

UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR



PROYECTO FIN DE CARRERA

Ingeniería de Telecomunicación

**MODELOS DE NEURONAS ARTIFICIALES EN
SOFTWARE PARA SU USO EN
PREPARACIONES DE ELECTROFISIOLOGÍA**

Jenniffer Cubillos Martínez

Febrero 2016

MODELOS DE NEURONAS ARTIFICIALES EN SOFTWARE PARA SU USO EN PREPARACIONES DE ELECTROFISIOLOGÍA

**Autor: Jenniffer Cubillos Martínez
Tutor: Pablo Varona Martínez**

Grupo de Neurocomputación biológica (GNB)
Departamento de Ingeniería Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid

Febrero de 2016

PROYECTO FIN DE CARRERA

TITULO:

Modelos de neuronas artificiales en software para su uso en preparaciones de electrofisiología

Resumen:

Este proyecto tiene por objetivo la implementación y validación online o en tiempo real de circuitos híbridos compuestos por neuronas vivas de sistemas biológicos y neuronas electrónicas implementadas en software. En estos circuitos híbridos las neuronas vivas y las neuronas artificiales interactúan bidireccionalmente a través de sinapsis artificiales.

El desarrollo de estos circuitos híbridos es de gran interés en el contexto de la investigación en neurociencia ya que permiten caracterizar la dinámica de neuronas y redes neuronales biológicas. Además, este tipo de circuitos facilitan la observación de aspectos del comportamiento de la actividad neuronal, permitiendo mejorar el control de la actividad irregular, lo cual no sería posible de llevar a cabo con las técnicas neurofisiológicas tradicionales.

Para llevar a cabo la validación del proyecto se implementaron en software modelos neuronales capaces de producir un comportamiento de ráfagas de disparo en la variable que representa el potencial de membrana. Durante el desarrollo de los circuitos híbridos, se tuvo en cuenta las restricciones temporales impuestas por la neurona viva, las particularidades del registro y escalado de la conexión bidireccional, adaptando de manera general el funcionamiento de los modelos para que fuera posible su implementación utilizando distintas plataformas de simulación.

La comunicación bidireccional se llevó a cabo en tiempo real estricto, por lo que se utilizó un sistema operativo linux con la extensión de RTAI, además del uso de las librerías de COMEDI, para gestionar el control de la tarjeta de adquisición de datos que permitió la interacción entre el mundo biológico y el digital.

Las implementaciones de los circuitos híbridos se validaron en el circuito generador central de patrones del crustáceo *Carcinus maenas*. Mediante una conexión eléctrica se consiguió sincronizar la actividad de los modelos de neuronas artificiales con el ritmo del circuito neuronal en función del valor de la conductancia de acoplamiento.

Palabras clave:

Circuitos híbridos, ciclo cerrado, modelos neuronales, sinapsis eléctrica artificial, COMEDI, RTAI, Software de tiempo real.

Summary:

This project aims at the implementation and online or real-time validation of hybrid circuits composed by living neurons in biological systems and electronic neurons implemented in software. In these hybrid circuits living neurons and artificial neurons interact bidirectionally across an artificial synapse.

The development of these hybrid circuits is of great interest in the context of neuroscience research as to characterize the dynamics of biological neurons and neural networks. In addition, they facilitate the observation of crucial aspects of neuronal dynamics, allowing better control of irregular activity, which would not be possible to accomplish with traditional neurophysiological techniques.

To validate the project, neural models capable of producing spiking-bursting activity in the variable representing the membrane potential were implemented in software. During the development of the hybrid circuits, the time constraints imposed by the living neuron, and the factors involved in the recording and scaling of the bidirectional connection were taken into account, generally adapting the operation of models to allow the implementation using different simulation platforms.

Bidirectional communication was carried out strictly in real time, so a Linux operating system was used with the RTAI extension, and the use of COMEDI libraries, to manage the data acquisition that allowed the interaction between the biological and digital counterparts.

The implementations of hybrid circuits were validated in a central pattern generator circuit of the crustacean *Carcinus maenas*. It was possible to synchronize the activity of the artificial neural models with the rhythm of the neural circuitry by adapting the value of the conductance coupling using an electrical connection.

Keywords:

Hybrid circuits, closed loop neural models, artificial electrical synapse, COMEDI, RTAI, Real time software.

