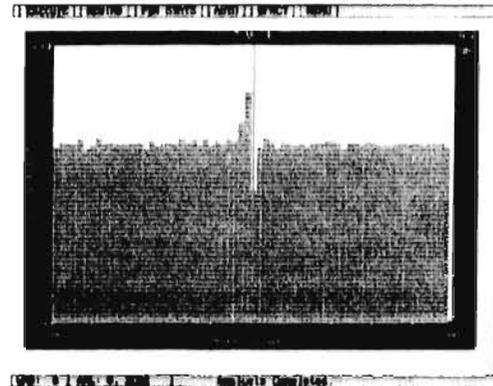


Figura 98 (5 Neuronas) Correlación cruzada 2 vs 4 correspondiente a la 5 Vs 2 original

aparece desplazado a la izquierda porque estamos analizando 4-2 y no 2-4 como se hizo originalmente.

La correlación 2-5 antes de GESSIN se presenta plana y después de GESSIN aparece un pico mismo que se reproduce en Discovery (Figura 98). No hay lugar a dudas que en este par de neuronas pierden su relación temporal al ser procesadas por GESSIN.

El par 3-4, que aquí es presentado como 4-3 muestra un histograma típico de una excitación. Tal como están ordenadas las relaciones, el histograma indicaría una excitación 3 a

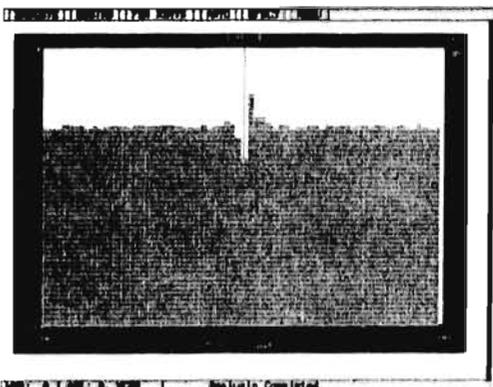


Figura 99 (5 Neuronas) Correlación cruzada 1 vs 4 correspondiente a la 4 Vs 2 original

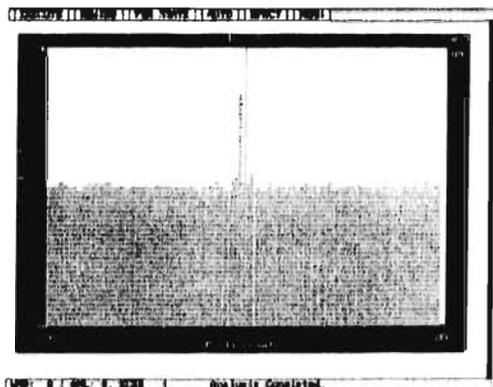


Figura 100 (5 Neuronas) Correlación cruzada 1 vs 5 correspondiente a la 4 Vs 3 original

4 que corresponde a los dos histogramas anteriores, antes y después de GESSIN, para ésta pareja, entonces podemos decir que aún con las pérdidas la dependencia en el tiempo entre estas neuronas se mantuvo (Figura 100).

El siguiente par es el 3-5, cuyo histograma se presenta en la Figura 101 y al igual que el último par 4-5 (Figura 102), presentan un hueco en el centro pero aparentemente superficiales de tal forma que podemos tomarlos como histogramas planos que denotan independencia. Por

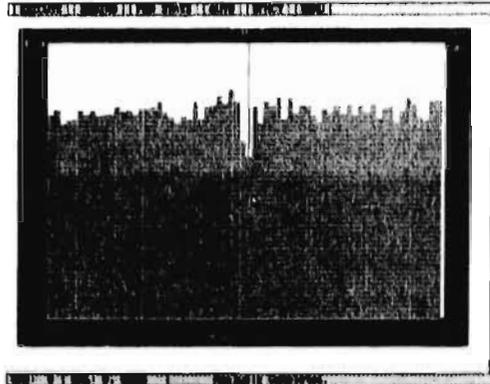


Figura 101 (5 Neuronas) Correlación cruzada 2 vs 5 correspondiente a la 3 Vs 5 original.

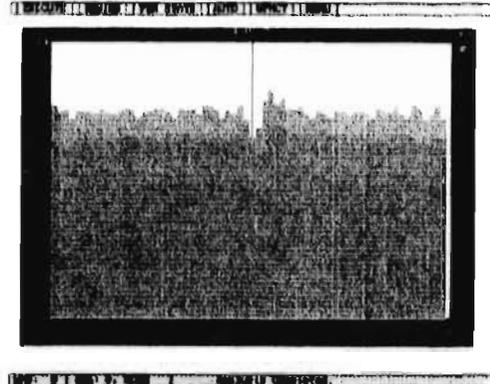


Figura 102 (5 Neuronas) Correlación cruzada 1 vs 2 correspondiente a la 4 Vs 5 original.

lo tanto podemos decir que, tanto las neuronas 3 y 4 como las 4 y 5 son independientes entre sí. ◆

Capítulo 4

Discusión de resultados y perspectivas.

En la presentación de los ejemplos anteriores es claro que uno de los puntos fundamentales a cuidar es el número de neuronas que se utilicen dado que mientras más grande sea éste, mas pérdida de información tendremos debido a los traslapes. Es lógico pensar que si en cada milisegundo de Neurored aparecen más de dos neuronas disparando al mismo tiempo, entonces de alguna manera se tiene que tomar la decisión de qué neurona es la que debe aparecer en la traza de la señal. Esto, como se explicó en el capítulo 2, se resolvió dando igual probabilidad de disparo a cada neurona, y como se vió en el ejemplo de 5 neuronas la pérdida de información se distribuye con igual peso para todas ellas. Sin embargo, dependiendo del tipo de conexión que presenten cada par, la pérdida de información al procesar la traza, puede llevar a la pérdida de la dependencia temporal, como sucedió en el caso de inhibición para el ejemplo de 5 neuronas. Tratándose de las simulaciones de sólo dos neuronas, la pérdida de información es mínima, como se pudo constatar en los ejemplos de este capítulo.

En los registros reales usando microelectrodos de metal, es decir, en registro extracelular, el mayor de los problemas es precisamente la superposición de las espigas que provienen de diferentes neuronas. Es de esperar, entonces que GESSIN reproduzca estos problemas conforme la simulación se acerca más a la configuración de un experimento real. En el ejemplo de 5 neuronas, presentado en el Capítulo 3, vemos que se presenta claramente este problema pero minimizado dado que, en primer lugar no se está agregando ruido, y en segundo lugar tenemos control sobre el conexionamiento de las neuronas. Si nos quedamos únicamente con los ejemplos de 2 neuronas, vemos una simulación casi perfecta, esta representaría más bien a un registro intracelular, en donde se usan micropipetas de vidrio. El interés dentro del Laboratorio de Cibernética es trabajar con el registro extracelular multiunitario. Aún con los problemas generados por los traslapes, el simulador GESSIN puede proporcionar información importante para trabajar la parte experimental.

Podemos afirmar que en todas las simulaciones presentadas aquí en las que se usaron dos neuronas, los resultados analizados antes del procesamiento con Discovery y después de él, concuerdan exactamente con los esperados, sin embargo en la simulación de 5 neuronas, no tenemos un grado de exactitud del 100%, conservadoramente, podríamos decir que tuvimos un 50% de confiabilidad en los resultados.

La conclusión más importante obtenida de estas simulaciones es la indicación que éstas hacen acerca de como plantear el montaje experimental de registro extracelular. Se puede ver que registrar más de tres neuronas con un sólo electrodo no es conveniente por la gran cantidad de información que se pierde al realizar la separación de las espigas que forman la señal compuesta. Tecnológicamente, según los resultados de Abeles [1], se pueden registrar de manera separable la actividad de hasta ocho neuronas con un microelectrodo de tungsteno, sin embargo, de acuerdo a los resultados de este trabajo, eso no es conveniente para estudios de conectividad funcional porque se pierde la detección de conectividad sobre todo en los casos donde existe inhibición. Los resultados obtenidos parecen sugerir que es mejor utilizar arreglos

grandes de microelectrodos de metal y con los cuales se registre cuando más hasta tres neuronas en cada microelectrodo del arreglo. Esto es factible porque se pueden fabricar microelectrodos de tungsteno con un afilado y barnizado tal que sólo registre dos o tres neuronas.

Por otra parte, los resultados indican que al hacer estudios de conectividad por medio de correlación cruzada, se debe tener mucha cautela con los datos que se manejan, ya provengan de simulaciones o de un experimento, pues es fácil hacer una interpretación errónea. Se deberían hacer registros experimentales en varios animales y en la misma zona cerebral, así como análisis a diferentes resoluciones. El simulador GESSIN es útil para preparar al analista a reconocer situaciones conflictivas, ya que se pueden generar todo tipo de señales extracelulares con variaciones difíciles de detectar. Mientras no se inventen nuevos métodos, el análisis por medio de correlación cruzada seguirá siendo la herramienta a utilizar en estudios de conectividad funcional.

El trabajar más ejemplos con circuitos de más de dos neuronas, podría traer dar una pauta de las posibles modificaciones para aumentar el grado de confiabilidad en los resultados de esas simulaciones, quizá el camino para lograrlo sea utilizar una distribución de probabilidad para decidir que hacer en los traslapes, o quizás utilizar una red neuronal artificial que, entrenada con algunas simulaciones, decida a que neuronas se les da prioridad para que en los traslapes, al ser procesadas por GESSIN, no pierdan su dependencia temporal.

La separación y clasificación de espigas para el estudio de conectividad entre neuronas es un problema que involucra muchos problemas aún no resueltos completamente, tanto desde el punto de vista instrumental, de hardware como de software. El simulador aquí presentado es una herramienta en software desarrollada con el objetivo de proporcionar al investigador de estas áreas, un apoyo para el entrenamiento en el análisis de señales neurofisiológicas, que puede servir también para tener una mejor idea de qué puede suceder durante la separación de espigas en un experimento real. El tiempo necesario para llevar a efecto la preparación en un animal vivo, para poder efectuar un registro, puede ser de varios días, y aún se tiene la dependencia de las características de resistencia del animal para llevar a buen termino la grabación del registro. Durante él se tienen que tomar decisiones rápidas acerca de la manipulación de los electrodos para llegar a zonas de interés y saber cuándo se tienen señales que pueden dar información acerca de como están comunicándose las neuronas entre sí. Todo esto registrando únicamente bajo condiciones de respuesta espontánea, el problema se complica aún más si ahora se considera un estímulo para generar potenciales evocados.