ÍNDICE DE CONTENIDOS

1.	INTRODUCCIÓN.....	14
1.1.	MOTIVACIÓN	14
1.2.	OBJETIVOS	15
1.3.	ORGANIZACIÓN DE LA MEMORIA.....	16
2.	ESTADO DEL ARTE	17
2.1.	EL SISTEMA NERVIOSO	17
2.1.1.	INTRODUCCIÓN	17
2.1.2.	LA NEURONA	17
2.1.3.	COMUNICACIÓN NEURONAL	18
2.1.4.	MODELO ELÉCTRICO DE UNA MEMBRANA NEURONAL	22
2.2.	MODELOS NEURONALES	23
2.3.	ESTIMULACIÓN NEURONAL EN CICLO CERRADO.....	25
2.4.	CIRCUITOS HÍBRIDOS	26
3.	HARDWARE Y SOFTWARE UTILIZADO.....	29
3.1.	CARACTERÍSTICAS DE LOS PC'S UTILIZADOS	29
3.1.1.	PC PERSONAL PARA DESARROLLOS INICIALES	29
3.1.2.	PC DEL LABORATORIO DEL GNB.....	29
3.2.	TARJETAS DE ADQUISICIÓN DE DATOS (DAQ).....	30
3.3.	REAL TIME APPLICATION INTERFACE (RTAI)	31
3.4.	COMEDI	32
3.5.	LA NEURONA ELECTRÓNICA	33
3.6.	NEURONA VIVA	34
4.	DISEÑO Y DESARROLLO.....	36
4.1.	MODELOS NEURONALES CANDIDATOS A SER IMPLEMENTADOS EN SOFTWARE ...	36
4.1.1.	MODELO HODGKIN-HUXLEY	37
4.1.2.	MODELO DE CONDUCTANCIA DE THOMAS NOWOTNY et al.....	40
4.1.3.	MODELO FITZHUGH-NAGUMO	43
4.1.4.	MODELO HINDMARSH-ROSE	44
4.1.5.	MAPA DE RULKOV	44
4.1.6.	MODELO IZHIKEVICH.....	45
4.2.	MÉTODOS DE INTEGRACIÓN NUMÉRICA DE LOS MODELOS.....	47
4.2.1.	MÉTODO DE EULER	47
4.2.2.	MÉTODO DE EULER MEJORADO	49
4.2.3.	MÉTODO RUNGE-KUTTA.....	49
4.3.	DISEÑO INICIAL DE LOS MODELOS NEURONALES	50
4.3.1.	PLANTEAMIENTOS INICIALES	51
4.3.2.	PROCESO DESARROLLADO	52
4.3.3.	CASO ESPECIAL MAPA DE RULKOV.....	61
4.4.	COMPARACIÓN DE LAS SIMULACIONES EN DIFERENTES EQUIPOS.....	64
4.5.	CONCLUSIONES.....	66