Uno de los problemas más comunes que se presentan durante el registro es la aparición de ruido de distintas fuentes que van desde el ruido de la línea de AC, relativamente fácil de quitar, hasta ruido de radiofrecuencia y autoinducido. Jaulas de Faraday y elementos blindados han ido resolviendo estos problemas. Pero este es un problema que todavía no se agregó a GESSIN, como una ampliación a este trabajo de tesis, agregaremos un módulo para añadir ruido a la señal, que va desde el ruido gaussiano y blanco hasta ruido coloreado. Cabe hacer notar que este no es problema trivial, dado que en señales reales pueden aparecer varios tipos de ruido mezclados.

Aunque el uso de GESSIN¹, requiere de una tarjeta de conversión D/A, hasta antes del módulo DAC-ESP puede correr en cualquier computadora PC-compatible con las

¹ Para quienes pudieran estar interesados en usar GESSIN, ponerse en contacto con el Laboratorio de Cibernética del Departamento de Física de la Facultad de Ciencias de la UNAM con Fis. Roberto Serna Herrera E-Mail: roberto@ciber2.fciencias.unam.mx / serna@servidor.unam.mx o con Dr. José Ismael Espinosa Espinosa E-Mail: ismael@ciber2.fciencias.unam.mx / espin@servidor.unam.mx

características descritas en el capítulo 2 (página 23). El manejo del módulo XCOR, puede ser de mucha utilidad no sólo para archivos generados por el mismo GESSIN, sino desde cualquier otro programa siempre y cuando se cumpla con el formato del archivo de entrada descrito también en el mismo capítulo (página 29).

Se espera poder continuar desarrollando este simulador, con las necesidades que vayan surgiendo durante su uso en paralelo con los experimentos que el Laboratorio de Cibernética viene realizando desde hace ya algunos años, a fin de conseguir resultados tanto para beneficio de las neurociencias como para la tecnología contemporánea. ◆

Apéndice A

Listado del archivo de constantes comunes.

```

Const
  NumEspigDatos=71;           {Numero de espigas patrón (n-1) }
  DatEspig=129;              {Numero de puntos datos para c/espiga=128
                             +1 como etiqueta de la espiga}

  DatAjuste=16;
  NumTiposDifereEspig=11;
  NumMaxEspig=60000;
  LimMaxEspig=967;
  LimMinEspig=53;
  Nivel_0=470;
  Voltaje_0=0;
  VoltMax=1;
  VoltMin=-1;
  PathDatosConst='\genera\datos\';
  EspigasPatron='\genera\invertid.dat';
  ArchivoSalida='salida.dat';
  bgi='';

Type
  Tvec                =Array[1..DatEspig] of Real;
  Tregistro           =Record
                        valor:Byte;
                        disparo:boolean;
  End;
  TNumTiposEspig     =Array[1..NumTiposDifereEspig]of Byte;
  TvecFile           =File of Tvec;
  TvecSenal          =Array[1..NumMaxEspig]of Byte;
  TvecSenalFile      =File of TvecSenal;

```


Listado del programa UNE-VEC

```

Program UneEspigas;
( MODIFICACION: 180497
Este programa crea , a partir del archivo base ESPIGAS.DAT, el archivo
SEÑAL.DAT (Binario) en donde se guarda la señal compuesta en vectores
de 129 elementos de reales, el primer elemento es la etiqueta cada espiga )

Uses
  crt, graph;
($i const)
Var
  PathDatos                :string(30);
  TiposEspig, PosNeuron   :TNumTiposEspig;
  VecSenal                 :TvecSenal;
  ColumNeurona            :Array[1..NumTiposDifereEspig] of Tregistro;
  dat, sal                 :TvecFile;
  VecSenalSal              :TvecSenalFile;
  tiempos, nRed            :Text;
  j, k, dato, Gd, Gm, Tx, Ty, h, OfsX1, OfsY1,
  OfsX2, OfsY2, DespX, DespY, NumNeuronas,
  NumEspig, NumColum       :Integer;
  i, CuentaEspig, EspigNoCeroEnSenal,
  CeroYNoCeroSenal        :Word;
  FacY, hreal              :Real;
  ViewPort                 :ViewPortType;
  vec, vecCero             :Tvec;
  nom, NomEntrad           :string;
  ch                        :char;

Function ExisteSalida(var arch:TvecSenalFile; nom:string):Boolean;
Begin
  assign(arch, nom);
  {$i-} Reset(arch); {$i+}
  ExisteSalida:=ioresult=0;
End;

Function Existe(var arch:TvecFile; nom:string):Boolean;
Begin
  assign(arch, nom);
  {$i-} Reset(arch); {$i+}
  Existe:=ioresult=0;
End;

Function ExisteTiempos(nom:string):Boolean;
Begin
  assign(tiempos, nom);
  {$i-} Reset(tiempos); {$i+}
  ExisteTiempos:=ioresult=0;
End;

Procedure AbreArchivoEntrada;
var
  x, y:Integer;
Begin
  repeat
    ClrScr;
    Write('Nombre del archivo de las espigas patron : ');
    x:=WhereX; y:=WhereY;
    Readln(Nom);
    NomEntrad:=Nom;
    if Nom='' then
      begin
        Nom:='espigas.dat';
        NomEntrad:='espigas.dat';
        TextColor(Red);
        GotoXY(x, y);
        writeln(Nom);
        TextColor(LightGray);
      end
    end
  until Nom<>'';
  Nom:=PathDatos+nom;
  if not existe(dat, Nom) then

```

```

        Begin
            writeln('El archivo ',NomEntrad,' no existe. ');
            writeln('FIN DEL PROGRAMA');
            repeat until keypressed;
            HALT;
        End;
    until
        existe(dat,Nom);
    Reset(dat);
    Read(dat,vec);
    NumEspig:=round(vec[1]);
End;

Procedure AbreArchivoTiempos;
var
    j,k,x,y:integer;
begin
    repeat
        Write('Nombre del archivo de tiempos de disparo : ');
        x:=WhereX; y:=WhereY;
        Readln(nom);
        if nom='' then
            Begin
                nom:='muestral.sal';
                TextColor(Red);
                GotoXY(x,y);
                writeln(Nom);
                TextColor(LightGray);
            End
        else
            nom:=PathDatos+nom;
            if not existeTiempos(nom) then
                Begin
                    writeln('El archivo ',nom,' no existe. ');
                    writeln('FIN DEL PROGRAMA');
                    repeat until keypressed;
                    HALT;
                End;
            until
                existeTiempos(nom);
            Reset(tiempos);
            read(tiempos,j);
            NumColumn:=0;
            while not eoln(tiempos) do
                begin
                    read(tiempos,j);
                    NumColumn:=NumColumn+1;
                end;

            write('El archivo ');
            textcolor(lightred+blink);
            write(nom);
            textcolor(lightgray);
            writeln(' tiene ',NumColumn,' columnas de tiempos. ');
            write('Cuantas neuronas se usaran ? : ');
            readln(NumNeuronas);
            for i:=1 to NumNeuronas do
                Begin
                    write('Posicion de las neuronas en el archivo ? : ');
                    readln(PosNeuron[i]);
                end;
            for j:=1 to NumColumn do
                With ColumnNeurona[j] do
                    Begin
                        valor:=0;
                        disparo:=False;
                    End;
            for j:=1 to NumNeuronas do
                for i:=1 to NumColumn do
                    if PosNeuron[j]=i then
                        ColumnNeurona[i].valor:=j;
                end;
            End;

Procedure AbreArchivoSalida;
Begin
    repeat
        ClrScr;
        Write('Nombre del archivo de salida para la senal compuesta : ');

```

```

Readln(nom);
if nom='' then
  nom:=PathDatos+'senal.dat'
else
  nom:=PathDatos+nom;
if ExisteSalida(VecSenalSal,nom) then
  Begin
    write('El archivo ');
    textcolor(lightred+blink);
    write(nom);
    textcolor(lightgray);
    writeln;
    writeln('ya existe quieres sobrescribirlo? [s/n] ');
    repeat
      ch:=readkey;
      if not (ch in ['s','S','n','N']) then
        write('^g');
    until
      ch in ['s','S','n','N']
  End;
until
  not ExisteSalida(VecSenalSal,nom) or (ch in ['s','S']);
Rewrite(VecSenalSal);
Write(VecSenalSal,VecSenal); { Se abre el espacio para el arreglo cuyo
                             primer elemento es el no. de ms }

End;

Function Xp(x:Integer):Integer;
Begin
  Xp:=x-Tx;
End;

Function Yp(y:Integer):Integer;
Begin
  Yp:=Ty-y;
End;

Procedure IniciaGraficos;
Begin
  Gd:=Detect;
  InitGraph(Gd,Gm,bgi);
End;

Procedure LeeVec;
Begin
  if VecSenal[i]=0 then
    Vec:=vecCero
  else
    begin
      reset(dat);
      seek(dat,VecSenal[i]);
      Read(dat,vec);
    End;
End;

Procedure LeeVecPatron;
Begin
  Read(dat,vec);
End;

Procedure Ventana(v,DespX,DespY:integer);
Begin
  SetColor(Yellow);
  Case v of
    0: Begin
      SetViewPort(0,0,GetmaxX,GetmaxY,ClipOn);
      ClearViewPort;
      GetViewSettings(ViewPort);
      with ViewPort do
        Rectangle(0, 0, x2-x1, y2-y1);
      End;
    1: Begin
      SetViewPort(0,0,GetmaxX,GetmaxY,ClipOn);
      SetViewPort(OfsX1+DespX-1,OfsY1+DespY-1,
        OfsX2+DespX+1,ofsY2+DespY+1,ClipOn);
      GetViewSettings(ViewPort);
      with ViewPort do
        Rectangle(0, 0, x2-x1, y2-y1);
    End;
  End;
End;

```

```

SetViewPort(ofsX1+DespX,ofsY1+DespY,
             ofsX2+DespX,ofsY2+DespY,ClipOn);
GetViewSettings(ViewPort);
with ViewPort do
  Begin
    h:=Round((x2-x1)/128);
    if h<=0 then
      h:=1;
    FacY:=(y2-y1)/(VoltMax-VoltMin);
    Ty:=(y2-y1) div 2;
    Tx:=0;
  End;
End;
2: Begin
SetViewPort( OfsX1-1, OfsY1+DespY-1,
             GetmaxX+1,OfsY2+DespY+1,ClipOn);
GetViewSettings(ViewPort);
SetViewPort( OfsX1, OfsY1+DespY,
             GetmaxX, OfsY2+DespY,ClipOn);
GetViewSettings(ViewPort);

with ViewPort do
  Begin
    hreal:=((x2-x1)/(128*30));
    { 30 es el numero de espigas a ver por linea}
    FacY:=(y2-y1)/(VoltMax-VoltMin);
    Ty:=(y2-y1) div 2;
    Tx:=0;
  End;
End;
3: Begin
SetViewPort(0,0,GetmaxX,GetmaxY,ClipOn);
SetViewPort(50,20,450,700,ClipOn);
GetViewSettings(ViewPort);
with ViewPort do
  Rectangle(0, 0, x2-x1, y2-y1);
SetViewPort(51,21,449,699,ClipOn);
GetViewSettings(ViewPort);
with ViewPort do
  Begin
    h:=Round((x2-x1)/128);
    if h<=0 then
      h:=1;
    FacY:=(y2-y1)/(VoltMax-VoltMin);
    Ty:=(y2-y1) div 2;
    Tx:=0;
  End;
End;
End;
SetColor(white);
End;

Procedure MuestraEspigas;
Var
  cad:string;
  k,w:Integer;
Begin
  IniciaGraficos;
  OfsX1:=1; OfsY1:=1;
  OfsX2:=GetmaxX Div 5;
  OfsY2:=GetmaxY Div 6;
  DespX:=0; DespY:=0;
  i:=1;
  k:=0;
  SetTextStyle(SmallFont,Horizdir,4);
  Repeat
    LeeVecPatron;
    w:=((i-1) Mod 25) mod 5;
    DespX:=w*OfsX2;
    DespY:=k*OfsY2;
    ventana(1,DespX,DespY);
    ClearViewPort;
    str(vec[1]:2:0,cad);
    OutTextXY(4,4,cad);
    if (round(Vec[1]) mod 25=1) or (vec[1]=1) or (i=1) then
      OutTextxy(4,65,NomEntrad);
    MoveTo(Xp(0),Yp(Round(vec[2]*Facy)));
  End;
End;

```

```

        for j:=3 to DatEspig do
            PutPixel(Xp((j-1)*h), Yp(Round(vec[j]*Facy)), LightGreen);
        Inc(i);
        if (i-1) Mod 25=0 then
            Begin
                ch:=Readkey;
                Ventana(0,0,0);
                ClearViewPort;
            End;
            k:=((i-1) Mod 25) div 5;
        Until
            (i=NumEspig+1) Or Keypressed;
        ch:=Readkey;
        RestoreCRTmode;
    End;

Procedure SuperponEspigas;
Var
    cad:string;
    k,w:Integer;
Begin
    IniciaGraficos;
    OfsX1:=1; OfsY1:=1;
    OfsX2:=GetmaxX Div 5;
    OfsY2:=GetmaxY Div 6;
    DespX:=0; DespY:=0;
    i:=1;
    k:=0;
    ch:='N';
    Repeat
        Reset(dat);
        SetTextStyle(SmallFont, Horizdir, 8);
        if Ucase(ch)='B' then
            SetBkColor(White)
        else
            SetBkColor(Black);
        For i:=1 to NumNeuronas do
            Begin
                Seek(dat, TiposEspig[i]);
                Read(dat, vec);
                ventana(3,0,0);
                MoveTo(Xp(0), Yp(Round(vec[2]*Facy)));
                for j:=3 to DatEspig do
                    Begin
                        Setcolor(i+8);
                        LineTo(Xp((j-1)*h), Yp(Round(vec[j]*Facy)));
                    End;
                ventana(3,0,0);
                str(TiposEspig[i], nom);
                SetColor(i+8);
                SetTextStyle(DefaultFont, Horizdir, 2);
                OutTextXY(20, i*16, nom);
            End;
        OutTextXY(2, GetMaxY-30, 'S salir, Fondo: B Blanco, N negro');
        ch:=ReadKey;
    until
        upcase(ch)='S';
    RestoreCRTmode;
End;

Procedure MuestraSenal;
var
    EspigasPorLinea,
    NumLineasEspig, x, y: Integer;
Begin
    IniciaGraficos;
    CuentaEspig:=0;
    SetGraphMode(gm);
    ClearDevice;
    Reset(VecSenalSal);
    Read(VecSenalSal, VecSenal);
    Read(VecSenalSal, VecSenal);
    Read(VecSenalSal, VecSenal);
    OfsX1:=1; OfsY1:=1;
    OfsX2:=0;
    OfsY2:=GetmaxY Div 8;
    DespX:=0; DespY:=0;
    NumLineasEspig:=0;

```

```

ch:=' ';
i:=1;
While (i<=CeroYNoCeroSenal) and (Ucase(ch)<>'S') do
  BBegin
    DespY:=NumLineasEspig*OfsY2;
    ventana(2,0,DespY);
    EspigasPorLinea:=1;
    While (i<=CeroYNoCeroSenal) and (EspigasPorLinea<=30) do
      Begin
        LeeVec;
        Inc(CuentaEspig);
        for j:=3 to DatEspig do
          Begin
            PutPixel(Xp(Round((j-1)*hreal+(EspigasPorLinea-1)*128*hreal)),
              Yp(Round(vec[j]*FacY)),LightGreen);
            x:=Xp(Round((j-1)*hreal+EspigasPorLinea*128*hreal));
          end;
          if (CuentaEspig+1) Mod 10=0 then
            Begin
              str(CuentaEspig+1,nom);
              SetColor(Cyan);
              SetTextStyle(SmallFont, HorizDir,
                SmallFont);
              if (CuentaEspig<=999) And ((CuentaEspig+1) Mod 30<>0) then
                OutTextXY(x-20,(GetMaxY Div 8)-8,nom)
              else
                OutTextXY(x-34,(GetMaxY Div 8)-8,nom);
              SetColor(Green);
            end;
            inc(EspigasPorLinea);
            inc(i);
          end;
          inc(NumLineasEspig);
          if NumLineasEspig Mod 8 = 0 then
            Begin
              NumLineasEspig:=0;
              ch:=ReadKey;
              ClearDevice;
            End;
          End;
          Str(CuentaEspig,nom);
          SetViewPort(0,0,GetmaxX,GetmaxY,ClipOn);
          SetColor(cyan);
          OutTextXY(GetMaxX-150,GetmaxY-30,'Espigas= '+nom);
          SetColor(green);
          ch:=ReadKey;
          RestoreCRTmode;
        End;
      End;
    End;
  End;
  Procedure Raster;
  var
    x,y:integer;
  Begin
    IniciaGraficos;
    SetGraphMode(gm);
    i:=1;
    y:=20;
    x:=0;
    Repeat
      if VecSenal[i]<>0 then
        PutPixel(x,y,lightgreen)
      else
        PutPixel(x,y,Black);
      inc(i);
      inc(x);
      for k:=1 to 3 do
        Begin
          PutPixel(x,y,Black);
          inc(x);
        End;
      if x mod (GetMaxX-1)=0 then
        Begin
          x:=0;
          y:=y+20;
        End;
      Until
        i>=2*CuentaEspig;
    ch:=readKey;
  End;

```

```

RestoreCRTmode;
End;

Procedure LeeNumEspig;
Var i,x,y:integer;
Begin
Write(' Qu. formas de onda quieres usar ? ');
x:=WhereX; y:=WhereY;
VecSenal[1]:=NumNeuronas;
For i:=1 to NumNeuronas do
Begin
Read(TiposEspig[i]);
VecSenal[i+1]:=TiposEspig[i];
x:=x+3;
GotoXY(x,y);
End;
Write(VecSenalSal,VecSenal); ( Se escribe, en el segundo arreglo del
archivo de salida el numero de neuronas
usadas y los tipos de las formas de onda
en los elementos siguientes )
End;

Procedure GeneraSenal;
Type
TLineaTiempos =Array[1..DatEspig]of byte;
var
j,k,m,DetectaUnos :Integer;
i,indice :word;
LineaTiempos :TLineaTiempos;
YaDisparo :Boolean;
Begin
TextColor(red);
gotoxy(15,10);
write('Paciencia... estoy generando la senal.');
```

TextColor(LightGray);

For i:=2 to DatEspig do

VecCero[i]:=0;

i:=1;

reset(tiempos);

EspigNoCeroEnSenal:=0;

CeroYNoceroSenal:=0;

While not eof(tiempos) do

Begin

read(tiempos,indice);

DetectaUnos:=0;

for j:=1 to NumColum do

read(tiempos,LineaTiempos[j]);

readln(tiempos);

For j:=1 to NumColum do

if ColumNeurona[j].valor<>0 then

DetectaUnos:=DetectaUnos+LineaTiempos[j];

Case DetectaUnos of

0:VecSenal[i]:=0;

1:Begin

j:=1;

repeat

If (LineaTiempos[j]=1) and

(ColumNeurona[j].valor<>0) then

Begin

VecSenal[i]:=TiposEspig[ColumNeurona[j].valor];

j:=NumColum;