5.	INTEGRACIÓN, PRUEBAS Y RESULTADOS.....	67
5.1.	ADAPTACIÓN DE LOS MODELOS A RTAI Y COMEDI	68
5.2.	RESULTADOS OBTENIDOS DE LAS SIMULACIONES UTILIZANDO RTAI Y COMEDI	70
5.3.	INTERACCIÓN BIDIRECCIONAL ENTRE LA NEURONA SOFTWARE Y LA NEURONA ELECTRÓNICA. COMUNICANDO NEURONAS.....	71
5.3.1.	MODELO MATEMÁTICO PARA LA COMUNICACIÓN ENTRE NEURONAS REPRESENTADAS POR EL MODELO HINDMARSH-ROSE	72
5.3.2.	RESULTADOS OBTENIDOS DE LA INTERACCIÓN BIDIRECCIONAL UTILIZANDO EL MODELO DE HINDMARS-ROSE	74
5.3.3.	INTERACCIÓN BIDIRECCIONAL UTILIZANDO EL MODELO DE IZHIKEVICH	78
5.3.4.	INTERACCIÓN BIDIRECCIONAL UTILIZANDO EL MAPA DE RULKOV	80
5.4.	LA COMUNICACIÓN BIDIRECCIONAL SE MANTIENE SIN EL USO DE RTAI.....	81
5.5.	INTERACCIÓN BIDIRECCIONAL ENTRE LA NEURONA SOFTWARE Y LA NEURONA VIVA. CIRCUITOS HÍBRIDOS	83
5.5.1.	IMPLEMENTACIÓN DEL CIRCUITO HÍBRIDO	83
5.5.2.	RESULTADOS OBTENIDOS EN LAS PRUEBAS DEL CIRCUITO HÍBRIDO FORMADO POR EL MODELO HINDMARSH-ROSE Y NEURONA VIVA	85
5.5.3.	RESULTADOS OBTENIDOS EN LAS PRUEBAS DEL CIRCUITO HÍBRIDO FORMADO POR EL MODELO IZHIKEVICH Y NEURONA VIVA.....	90
6.	CONCLUSIONES.....	95
7.	TRABAJOS FUTUROS	98
8.	REFERENCIAS	99
9.	GLOSARIO	104
10.	ANEXO A	105
10.1.	MODELO HODGKIN-HUXLEY.....	105
10.2.	MODELO FITZHUGH NAGUMO	107
10.3.	MODELO HINDMARSH-ROSE.....	109
10.4.	MAPA DE RULKOV.....	111
10.5.	MODELO IZHIKEVICH.....	112
11.	ANEXO B	116
11.1.	FUNCIONES EXTERNAS.....	116
11.1.1.	FUNCIÓN GETTIMEOFDAY()	116
11.1.2.	FUNCIÓN NANOSLEEP().....	118
12.	ANEXO D.....	120
12.1.	PRESUPUESTO.....	120
12.2.	PLIEGO DE CONDICIONES	121
12.3.	CONDICIONES GENERALES	121
12.4.	CONDICIONES PARTICULARES	124

ÍNDICE DE FIGURAS

<i>Figura 2.1. Diagrama que presenta la estructura básica de una neurona, indicando las tres partes más relevantes de la misma.</i>	<i>18</i>
<i>Figura 2.2. Cuando el potencial de membrana aumenta su valor superando el umbral elegido, comienza una fase de despolarización hasta alcanzar un valor pico a partir del cual disminuye su valor, entrando en la fase de despolarización hasta llegar a la zona de reposo.</i>	<i>19</i>
<i>Figura 2.3. El cambio rápido desde despolarización a hiperpolarización produce un potencial de acción o spike. El conjunto de varios potenciales de acción en un corto intervalo de tiempo es denomina ráfaga o burst.....</i>	<i>20</i>
<i>Figura 2.4. Componentes de una sinapsis química.....</i>	<i>20</i>
<i>Figura 2.5. Modelo eléctrico del comportamiento pasivo de una membrana neuronal con canales de Na+, K+ y Cl-.....</i>	<i>22</i>
<i>Figura 2.6. Representación del canal de potasio de una neurona.....</i>	<i>23</i>
<i>Figura 3.1. Tarjeta de Adquisición de datos similar a la utilizada en el desarrollo del proyecto. Esta imagen corresponde a la tarjeta de un PC fuera de servicio del laboratorio del GNB.....</i>	<i>30</i>
<i>Figura 3.2. Neurona electrónica del laboratorio GNB conectada a la neurona software a través de la tarjeta de adquisición (DAQ).....</i>	<i>33</i>
<i>Figura 3.3. Dibujo del Carcinus maenas (cangrejo común), el cangrejo utilizado para extraer muestras del sistema nervioso para las validaciones del modelo de Hindmarsh-Rose y del modelo de Izhikevich, llevado a cabo en el laboratorio del GNB</i>	<i>34</i>
<i>Figura 3.4.Preparación in vitro de la muestra del sistema nervioso obtenida del Carcinus Maenas.</i>	<i>34</i>
<i>Figura 3.5. La imagen de la izquierda presenta la preparación in vitro de la muestra del sistema nervioso extraída del Carcinus maenas bajo el microscopio de electrofisiología. Se observan dos electrodos conectados a la muestra. La imagen de la derecha muestra el microscopio de electrofisiología del laboratorio del GNB, bajo él se encuentra la muestra del sistema nervioso conectada a los electrodos.</i>	<i>35</i>
<i>Figura 4.1. Circuito eléctrico equivalente al modelo propuesto por Hodgkin y Huxley</i>	<i>37</i>
<i>Figura 4.2. Circuito eléctrico equivalente de Nagumo.....</i>	<i>43</i>
<i>Figura 4.3. Resultados obtenidos de la simulación del Mapa de Rulkov con el método de interpolación lineal y muestreo de puntos para generar 10.000 puntos en un tiempo de un segundo.....</i>	<i>63</i>
<i>Figura 5.1. La imagen muestra el comportamiento en ráfagas que deben tener los modelos neuronales implementados en el proyecto.....</i>	<i>67</i>
<i>Figura 5.2. Se realiza la simulación del modelo de Hindmarsh-Rose utilizando el método de Euler con un paso de integración fijo de 0,001. La gráfica permite comparar que el uso de la función Gettimeofday () o la función rt_get_time() no cambia los resultados en el cálculo de tiempo de ejecución del modelo.</i>	<i>69</i>
<i>Figura 5.3. Comparación de los resultados entre la simulación de código implementado para su ejecución en sistemas RTAI, gráfica izquierda, con los resultados obtenidos de la simulación en un sistema GPOS, gráfica derecha.</i>	<i>70</i>