End;

inc(j);

until

j>=NumColum

End

Else

Begin

j:=1; YaDisparo:=False;

repeat

repeat

If (LineaTiempos[j]=1) and

(ColumNeurona[j].valor<>0) and

(Not ColumNeurona[j].disparo) then

Begin

VecSenal[i]:=TiposEspig[ColumNeurona[j].valor];

ColumNeurona[j].disparo:=True;

End

until

until

End

End

End

```

YaDisparo:=True;
m:=1;
repeat
  if Not ColumNeurona[m].disparo then
    m:=NumNeuronas+2
  else
    inc(m);
    if m=NumNeuronas+1 then
      For k:=1 to NumNeuronas do
        ColumNeurona[k].disparo:=False;
      until
        m>=NumNeuronas+1;
      j:=NumColum;
    End;
    inc(j);
  until
    j>=NumColum;
  if Not YaDisparo then
    Begin
      For k:=1 to NumNeuronas do
        ColumNeurona[k].disparo:=False;
      j:=1
    End;
  until
    Yadisparo;
end;
End; (END CASE)
if VecSenal[i]=0 then
  Begin
    (write(sal,vecCero);}
    Inc(CeroYNoCeroSenal);
  end
else
  begin
    Inc(EspigNoCeroEnSenal);
    Inc(CeroYNoCeroSenal);
  End;
  inc(i);
end; (END WHILE)
Write(VecSenalSal,VecSenal); { Como tercer arreglo se guarda la senal }
i:=VecSenal[1];
Reset(VecSenalSal);
VecSenal[1]:=CeroYNoCeroSenal DIV 1000;
Write(VecSenalSal,VecSenal); { como primer arreglo el no. de ms }
VecSenal[1]:=i;
gotoxy(15,11);
write('No. Total de ms usados en la traza= ',CeroYNoCeroSenal);
gotoxy(15,12);
write('No. de espigas = ',EspigNoCeroEnSenal);
gotoxy(15,13);
TextColor(red+blink);
write('Listo!, presiona una tecla para continuar.');
```

```

TextColor(lightGray);
ch:=readkey;
ClrScr;
End;

Procedure GenSalNeurored;
Begin
  clrscr;
  write('Cual es el nombre del archivo para la salida de Neurored ? ');
  readln(nom);
  nom:=PathDatos+nom;
  assign(nRed,nom);
  Rewrite(nRed);
  For i:=1 to CeroYNoCeroSenal do
    Begin
      Write(nRed,i:5);
      for j:=1 to NumNeuronas do
        Begin
          if VecSenal[i]=0 then
            write(nRed,0:5)
          else
            if VecSenal[i]=TiposEspig[j] then
              write(nRed,1:5)
            else
              write(nRed,0:5);
        End;
      End;
    End;
  End;
End;

```


Listado del programa DAC-ESP

```

Program DAC;
{Ultima modificacion 200497
Programa para usar el DAC del la tarjeta DT821.
Se saca por DAC la senal generada a traves del programa UNE-vec.PAS
Roberto Serna
Uses
  Dos,crt;
{$i const.pas}

TYPE
  Str4 = string[4];
  TListaEspigas = ^Campos;
  Campos = Record
    Vec : Tvec;
    Siguiente: TListaEspigas;
  End;

VAR
  dat : TvecFile;
  VecSenalFile : TvecSenalFile;
  VecSenal : TvecSenal;
  VecEspigaPatron,
  VecEspigaPatronCero : Tvec;
  NumNeuron,i,j,base : integer;
  k,h1,h2,m1,m2,s1,s2,cs1,cs2,
  miliseq : Word;
  h,facy : real;
  ch : char;
  Nom,PathDatos : string;
  ListaEspigas,
  Espiga : TListaEspigas;
  TiposEspiga : TNumTiposEspig;

CONST
  Bit0 = $1 ;
  Bit1 = $2 ;
  Bit2 = $4 ;
  Bit3 = $8 ;
  Bit4 = $10 ;
  Bit5 = $20 ;
  Bit6 = $40 ;
  Bit7 = $80 ;
  Bit8 = $100 ;
  Bit9 = $200 ;
  Bit10 = $400 ;
  Bit11 = $800 ;
  Bit12 = $1000 ;
  Bit13 = $2000 ;
  Bit14 = $4000 ;
  Bit15 = $8000 ;

  DA_DIOCSR : integer = $246; { Registro de control R/W (BASE + 6) del D/A y DIO}
  DADAT : integer = $248; { Registro de datos del D/A WO (BASE + 8) }
  TMRCTR : integer = $24E; { Regsitro del contador/timer R/W (BASE + E) }
  SUPCSR : integer = $24C; { Registro de control del DMA R/W (BASE + C) }

Function PonCero(x:word):String;
Begin
  str(x,nom);
  if length(nom)=1 then
    Poncero:='0'+nom
  else
    Poncero:=nom;
End;

Function Existe(var arch:TvecSenalFile; nom:string):Boolean;
Begin
  assign(arch,nom);
  {$i-} Reset(arch); {$i+}
  Existe:=ioresult=0;
End;

Procedure AbreArchivoEntrada;
var
  x,y:Integer;

```

```

nomentrad:String;
Begin
  repeat
    ClrScr;
    Write('Nombre del archivo de datos a sacar por el DCA : ');
    x:=WhereX; y:=WhereY;
    Readln(Nom);
    NomEntrad:=Nom;
    if Nom='' then
      begin
        Nom:='senal.sen';
        NomEntrad:=nom;
        TextColor(Red);
        GotoXY(x,y);
        writeln(Nom);
        TextColor(LightGray);
      end
    else
      Nom:=PathDatosConst+nom;
      if not existe(VecSenalFile,Nom) then
        Begin
          writeln('El archivo ',NomEntrad,' no existe. ');
          writeln('FIN DEL PROGRAMA');
          ch:=ReadKey;
          HALT;
        End;
      until
        existe(VecSenalFile,Nom);
      Reset(VecSenalFile);
    End;

procedure DAC_Output( BothDacs : Str4;
                      DataValue : Integer );
VAR
  Timer:Integer;
  DAready:Boolean;

BEGIN
  DAready := False;      (* initialize flag *)
  Timer := 10000;        (* initialize Timer *)
  repeat
    DAready := (Portw[DA_DIOCSR] AND Bit7) = Bit7;
    Timer := Timer - 1;
    if (Timer = 0) then
      begin
        writeln;
        writeln('*** El bit de listo en DA/DIOCSR no esta prendido');
        Timer := 10000;
      end;
    until
      DAready;
    Portw[DADAT] := DataValue;
    if BothDacs = 'Both' then
      Portw[DADAT] := DataValue;
    Portw[SUPCSR] := Bit7;
    if (Portw[DA_DIOCSR] AND Bit15) <> 0 then
      begin
        ClrScr;
        writeln(' *** El bit de error D/A se prendio durante la conversion');
        Readln;
      end;
    END;

Procedure InsertaNodo(Var ListaEspigas:TListaEspigas; Elemento:TVec);
Var
  Nuevo:TListaEspigas;
Begin
  New(Nuevo);
  With Nuevo^ do
    Begin
      Vec:=Elemento;
      Siguiente:=Nil;
    End;
  ListaEspigas^.siguiente:=Nuevo;
  ListaEspigas:=Nuevo;
End;

```

```

Procedure CreaListaEspigas;
Begin
  New(ListaEspigas);
  ListaEspigas^.siguiente:=nil;
  Assign(dat, 'espigas.dat');
  For j:=1 to 129 do
    VecEspigaPatronCero[j]:=0;
  ListaEspigas^.vec:=VecEspigaPatronCero;
  Espiga:=ListaEspigas;
  For i:=1 to NumNeuron do
    Begin
      reset(dat);
      seek(dat,TiposEspiga[i]);
      Read(dat, VecEspigaPatron);
      InsertaNodo(Espiga, VecEspigaPatron);
    End;
End;

Procedure LeeEspigaPatron(Numero: Integer; Espiga: TListaEspigas; Var vec: Tvec);
Begin
  While (Espiga<>Nil) and (Espiga^.vec[1]<>Numero) do
    Espiga:=Espiga^.siguiente;
  vec:=Espiga^.vec;
End;

Begin
  ClrScr;
  GotoXy(10,24);
  PathDatos:=PathDatosConst;
  Write('La ruta por defecto para datos es: ');
  textcolor(lightred);
  Write(PathDatos);
  textcolor(lightgray);
  GotoXy(45,24);
  Readln(nom);
  if nom<>' ' then
    PathDatos:=nom;
  AbreArchivoEntrada;
  Read(VecSenalFile, VecSenal); (1er. Arreglo No. de ms )
  MiliSeg:=VecSenal[1]*1000;
  Read(VecSenalFile, VecSenal); (2o. Arreglo No. de Neuronas y tipos)
  NumNeuron:=VecSenal[1];
  For i:=1 to NumNeuron do
    TiposEspiga[i]:=VecSenal[i+1];
  Read(VecSenalFile, VecSenal); (3er. Arreglo Traza de la senal )
  CreaListaEspigas;
  write('Listo para empezar, presiona ENTER ');
  readln;
  reset(dat);
  facy:=20/4096;
  gotoxy(30,10);
  writeln('Sacando datos');
  GetTime(h1,m1,s1,cs1);
  For k:=1 to MiliSeg do
    Begin
      If VecSenal[k]=0 then
        LeeEspigaPatron(0, ListaEspigas, VecEspigaPatron)
      else
        LeeEspigaPatron(VecSenal[k], ListaEspigas, VecEspigaPatron);
      for i:=2 to 129 do
        dac_output('Both', round((VecEspigaPatron[i]+10)/facy));
    End;
  GetTime(h2,m2,s2,cs2);
  if cs1>cs2 then
    Begin
      s2:=s2-1;
      cs2:=cs2+100;
    End;
  if s1>s2 then
    BEgin
      m2:=m2-1;
      s2:=s2+60;
    End;
  if m1>m2 then
    BEgin
      h2:=h2-1;
      m2:=m2+60;
    End;
End;

```

```
Write('El tiempo total de la señal es: (hh:mm:ss.cc) ');
Write(PonCero(h2-h1)+' ');
Write(PonCero(m2-m1)+' ');
Write(PonCero(s2-s1)+' ');
Write(PonCero(cs2-cs1));
Writeln;
Writeln(' Todo Hecho...');
Readln;
End.
```

Listado del programa XCOR

```

Program Xcor;
{Ultima Modificacion :      310397
Este programa calcula la correlacion entre dos columnas de un archivo
de texto en el formato de salida de neurored}

Uses
  crt, graph;
{$i const}

const
  tope                =30000;
  TopeAnchoVentana    =300;

Type
  Tcolum              =Array[1..tope]of byte;

Var
  dat, sal             :text;
  LimInf, LimSup       :word;
  serie1, serie2       :Tcolum;
  suma                :Array[-TopeAnchoVentana..
                          TopeAnchoVentana] of longint;

  i, j, k, NumColum, Neu1, Neu2, dato, lineas,
  gd, gm, Tx, Ty, AnchoVentana, Maximo, Total: Integer;
  nomentrad, nom, PathDatos : String;
  ch                   : Char;

Function Existe(nom:string): Boolean;
Begin
  assign(dat, nom);
  {$i-} Reset(dat); {$i+}
  Existe:=ioresult=0;
End;

Procedure AbreArchivoEntrada;
const
  pathDatos='\genera\';
var
  x, y: Integer;
Begin
  repeat
    clrscr;
    Write('Nombre del archivo de disparos : ');
    x:=WhereX; y:=WhereY;
    Readln(Nom);
    NomEntrad:=Nom;
    if Nom='' then
      begin
        Nom:='muestral.sal';
        NomEntrad:=Nom;
        TextColor(Yellow);
        GotoXY(x, y);
        writeln(Nom);
        TextColor(LightGray);
      end
    else
      Nom:=PathDatosConst+nom;
      if not existe(Nom) then
        Begin
          writeln('El archivo ', NomEntrad, ' no existe. ');
          writeln('FIN DEL PROGRAMA');
          repeat until keypressed;
          HALT;
        End;
      until
        existe(Nom);
  End;

End;

Procedure LeeDatos;
Var
  x, y: Integer;
Begin

```

ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA

```

Reset(dat);
read(dat,j);
NumColum:=0;
while not eoln(dat) do
begin
    read(dat,j);
    NumColum:=NumColum+1;
end;
write('En el archivo ');
textcolor(Yellow);
write(nomEntrad);
textcolor(lightgray);
writeln(' se tienen ',NumColum,' columnas de datos. ');
write('Neuronas a correlacionar (NEU1,NEU2): ');
readln(Neu1,Neu2);
TextColor(Yellow+blink);
x:=WhereX; y:=WhereY;
Write('Espera');
textcolor(lightgray);
Writeln(' estoy leyendo los datos... ');
Reset(dat);
i:=1;
(Se lee la primera serie)
While not eof(dat) do
Begin
    read(dat,dato);
    for j:=1 to Neu1-1 do
        read(dat,dato);
        read(dat,serie1[i]);
        readln(dat);
        inc(i);
    End;
Reset(dat);
i:=1;

(Se lee la segunda serie)
While not eof(dat) do
Begin
    read(dat,dato);
    for j:=1 to Neu2-1 do
        read(dat,dato);
        read(dat,serie2[i]);
        readln(dat);
        inc(i);
    End;
Lineas:=i-1;
GotoXY(X,Y);
Write('Datos Leidos. ');
Clreol;
Writeln;
Write('Este archivo tiene ');
TextColor(Yellow);
Write(lineas);
TextColor(lightgray);
writeln(' milisegundos. ');
Write('Intervalo en milisegundos a usar (T1,T2): ');
Readln(LimInf, LimSup);
Write('Ancho de la ventana de despliegue (A): ');
readln(AnchoVentana);
TextColor(Yellow+blink);
Write('Espera');
textcolor(lightgray);
Writeln(' estoy realizando unos calculos... ');
End;

Procedure CuentaOurrencias;
Begin
    for k:=-AnchoVentana to AnchoVentana do
        suma[k]:=0;

    (Se cuentan las ocurrencias del lado POSITIVO)
    if LimInf=0 then
        LimInf:=1;
    for i:=LimInf to LimSup do
        if i<=LimSup-Anchoventana then
            Begin
                if serie1[i]=1 then
                    for j:=i to i+AnchoVentana do

```

```

        if serie2[j]=1 then
            inc(suma[j-i]);
        End
    else
        if Serie1[i]=1 then
            for j:=i to LimSup do
                if serie2[j]=1 then
                    inc(suma[j-i]);
            End
        End
    End
    (Se cuentan las ocurrencias del lado NEGATIVO)
    for i:=LimSup downto LimInf do
        if i>Liminf*Anchoventana then
            Begin
                if serie1[i]=1 then
                    for j:=i-1 downto i-AnchoVentana do
                        if serie2[j]=1 then
                            inc(suma[j-i]);
                    End
                End
            else
                if Serie1[i]=1 then
                    for j:=i-1 downto LimInf do
                        if serie2[j]=1 then
                            inc(suma[j-i]);
                    End
                End
            End
        End
        Maximo:=0;
        For i:=-AnchoVentana to Anchoventana do
            if suma[i]>Maximo then
                Maximo:=Suma[i];
        Total:=Maximo;
        if (Maximo>=350) And (Maximo<3750) then
            For i:=-AnchoVentana to Anchoventana do
                suma[i]:=Round(0.1*suma[i]);
            End
        End
        if Maximo>=3750 then
            For i:=-AnchoVentana to Anchoventana do
                suma[i]:=Round(0.01*suma[i]);
            End
        End
    End;

    Procedure IniciaGraficos;
    Begin
        Gd:=Detect;
        InitGraph(Gd,Gm,bgi);
    End;

    Procedure Ventana(v,DespX,DespY:integer);
    Const
        AnchoMenSupe=25;
    Var
        ViewPort          :ViewportType;
    Begin
        SetColor(Brown);
        Case v of
            0: Begin
                SetViewPort(0,AnchoMenSupe+1,GetmaxX,GetmaxY,ClipOn);
                ClearViewPort;
                GetViewSettings(ViewPort);
                with ViewPort do
                    Rectangle(0,0,x2-x1,y2-y1);
                End;
            1: Begin
                SetViewPort(0,0,GetmaxX,AnchoMenSupe,ClipOn);
                GetViewSettings(ViewPort);
                with ViewPort do
                    Rectangle(0,0,x2-x1,y2-y1);
                SetViewPort(1,1,GetmaxX-1,AnchoMenSupe-1,ClipOn);
            End;
            2: Begin
                (SetViewPort( OfsX1-1, OfsY1+DespY-1,
                    GetmaxX+1,OfsY2+DespY+1,ClipOn);
                GetViewSettings(ViewPort);
                SetViewPort( OfsX1, OfsY1+DespY,
                    GetmaxX, OfsY2+DespY,ClipOn);
                GetViewSettings(ViewPort);
                with ViewPort do

```

```

                Begin
                hreal:=(x2-x1)/(128*30);
                30 es el numero de espigas a ver por linea
                FacY:=(y2-y1)/(VoltMax-VoltMin);
                Ty:=(y2-y1) div 2;
                Tx:=0;
                End; }
            End;
        End;
        SetColor(Blue);
    End;

    Procedure Histograma;
    var
        bin, FacY, FacX, Ticks, MarcaTicks: Integer;
        FillInfo: FillSettingsType;

        Procedure Ejes(X, Y: Integer);
        Begin
            SetColor(Blue);
            Line(0, Y, GetMaxX, Y);
            Line(X, 0, X, GetMaxY);
        End;

    Begin
        SetColor(Blue);
        Ty:=4*(GetMaxY div 5);
        Tx:=GetMaxX div 2;
        Bin:=1;
        FacY:=1;
        FacX:=1;
        repeat
            ClearDevice;
            Ventana(1, 0, 0);
            OutTextXY(1, 2, 'Archivo: '+NomEntrad);
            str(LimInf, nom);
            OutTextXY(350, 2, 'Intervalo usado: '+nom+', ');
            str(LimSup, nom);
            OutTextXY(530, 2, nom);
            str(AnchoVentana, nom);
            OutTextXY(1, 15, 'Ventana de calculo: -'+nom+', '+nom);
            str(neu1, nom);
            OutTextXY(350, 15, 'Correlación Neuronas '+nom+' Vs ');
            str(neu2, nom);
            OutTextXY(560, 15, nom);
            Ventana(0, 0, 0);
            Ejes(Tx, Ty);
            SetbkColor(white);
            GetFillSettings(FillInfo);
            SetFillStyle(SolidFill, darkgray);
            Ticks:=0;
            Maximo:=0;

            if Neu1<>Neu2 then
                For i:=-AnchoVentana to AnchoVentana do
                    Begin
                        Bar(i*Bin*FacX+Tx, Ty, (i+1)*bin*FacX+Tx, Ty-Suma[i]*FacY);
                        if suma[i]>maximo then
                            Maximo:=suma[i];
                    End
                else
                    For i:=0 to AnchoVentana do
                        Begin
                            Bar(i*Bin*FacX+Tx, Ty, (i+1)*bin*FacX+Tx, Ty-Suma[i]*FacY);
                            if suma[i]>maximo then
                                Maximo:=suma[i];
                        End;
                    For i:=1 to 3 do
                        Begin
                            Ticks:=Ticks+100;
                            If FacX>1 then
                                MarcaTicks:=Ticks Div FacX
                            else
                                MarcaTicks:=Ticks;
                            str(MarcaTicks, nom);
                            if FacX>=1 then
                                Begin
                                    OutTextXY(Tx+Ticks-13, Ty+6, nom);
                                End;
                            End;
                        End;
                    End;
                End;
            End;
        End;
    End;