<i>Figura 5.4. Resultado de la simulación del modelo Hindmarsh-Rose de una ráfaga de 10.000 puntos en un tiempo determinado de un segundo</i>	71
<i>Figura 5.5. Neurona electrónica del laboratorio del GNB desarrollada a partir del modelo matemático de neurona de Hindmarsh-Rose</i>	72
<i>Figura 5.6. Diagrama que representa el circuito entre la neurona software y la neurona electrónica, conectadas a través de la tarjeta de adquisición de datos del laboratorio del GNB. Se añade al sistema el valor de una corriente, por lo tanto se genera una sinapsis eléctrica artificial entre ambas neuronas</i>	74
<i>Figura 5.7. Representación de la sincronización resultado de la comunicación bidireccional entre la neurona implementada en software (gráfica roja) y la neurona electrónica (gráfica azul) durante 10 segundos utilizando el mismo valor de conductancia, $g=3$, en ambas direcciones de la comunicación</i>	77
<i>Figura 5.8. Representación del acoplamiento resultado de la comunicación bidireccional entre la neurona implementada en software (gráfica roja) y la neurona electrónica (gráfica azul) durante 10 segundos con $g=1$, se utiliza el mismo valor de conductancia en ambas direcciones</i>	79
<i>Figura 5.9. Representación inicial del circuito híbrido con el Mapa de Rulkov. En la imagen se observan los resultados con valor de acoplamiento $g=0$. Se han ajustan los valores de Ampli y offset de la neurona software con objeto de sincronizarla con la neurona electrónica. En rojo se representa el potencial de membrana del modelo y en azul el de la neurona electrónica</i>	80
<i>Figura 5.10. Amplificador de señal del laboratorio del GNB</i>	83
<i>Figura 5.11. Valores iniciales de la simulación del circuito híbrido con el modelo de Izhikevich, sin ajustes de parámetros ni cambio de ejes. En rojo se representa el potencial de membrana del modelo y en azul el de la neurona viva. $gmv=0,0$ $gvm=0,0$ $Offset=0,0$ $Ampli=0,0$</i>	91
<i>Figura 9.1 Comportamiento del oscilador de Van der Pol, caso $a=b=0$</i>	107
<i>Figura 9.2. Representación de dos ráfagas del modelo del Mapa de Rulkov</i>	111