```

```

        OutTextXY(Tx-Ticks-13,Ty+6,'-'+nom);
    End
else
    Begin
        OutTextXY(Tx+Ticks-13,Ty+6,'-'+nom);
        OutTextXY(Tx-Ticks-13,Ty+6,nom);
    End;
End;
Maximo:=Maximo*FacY;
Str(Total,nom);
OutTextXY(Tx+4,Ty-Maximo-10,nom);

OutTextXY(2,443,'S para continuar.  '+'Teclas del cursor para cambiar escala.'
          '+' INICIO para regresar');

Repeat
    ch:=Readkey;
until
    (ord(ch) in [72,75,77,80,71]) or (upcase(ch)='S');
case ord(ch) of
    72 :Inc(FacY);
    80 :Dec(FacY);
    77 :Inc(FacX);
    75 :Dec(FacX);
    71 :Begin
        FacY:=1; FacX:=1;
    End;
end;

until
    (upcase(ch)='S');
CloseGraph;
End;

Begin (////////// Programa Principal (XCOR) ////////////)
    ClrScr;
    Gotoxy(10,24);
    PathDatos:=PathDatosConst;
    Write('La ruta por defecto para datos es: ');
    textcolor(lightred);
    Write(PathDatos);
    textcolor(lightgray);
    Gotoxy(45,24);
    Readln(nom);
    if nom<>' ' then
        PathDatos:=nom;
    AbreArchivoEntrada;
    repeat
        LeeDatos;
        CuentaOcurrencias;
        IniciaGraficos;
        Histograma;
        Write('Otro analisis sobre el mismo archivo ? [s/n] ');
        ch:=readkey;
        writeln;
    Until
        upcase(ch)='N';
    Close(dat);
End.◆

```

Apéndice B

Especificaciones tarjeta A/D D/A.

Tabla 7 Especificaciones de la tarjeta DT2821-F-16SE.

MÓDULO A/D	DT5725
ENTRADAS ANALÓGICAS	
Número de entradas analógicas	16 en modo simple, 8 en modo bipolar
Rangos de entradas	0 a +10 V (unipolar) ±10 V (Bipolar)
Codificación de los datos adquiridos	Binario (unipolar) Complemento a dos o binario con desplazamiento (bipolar)
Rangos de ganancias	1, 2, 4, 8
Impedancias de entrada	Canal apagado : 100 MΩ, 10 pF Canal prendido : 100 MΩ, 50 pF
Bias Current	±10 nA
Máximo voltaje de entrada en modo común	±10.5 V
Máximo voltaje de entrada sin daño, Prendido	±27 V
Máximo voltaje de entrada sin daño, Apagado	±12 V
Amplificación del ruido de entrada	0.2 LSB rms
Voltaje de error de canal-a-canal de entrada	±0.1 mV
Precisión	
Resolución	12 bits
No-linealidad	menos de ±1/2 LSB
No-linealidad diferencial	menos de ±1/2 LSB
Error de cuantización inherente	±1/2 LSB
Precisión del sistema	
Ganacia=1	dentro de ±0.03% FSR
Ganacia=2	dentro de ±0.04% FSR
Ganacia=4	dentro de ±0.05% FSR
Ganacia=8	dentro de ±0.07% FSR
Crosstalk canal	-100 dB a 1 kHz
Error de ganancia	Ajustable a cero
Error de cero	Ajustable a cero
DESEMPEÑO DINÁMICO	
Tiempo de adquisición dentro de ±1/2 LSB	7.65 μs
Tiempo de conversión A/D	4 μs
Throughput de convertidor A/D	150 kHz
Throughput A/D a la memoria del sistema	150,000 muestras por segundo.
Incertidumbre en la apertura del Sample and Hold	0.5 ns
Retardo en la apertura del Sample and Hold	100 ns
Rango Droop del Sample and Hold	50 mV/ms
Atenuación Feedthrough del Sample and Hold	80 db a 1 kHz
MODULO D/A	
SALIDAS ANALÓGICAS	
Número de canales	2
Rangos de salida (Seleccionables por jumpers)	0 a 5 V (unipolar) 0 a 10 V (unipolar) ±2.5 V, ±5 V, ±10 V (bipolar)
Codificación de los datos a sacar	Binario (unipolar) Binario con desplazamiento (bipolar)
Throughput	130 kHz max, canal simple 260 kHz max, agregado

Corriente de salida	±5 mA
Impedancia de salida	0.1 Ω max
Capacidad de manejo de capacitancia	0.004 μF
Protección redundante	Cortocircuitado a común analógico
PRECISIÓN	
Resolución	12 bits
No-linealidad	dentro de ±1/2 LSB
No-linealidad bipolar	dentro de ±1/2 LSB
Error de ganancia	Ajustable a cero
Error de cero	Ajustable a cero
DESEMPEÑO DINÁMICO	
Tiempo para poner	5 μs, 20v paso
	1 μs, 100 mV paso
	10 V/μs
Rango de slew	
RELOJ EN LA TARJETA	
Frecuencia base	4.00 MHz ±0.01%
Rango pre-escalable	Potencias de 2 desde 0 hasta 15
Rango de divisor	Valores enteros entre 1 y 256
Rango de frecuencias útiles	250 kHz a 0.5 Hz

Referencias

- [1] Abeles, M.; Goldstein, M.H. "**Multispikes train analysis**", Proc. IEEE, vol. 65, pp. 762-772, 1977.
- [2] Schmidt, E.M. "**Instruments for sorting neuroelectric data: a review**", Journal of Neuroscience, vol. 12, pp. 1-24, 1984
- [3] Schmidt, E.M. "**Computer separation of multi-unit neuroelectric data: a review**", Journal of Neuroscience, vol. 12, pp. 95-111, 1984.
- [4] Vibert, J.V.; Albert, J.N.; Costa, J. "**Intelligent software for spike separation in multiunit recordings**", Med. & Biol. Eng. & Comput., pp. 366-372, July 1987.
- [5] Yang, X.; Shamma, S. "**A totally automated system for the detection and classification of neural spikes**", IEEE Trans. Biomed. Eng., Vol 35, pp.806-816, 1988.
- [6] Kreiter, A.K.; Aertsen, M.H.J.; Gerstein, G.L. "**A low-cost single-board solution for real-time, unsupervised waveform classification of multineuron recordings**", Journal of Neuroscience Methods, Vol. 30, pp. 59-69, 1989.
- [7] Atiya, A. F. "**Recognition of Multiunit Neural Signals**", IEEE Trans. Biomed. Eng., Vol 39, pp.723-729, 1992.
- [8] Roberts, W.M.; Hartline, D.K. "**Separation of multi-unit nerve impulse trains by a multichannel linear filter algorithm**", Brain Research, 94 (1975), pp. 141-149
- [9] Heetderks, W.; Kwan, P. "**A hardware device for multispikes signal detection and sorting by template matching**", Proc. 7th NewEngland Bioeng. conf. Ostrander, L.E. (de) pp. 1-4, 1979.
- [10] Friedman, D.H. "**Separation of signals by template matching**" The Johns Hopkins Press, Baltimore, 1968.
- [11] Blum, N.A.; Carkhuff, B.G.; Charles, H.K.; Edwards R.L.; Meyer, R.A. "**Multisite Microprobes for neural recordings**", IEEE Trans. Biomed. Eng., Vol 38, No. 1, pp.68-74, January 1991.
- [12] Rutten, W.L.C.; Wier, H.J.; Put, J.H.M. "**Sensitivity and selectivity of intraneural stimulation using a silicon electrode array**", IEEE Trans. Biomed. Eng., Vol 38, No.2, pp.192-198, February 1991.
- [13] Glaser, E.M.; Ruchkin, D.S. "**Principles of neurobiological signal analysis**", Academic Press, 1976.
- [14] Serna, H.R.; Santamaría P.F.; Domínguez, S.I.; Espinosa, E.I. "**Integración y validación de un sistema de adquisición y procesamiento de señales neurofisiológicas**". Memorias del SOMI XI, Congreso de Instrumentación, Sept. 1994, pag. 98-101.
- [15] Serna, H.R.; Austrich, J.S.; Espinosa, E.I. "**Reconocimiento, mediante programa, de espigas en registros neurofisiológicos compuestos**". Revista Mexicana de Ingeniería Biomédica Vol 13(2) No.1 pag. 353-356 Septiembre 1992.
- [16] Ayhlón, M.A.; Espinosa, E.I. "**Resultados preliminares para un sistema de detección de señales neuroeléctricas**", Rev. Mex. Ing. Biomed. 11(1) pp. 169-182, Nov. 1990.

- [17] Tuckwell, H.C. ***“Introduction to theoretical neurobiology: Volume I linear cable theory and dendritic structure”***, Cambridge University Press, USA, 1988.
- [18] Mead, C. ***“Analog VLSI and neural systems”***, Addison-Wesley Publishing Co., USA, 1989.
- [19] Freeman, J.A.; Skapura, D.M. ***“Redes neuronales algoritmos, aplicaciones y técnicas de programación”***, Addison-Wesley Iberoamericana, S.A., USA, 1993.
- [20] Pasantes, H.; Sánchez, J.; Tapia, R. ***“Neurobiología Celular”***, Fondo de Cultura Económica, S.A. de C.V., México, 1991.
- [21] Tapia, R. ***“Las células de la mente”***, Fondo de Cultura Económica, S.A. de C.V., México, 1987.
- [22] Avalos, V.M.; Leonetti, G.N; Alcántara, M.; Espinosa, E.I. ***“Estudio comparativo de métodos para el análisis de conectividad neuronal”***, Rev. Mex. Ing. Biomed. 11(1) pp. 109-123, Nov. 1990.
- [23] Gerstein, G.L.; Bloom, M.J.; Espinosa, E.I.; Evanczuk, S.; Turner, M. ***“Design of a laboratory for multineuron studies”***, IEEE Trans. Systems, Man and Cybernetics, SMC-13, pp. 668-676, 1983.
- [24] Alcántara, M.A.G. ***“Neurored: Simulador y analizador de redes neuronales artificiales tipo biológico”***, Tesis Licenciatura, Ingeniero en Computación, Fac. de Ing., UNAM, 1992.
- [25] McCulloch, W.S.; Pitts W. ***“A logical calculus of the ideas immanent in nervous activity”***, Bulletin of Mathematical Biophysics, pp 115-133, 1943.
- [26] Calvin, W.H. ***“Generation of spike trains in CNS neurons”***, Brain Res., Vol. 84, pp. 1-22, 1975.
- [27] Quiza, T.J. ***“Clasificación de potenciales de acción con una red neuronal que emplea retropropagación”***, Tesis Maestría en Ing. Biomédica, UAM-Iztapalapa, 1991.

Índice de figuras

Figura 1 Disposición física del montaje experimental para el registro de señales neurofisiológicas	5
Figura 2 Ejemplo de un pantalla de adquisición de Discovery.	6
Figura 3 Ejemplo de una señal real, tres tramos de 20 ms	7
Figura 4 Esquema de la estructura de una neurona biológica. [19]	10
Figura 5 Generación de un potencial de acción. [19]	11
Figura 6 Descripción de la región sináptica.	12
Figura 7 Detalles del potencial de acción de una neurona.	13
Figura 8 Parámetros de la forma de onda de un potencial de acción.	13
Figura 9 Representación gráfica del ajuste por mínimos cuadrados. Se hace la aproximación al vector medido $\langle x$ con la combinación lineal de dos vectores ortonormales $\langle \tilde{g}_1$ y $\langle \tilde{g}_2$. La mejor aproximación se da cuando el vector de error E , es perpendicular al plano generado por los vectores ortonormales.	16
Figura 10 Representación gráfica de los componentes principales.	17
Figura 11 Segunda pantalla de UNE-VEC.	25
Figura 12 Información de UNE-VEC al generar la señal.	26
Figura 13 Formas de onda de las espigas de la base de datos usada.	27
Figura 14 Un tramo de una señal sintética.	27
Figura 15 Superposición de las formas de onda usadas en una señal sintética.	28
Figura 16 Formato para el archivo de entrada a XCOR.	29
Figura 17 Información necesaria para el módulo XCOR.	30
Figura 18 Generación del histograma de correlación. Contribución del primer elemento de la serie 1.	31
Figura 19 Generación del histograma de correlación. Contribución del i -ésimo elemento de la serie 1.	31
Figura 20 Ejemplo de correlación. En el recuadro superior aparece la información del archivo usado.	32
Figura 21 Pseudocódigo del módulo XCOR que sirve para obtener correlación cruzada.	33
Figura 22 Archivo para calibración. Espigas con distancia 0 ms.	36
Figura 23 Archivo para calibración. Espigas con distancia 1 ms.	36
Figura 24 Esquema de tareas para las simulaciones.	37
Figura 25 Diagrama de conexionamiento para las neuronas usadas para la simulación de excitación.	38
Figura 26 Excitación. Un tramo de la señal sintética para dos neuronas.	39
Figura 27 Excitación. Formas de onda usadas para el ejemplo de excitación.	39
Figura 28 Excitación. Correlación cruzada entre las dos neuronas del archivo EXCIT4.SAL. Antes de GESSIN.	39
Figura 29 Excitación. Autocorrelación de la neurona 1. Antes de GESSIN.	40
Figura 30 Excitación. Autocorrelación de la neurona 2. Antes de GESSIN.	40
Figura 31 Excitación. Correlación cruzada entre las dos neuronas del archivo EXCIT4.NRD. Después de GESSIN.	40
Figura 32 Excitación. Autocorrelación de la neurona 1. Después de GESSIN.	40
Figura 33 Excitación. Autocorrelación de la neurona 2. Después de GESSIN.	40
Figura 34 Excitación. Edición de cúmulos para la separación de las espigas.	41
Figura 35 Excitación. Formas de onda de las espigas separadas, (comparar con las originales en la Figura 27).	41
Figura 36 Excitación. Correlación cruzada para las neuronas del archivo EXCIT4.UFF y representadas por los cúmulos de la Figura 34.	41
Figura 37 Diagrama de conexionamiento para las neuronas usadas para la simulación de inhibición.	42
Figura 38 Inhibición. Formas de onda usadas para el ejemplo de Inhibición.	43
Figura 39 Inhibición. Correlación cruzada entre las dos neuronas del archivo INHIB1.SAL. Antes de GESSIN43	43
Figura 40 Inhibición. Autocorrelación de la neurona 1. Antes de GESSIN.	43
Figura 41 Inhibición. Autocorrelación de la neurona 2. Antes de GESSIN.	43

Figura 42 Inhibición. Correlación cruzada entre las dos neuronas del archivo INHIB1.NRD. Después de GESSIN	44
Figura 43 Inhibición. Autocorrelación de la neurona 1. Después de GESSIN.	44
Figura 44 Inhibición. Autocorrelación de la neurona 2. Después de GESSIN.	44
Figura 45 Inhibición. Edición de cúmulos para la separación de las espigas	45
Figura 46 Inhibición. Formas de onda de las espigas separadas	45
Figura 47 Inhibición. Correlación cruzada para las neuronas del archivo INHIBE1.UFF	45
Figura 48 Inhibición. Autocorrelación de la neurona 1 del archivo INHIBE1.UFF	45
Figura 49 Diagrama de conexiónamiento para las neuronas usadas para la simulación de independencia.	45
Figura 50 Independencia. Correlación cruzada entre las dos neuronas del archivo INDEP2.SAL. Antes de GESSIN	46
Figura 51 Independencia. Autocorrelación de la neurona 1. Antes de GESSIN.	46
Figura 52 Independencia. Autocorrelación de la neurona 2. Antes de GESSIN.	46
Figura 53 Independencia. Correlación cruzada entre las dos neuronas del archivo INDEP2.NRD. Después de GESSIN	46
Figura 54 Independencia. Autocorrelación de la neurona 1. Después de GESSIN.	47
Figura 55 Independencia. Autocorrelación de la neurona 2. Después de GESSIN.	47
Figura 56 Independencia Edición de cúmulos para la separación de las espigas.	47
Figura 57 Independencia. Formas de onda de las espigas separadas	47
Figura 58 Independencia. Correlación cruzada para las neuronas del archivo INDEP2.UFF	47
Figura 59 Diagrama de conexiónamiento para las neuronas usadas para la simulación de oscilación.	48
Figura 60 Oscilación. Formas de onda usadas para el ejemplo de oscilación	49
Figura 61 Oscilación. Correlación cruzada entre las neuronas 1 y 2 del archivo OSCII.SAL. Antes de GESSIN	49
Figura 62 Oscilación. Autocorrelación de la neurona 1. Antes de GESSIN.	49
Figura 63 Oscilación. Edición de cúmulos para la separación de las espigas.	50
Figura 64 Oscilación. Formas de onda de las espigas separadas	50
Figura 65 Oscilación. Correlación cruzada para las neuronas 1 Vs 2 del archivo OSCII.UFF. (ventana -400, 400 ms)	50
Figura 66 Oscilación. Correlación cruzada para las neuronas 3 Vs 4 del archivo OSCII.UFF (ventana -400, 400 ms)	50
Figura 67 Oscilación Correlación cruzada para las neuronas 1 Vs 2 del archivo OSCII.UFF (ventana -600, 600 ms)	50
Figura 68 Oscilación. Autocorrelación neurona 1.	50
Figura 69 Diagrama de conexiónamiento de una red con cinco neuronas.	51
Figura 70 (5 Neuronas) Formas de onda usadas para el ejemplo de 5 neuronas.	52
Figura 71 (5 Neuronas) Correlación cruzada 1 Vs 2. Antes de usar GESSIN.	53
Figura 72 (5 Neuronas) Correlación cruzada 1 Vs 2. Después de usar GESSIN.	53
Figura 73 (5 Neuronas) Correlación cruzada 1 Vs 3. Antes de usar GESSIN.	53
Figura 74 (5 Neuronas) Correlación cruzada 1 Vs 3. Después de usar GESSIN.	53
Figura 75 (5 Neuronas) Correlación cruzada 1 Vs 4. Antes de usar GESSIN.	53
Figura 76 (5 Neuronas) Correlación cruzada 1 Vs 4. Después de usar GESSIN.	53
Figura 77 (5 Neuronas) Correlación cruzada 1 Vs 5. Antes de usar GESSIN.	54
Figura 78 (5 Neuronas) Correlación cruzada 1 Vs 5. Después de usar GESSIN.	54
Figura 79 (5 Neuronas) Correlación cruzada 2 Vs 3. Antes de usar GESSIN.	54
Figura 80 (5 Neuronas) Correlación cruzada 2 Vs 3. Después de usar GESSIN.	54
Figura 81 (5 Neuronas) Correlación cruzada 2 Vs 4. Antes de usar GESSIN.	55
Figura 82 (5 Neuronas) Correlación cruzada 2 Vs 4. Después de usar GESSIN.	55
Figura 83 (5 Neuronas) Correlación cruzada 2 Vs 5. Antes de usar GESSIN.	55
Figura 84 (5 Neuronas) Correlación cruzada 2 Vs 5. Después de usar GESSIN.	55
Figura 85 (5 Neuronas) Correlación cruzada 3 Vs 4. Antes de usar GESSIN.	56
Figura 86 (5 Neuronas) Correlación cruzada 3 Vs 4. Después de usar GESSIN.	56
Figura 87 (5 Neuronas) Correlación cruzada 3 Vs 5. Antes de usar GESSIN.	56
Figura 88 (5 Neuronas) Correlación cruzada 3 Vs 5. Después de usar GESSIN.	56
Figura 89 (5 Neuronas) Correlación cruzada 4 Vs 5. Antes de usar GESSIN.	57
Figura 90 (5 Neuronas) Correlación cruzada 4 Vs 5. Después de usar GESSIN.	57
Figura 91 (5 Neuronas) Edición de cúmulos para la separación de espigas, archivo REDS2F.UFF	57