ÍNDICE DE TABLAS

<i>Tabla 4.1. Los modelos RS, IB y CH corresponden a neuronas corticales excitatorias:.....</i>	<i>46</i>
<i>Tabla 4.2. Los modelos FS y LTS a interneuronas corticales inhibitorias</i>	<i>46</i>
<i>Tabla 4.3. Neuronas tálamo-corticales:</i>	<i>47</i>
<i>Tabla 4.4. Representación del potencial de membrana de la simulación de los modelos neuronales: Hindmarsh-Rose, Eugene Izhikevich y el Mapa de Rulkov; utilizando para su resolución el método de Euler, con un paso de integración fijo de 0,001. Estas representaciones permiten comprobar que dichos modelos presentan un comportamiento de ráfaga en su potencial de acción.</i>	<i>53</i>
<i>Tabla 4.5. Representación del potencial de membrana generado por el modelo Hindmarsh-Rose y el tiempo de ejecución total en segundos. La representación superior muestra los resultados obtenidos de resolver el modelo de Hindmarsh-Rose utilizando el método de Euler con un paso de integración fijo de 0,001. En la representación inferior se ha resuelto el modelo utilizando el método de Runge-Kutta de cuarto orden con paso de integración fijo de 0,001. Ambas representaciones permiten comparar la diferencia entre los tiempos de simulación según el modelo matemático utilizado en la implementación del modelo Hindmarsh-Rose con paso de integración fijo 0,001.....</i>	<i>54</i>
<i>Tabla 4.6. Se representa el potencial de membrana obtenido de la simulación del modelo de Hindmarsh- Rose, utilizando el método de Euler, para diferentes valores de paso de integración fijo y el tiempo de ejecución total en segundos, con el objetivo de elegir el valor de paso que mejor aproxime la simulación del modelo a un tiempo cercano pero inferior de un segundo. Se representan un total de 280.000 puntos, los cuales para el modelo de Hindmarsh-Rose supone la simulación de una ráfaga. La representación (A.) muestra los resultados del uso de un paso de integración fijo de 0,01. (B.) representa los resultados obtenidos con un paso fijo de 0,001 y (C.) muestra los resultados para el paso fijo de 0,0001. Con objeto de realizar la comparación entre gráficas, se realiza la representación temporal durante 2 segundos.....</i>	<i>56</i>
<i>Tabla 4.7. Se representa el potencial de membrana obtenido de la simulación del modelo de Izhikevich, utilizando el método de Euler, para diferentes valores de paso de integración fijo y el tiempo de ejecución total en segundos, con el objetivo de elegir el valor de paso que mejor aproxime la simulación del modelo a un tiempo cercano pero inferior de un segundo. Se representan un total de 280 000 puntos, los cuales para el modelo de Izhikevich, a diferencia del modelo anterior, supone la simulación de cuatro ráfagas. La representación (A.) muestra los resultados del uso de un paso de integración fijo de 0,01. (B.) representa los resultados obtenidos con un paso fijo de 0,001 y (C.) muestra los resultados para el paso fijo de 0,0001. Con objeto de realizar la comparación entre gráficas, se realiza la representación temporal durante 2 segundos</i>	<i>57</i>
<i>Tabla 4.8. Valores de paso de integración fijo seleccionados para los modelos de Hindmarsh-Rose e Izhikevich utilizados en su resolución utilizando el método de Euler</i>	<i>59</i>
<i>Tabla 4.9. Se calcula el tiempo de ejecución esperado para cada punto</i>	<i>59</i>