Figura 92 (5 Neuronas) Formas de onda de las espigas separadas. Archivo REDS2F.UFF.	57
Figura 93 (5 Neuronas) Correlación cruzada 3 vs 4 correspondiente a la 1 Vs 2 original.	58
Figura 94 (5 Neuronas) Correlación cruzada 3 vs 5 correspondiente a la 1 Vs 3 original.	58
Figura 95 (5 Neuronas) Correlación cruzada 1 vs 3 correspondiente a la 4 Vs 1 original.	58
Figura 96 (5 Neuronas) Correlación cruzada 2 vs 3 correspondiente a la 5 Vs 1 original.	58
Figura 97 (5 Neuronas) Correlación cruzada 4 vs 5 correspondiente a la 2 Vs 3 original.	59
Figura 98 (5 Neuronas) Correlación cruzada 2 vs 4 correspondiente a la 5 Vs 2 original.	59
Figura 99 (5 Neuronas) Correlación cruzada 1 vs 4 correspondiente a la 4 Vs 2 original.	59
Figura 100 (5 Neuronas) Correlación cruzada 1 vs 5 correspondiente a la 4 Vs 3 original.	59
Figura 101 (5 Neuronas) Correlación cruzada 2 vs 5 correspondiente a la 3 Vs 5 original.	60
Figura 102 (5 Neuronas) Correlación cruzada 1 vs 2 correspondiente a la 4 Vs 5 original.	60

Índice alfabético

A

adquisición · 3, 5, 6, 7, 14, 20, 26, 55, 83, 85
 amplitud · 14, 18, 19
 análisis espectral · 16
 analógica · 4, 5, 18, 19, 21, 26, 33, 34, 35, 39, 42, 48, 49, 55, 83
 archivo · 21, 22, 23, 24, 26, 27, 28, 30, 31, 33, 34, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 50, 55, 61, 63, 64, 65, 66, 70, 71, 72, 74, 77, 78, 81
 ASCII · 23, 33
 Australia · 20
 autocorrelación · 15, 17, 27, 30, 33, 42, 43, 46, 47, 49
 axón · 9, 10, 11, 12

B

banda
 ancho de · 15, 20
 base · 4, 9, 13, 15, 17, 19, 21, 23, 24, 25, 49, 51, 64, 73, 84

C

canal · 5, 6, 20, 83
 célula · 9, 10, 11, 12, 86
 cerebro · 3, 5, 9
 Cibernética
 Laboratorio de · 3, 7, 19, 60, 61
 circuito · 3, 4, 9, 14, 23, 33, 43, 46, 47, 49, 60
 componentes principales · 13, 15, 17, 19
 computadora · 5, 19, 21, 60
 conectividad · 3, 33, 40, 49, 60, 86
 conexión · 3, 4, 33, 36, 40, 43, 45, 46, 49, 51, 52, 53, 54, 57, 59
 conversión · 3, 4, 21, 26, 42, 56, 60, 83
 convertidor
 A/D D/A · 18, 20, 83
 convolución · 16

Correlac · 27, 33, 34, 37, 38, 39, 41, 42, 43, 44, 45, 47, 48, 51, 52, 53, 54, 55, 56, 57, 58, 80
 correlación · 6, 15, 17, 18, 20, 21, 26, 27, 28, 29, 30, 31, 33, 34, 35, 37, 38, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 55, 56, 57
 cúmulos · 39, 42, 43, 45, 48, 49, 55

D

DAC · 21, 22, 26, 34, 39, 42, 48, 55, 60, 73, 74
 DAC-ESP · 39
 DAD
 Discriminador de Amplitud Diferencial · 18, 73, 74
 datos · 19, 21, 22, 23, 24, 25, 26, 27, 28, 30, 31, 33, 36, 39, 40, 44, 46, 50, 63, 72, 73, 74, 75, 78, 81, 83
 dendritas · 9
 dendríticos · 9
 despolarización · 11, 12
 detectores · 18
 digital · 5, 6, 14, 17, 19, 26, 34, 37, 39, 49
 digitales · 17
 Discovery · 3, 4, 6, 7, 13, 16, 19, 21, 24, 26, 33, 34, 35, 39, 40, 42, 43, 46, 49, 50, 55, 56, 57, 59
 discriminador · 19
 discriminadores · 18, 19
 disparo · 4, 18, 23, 24, 26, 28, 33, 34, 36, 37, 39, 40, 41, 43, 44, 45, 46, 47, 50, 59, 63, 65, 70, 71, 77
 DOS · 22
 DSP · 14, 20

E

electrodo · 3, 5, 60
 electrofisiológico · 3, 4
 electrónica · 3, 14, 18, 19
 espiga · 3, 4, 6, 7, 9, 13, 14, 15, 17, 18, 19, 20, 21, 23, 24, 25,

26, 33, 34, 39, 42, 43, 45, 46, 48, 49, 50, 55, 60, 63, 64, 67, 71, 72, 75, 79, 85
 Estados Unidos · 20
 estereotáxico · 5
 estocásticos
 procesos · 17
 Excitación · 33, 35, 36, 37, 38, 39, 46, 54, 56, 57
 excitatorio · 11, 12
 extracelular · 3, 4, 10

F

Faraday · 60
 fibra · 11, 36, 37, 40, 43, 46, 51, 52
 filtro · 17
 fisiología · 20
 Fourier
 Transformada rápida · 6, 15, 20
 frecuencia · 5, 11, 15, 16, 20, 84

G

GENERA · 22
 GESSIN · 4, 5, 21, 22, 23, 33, 34, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 50, 51, 52, 53, 54, 55, 57, 60
 gráfica · 7, 13, 16, 17, 28, 30, 36, 39, 46, 56

H

hardware · 6, 18, 19, 20, 60, 85
 hiperpolarizante · 12
 hipocampo · 6
 histograma · 20, 21, 27, 28, 29, 30, 31, 33, 35, 37, 39, 40, 41, 42, 43, 46, 47, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58

I

impulso
 respuesta al · 3, 11, 12, 17
 impulsos eléctricos · 3

Independencia · 33, 43, 44, 45, 53, 54, 58
Inhibición · 33, 40, 41, 42, 43, 53, 57, 59
inhibitorio · 11, 12
instrumental · 60
intracelular · 10, 11
ion · 10, 12, 14
isomorfa · 16
Israel · 20

J

Jerusalén · 20

K

Karhunen-Loève · 17

L

lenguaje
de programación · 21
lógica · 3, 4, 5, 10, 11, 15, 18, 20, 21, 26, 33, 34, 35, 39, 42, 48, 49, 55, 60, 83, 85

M

matriz
de correlación · 15, 17
membrana · 10, 12
microelectrodo · 3, 5
micromanipulador · 5
microprocesador · 18
mielina · 10
mínimos cuadrados · 16
MSD
Detector Multi-Espigas · 20

N

nervioso
impulso
sistema · 3, 9, 11
neurociencias · 61
neurofisiológica · 5, 15, 20, 60, 85
neurona · 3, 4, 6, 9, 10, 12, 14, 17, 18, 23, 24, 26, 28, 30, 31, 33, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 65, 70, 86
neuronal

conectividad · 3, 36, 60
Neurored · 4, 21, 23, 26, 27, 28, 31, 33, 34, 36, 37, 38, 39, 42, 46, 48, 50, 55, 59, 71, 72, 86
neurotransmisores · 12
Nyquist · 15

O

octal · 23
onda
forma de · 3, 6, 7, 13, 14, 19, 20, 25, 26, 37, 39, 41, 42, 43, 45, 46, 47, 48, 49, 50, 55, 70
ortonormal · 15, 16, 17
Oscilación · 33, 46, 47, 48, 49
osciloscopio · 7, 26

P

pantalla · 5, 6, 7, 23, 24, 25, 26, 30, 72
Pascal · 21, 26, 27, 36
patrón · 19, 63, 64, 72
PC · 5, 20, 60, 73, 74
plantilla · 13, 15, 16, 19, 20
polarización · 11, 12
postsinápticas · 12
potencial
de acción · 3, 4, 6, 9, 10, 11, 12, 13, 14, 19, 23, 24, 26, 60
presináptica · 12
procesador · 5, 18, 21, 22, 23
programa · 4, 19, 21, 22, 23, 24, 25, 26, 27, 28, 30, 33, 36, 39, 42, 48, 61, 64, 73, 77, 85, 86

R

radiofrecuencia · 60
RAM · 5, 21, 22, 65, 74, 77
Ranvier · 10, 11
red neuronal
biológica · 3, 36, 60
registro · 3, 4, 5, 14, 18, 19, 23, 60, 63, 64, 85

S

señal
traza de la
eléctrica · 3, 4, 5, 6, 7, 10, 13, 15, 16, 17, 18, 19,

20, 21, 23, 24, 25, 26, 27, 33, 34, 37, 38, 39, 40, 42, 43, 46, 48, 49, 50, 55, 60, 64, 85
traza de la · 25, 26, 37, 55, 59
serie · 9, 11, 24, 27, 28, 29, 30, 31, 37, 41, 77, 78, 79
simulación · 23, 33, 34, 35, 36, 37, 39, 40, 42, 43, 46, 49, 59
simulador · 3, 4, 5, 6, 21, 22, 26, 27, 33, 34, 60, 61
sinapsis · 9, 10, 11
SNC
Sistema Nervioso Central · 9
software · 3, 6, 18, 19, 20, 34, 60, 85
soma · 9, 10, 11

T

texto · 9, 13, 14, 23, 27, 56, 77
traslapo · 26, 33, 35, 38, 40, 46, 49, 50, 52, 59, 60

U

umbral · 3, 7, 12, 14, 18, 19
UNE-VEC · 22, 23, 24, 26, 28, 33, 34, 36, 40, 45, 50, 64, 72
UNIX · 27

V

valores propios · 17
vectores · 15, 16, 17, 18, 23, 27, 64
vectores propios · 17
ventana
de tiempo · 3, 7, 14, 18, 19, 21, 28, 30, 40, 48, 49, 67, 68, 69, 78, 79
VGA · 22
VSLI · 14

X

XCOR · 22, 26, 27, 28, 31, 33, 34, 40, 41, 42, 56, 61, 77, 81