<i>Tabla 4.10. Resultados obtenidos de la simulación, utilizando el método de Euler, de los modelos de Hindmarsh-Rose e Izhikevich con el método de parada 1, generándose 10000 puntos en un tiempo de un segundo en tiempo real. Las columnas 2 y 3 de la tabla permiten comparar la diferencia de tiempos antes y después de aplicar las paradas del sistema. La imagen superior contiene la representación del modelo Hindmarsh-Rose con paso de integración fijo de 0,001 y muestreo cada 28 puntos de los valores generados. La imagen inferior representa el modelo Izhikevich con paso de integración fijo de 0,0001 y muestreo cada 100 puntos de los valores generados.....</i>	<i>60</i>
<i>Tabla 4.11. Resultados obtenidos de la simulación con el método de parada 2, utilizando el método de Euler para la resolución de los modelos de Hindmarsh-Rose e Izhikevich, generándose 10000 puntos en un tiempo de un segundo en tiempo real. Las columnas 2 y 3 de la tabla permiten comparar la diferencia de tiempos antes y después de aplicar las paradas del sistema. La imagen superior contiene la representación del modelo Hindmarsh-Rose con paso de integración fijo de 0,001 y muestreo cada 28 puntos de los valores generados. La imagen inferior representa el modelo Izhikevich con paso de integración fijo de 0,0001 y muestreo cada 100 puntos de los valores generados.....</i>	<i>61</i>
<i>Tabla 4.12. Representaciones del Mapa de Rulkov. Para la simulación de 2000 puntos se obtiene una actividad de 6 ráfagas y para 400 puntos se representa una sola ráfaga.....</i>	<i>62</i>
<i>Tabla 4.13. Representación del modelo de Hindmarsh-Rose y el modelo de Izhikevich con un paso de integración de 0,001 para ambos modelos, y un límite de 280.000 puntos generados en el PC para el desarrollo inicial.....</i>	<i>64</i>
<i>Tabla 4.14. Representación del modelo de Hindmarsh-Rose y el modelo de Izhikevich con un paso de integración fijo de 0,001 para ambos modelos, y un límite de 280000 puntos generados en el PC del laboratorio del GNB.....</i>	<i>65</i>
<i>Tabla 4.15. Comparación de los resultados obtenidos utilizando el PC del laboratorio del GNB y el PC personal de desarrollo inicial. La columna de la izquierda contiene las simulaciones llevadas a cabo en el PC del laboratorio del GNB. Las imágenes de la columna de la derecha corresponden a las simulaciones iniciales en el PC personal.</i>	<i>66</i>
<i>Tabla 5.1. Circuito híbrido con el modelo Hindmarsh-Rose sin ajustes de amplitud y acoplamiento $g=0$; es decir, sin utilizar valor de corriente. La imagen azul corresponde al valor del potencial que genera la neurona electrónica, la imagen roja al potencial generado por la neurona software. Las imágenes se representan en un segundo.....</i>	<i>75</i>
<i>Tabla 5.2. Resultados obtenidos para el circuito formado por la neurona software con el modelo Hindmarsh-Rose y la neurona electrónica, variando el valor de acoplamiento.....</i>	<i>76</i>
<i>Tabla 5.3. Resultados obtenidos para el circuito formado por la neurona software con el modelo Izhikevich y la neurona electrónica, variando el valor de acoplamiento</i>	<i>79</i>
<i>Tabla 5.4. Representación de los potenciales de membrana de la implementación del Mapa de Rulkov (gráfica roja) y la neurona electrónica (gráfica azul), generados en el circuito establecido mediante la sinapsis eléctrica con valor de acoplamiento $g=0,5$. La imagen de la izquierda con $\alpha = 6$ representa el estado caótico del modelo. La de la derecha $\alpha = 4$, representa el estado regular.....</i>	<i>81</i>

<i>Tabla 5.5. Representaciones de los resultados obtenidos de la comunicación bidireccional entre la neurona software y la neurona electrónica para varios valores de conductancia con el objeto de comprobar que las implementaciones llevadas a cabo siguen su correcto funcionamiento sin el uso de las prestaciones del RTAI. Se ha utilizado el mismo valor de conductancia g en ambos sentidos de la comunicación y se mantiene el uso de las librerías de COMEDI necesarias para el uso de la tarjeta de adquisición de datos.</i>	<i>82</i>
<i>Tabla 5.8. Resultados de las simulaciones iniciales del circuito híbrido entre el modelo de Hindmarsh-Rose y la neurona utilizando una sinapsis eléctrica artificial, sin valores de acoplamiento ni offset. Se ha realizado dos pruebas con dos neuronas diferentes. En rojo se representa el potencial de membrana del modelo y en azul el de la neurona viva</i>	<i>86</i>
<i>Tabla 5.9. Resultados obtenidos utilizando un único electrodo conectado a la neurona viva. Este electrodo se encarga de aplicar la corriente a la neurona y de tomar los valores producidos por la misma. Se lleva a cabo la variación de los valores de acoplamiento y de ejes con el objeto de sincronizar el comportamiento de ambas neuronas. En rojo se representa el potencial de membrana del modelo y en azul el de la neurona viva</i>	<i>87</i>
<i>Tabla 5.10. En esta comunicación se utilizan dos electrodos conectados a la neurona viva. Un electrodo se encarga de aplicar corriente a la neurona y el otro de capturar las señales de membrana que produce la misma. En rojo se representa el potencial de membrana del modelo y en azul el de la neurona viva</i>	<i>88</i>
<i>Tabla 5.11. Se comprueba que el cambio de signo en uno de los valores de acoplamiento (transmisión de una corriente negativa) cambia el comportamiento de los potenciales de membrana de las neuronas que intervienen. En rojo se representa el potencial de membrana del modelo y en azul el de la neurona viva</i>	<i>89</i>
<i>Tabla 5.12. Resultados de la comunicación utilizando dos electrodos conectados a la neurona viva del segundo día de pruebas. Un electrodo se encarga de aplicar corriente a la neurona y el otro de capturar las señales de membrana que produce la misma. En rojo se representa el potencial de membrana del modelo y en azul el de la neurona viva</i>	<i>89</i>
<i>Tabla 5.13. Resultados obtenidos de la implementación del circuito híbrido entre el modelo de Izhikevich y la neurona viva conectada a dos electrodos. En rojo se representa el potencial de membrana del modelo y en azul el de la neurona viva. Las imágenes permiten observar el cambio de comportamiento de ambas neuronas al variar el valor de acoplamiento y el ajuste de ejes. La imagen inferior presenta mayor sincronismo que la superior</i>	<i>91</i>
<i>Tabla 5.14. Validación de la comunicación del circuito híbrido entre el modelo de Izhikevich y la neurona viva, variando la frecuencia de muestreo del modelo a valores diferentes de 10kHz. En rojo se representa el potencial de membrana del modelo y en azul el de la neurona viva</i>	<i>92</i>
<i>Tabla 5.15. Representación de los potenciales de membrana del circuito híbrido formado por el modelo de Izhikevich y la neurona viva, cambiando los parámetros característicos del modelo. En rojo se representa el potencial de membrana del modelo y en azul el de la neurona viva</i>	<i>93</i>
<i>Tabla 5.16. Representación en antifase de la comunicación del circuito híbrido. En rojo se representa el potencial de membrana del modelo y en azul el de la neurona viva</i>	<i>94</i>

<i>Tabla 9.1. Resultados obtenidos de la simulación del modelo Fitzhugh Nagumo para distintos valores de I_{ext}. Se observa que para valores de I_{ext} igual o mayores a 10 mA el modelo pierde su comportamiento de spiking (no produce potenciales de acción)</i>	<i>108</i>
<i>Tabla 9.2. Resultados obtenidos de la simulación del modelo de Hindmarsh-Rose. La imagen superior muestra el comportamiento regular del potencial de membrana del modelo cuando $e=3,0$. La imagen inferior, con $e=3,281$ representa el comportamiento no regular o caótico del modelo.....</i>	<i>110</i>
<i>Tabla 9.3. Representaciones de los resultados obtenidos de las simulaciones del modelo de Izhikevich para los casos RS, IB y CH correspondientes a neuronas corticales excitatorias</i>	<i>113</i>
<i>Tabla 9.4. Representaciones de los resultados obtenidos de las simulaciones del modelo de Izhikevich para los casos FS y LTS a interneuronas corticales inhibitorias</i>	<i>114</i>
<i>Tabla 9.5. Representaciones de los resultados obtenidos de las simulaciones del modelo de Izhikevich para Neuronas tálamo-corticales</i>	<i>115</i>

1.INTRODUCCIÓN

En este capítulo se explica la motivación por la cual se ha decidido desarrollar el tema del proyecto, los objetivos a alcanzar y la estructura usada en la memoria. El trabajo se desarrolla en el contexto del estudio de modelos de neurociencia computacional susceptibles de ser implementados online o en tiempo real para la construcción de circuitos híbridos, los cuales tienen una gran utilidad en el estudio de la dinámica de neuronas y redes biológicas.

En los circuitos híbridos se implementa una interacción bidireccional entre neuronas vivas y neuronas artificiales, teniendo en cuenta las restricciones temporales impuestas por la neurona viva, las particularidades del registro y escalado de la conexión; permitiendo observar y caracterizar la dinámica de las neuronas y de las redes en las que están integradas, exponiendo de esta forma aspectos de sus comportamientos que no son observables en experimentos con técnicas de estimulación tradicionales. Estos circuitos también pueden contribuir a un mejor control de la actividad irregular o patológica observada de sistemas neuronales.

1.1.MOTIVACIÓN

Este proyecto Fin de Carrera tiene como motivación el estudio de modelos neuronales que permiten simular el comportamiento de una neurona en tiempo real, e implementar circuitos híbridos con neuronas vivas en interacción bidireccional. En particular, se llevará a cabo la simulación de modelos que presentan comportamiento en ráfagas para su validación en circuitos generadores centrales de patrones compuestos por motoneuronas que producen un ritmo robusto para el control motor (Selverston et al., 2000).

A partir de las contribuciones llevadas a cabo por Santiago Ramón y Cajal (Ramón y Cajal, 1909), con su detallada descripción de la estructura de las neuronas, su independencia del tejido nervioso y el flujo de información que entre ellas se producía, utilizando las técnicas de análisis morfológico y fisiológico hoy se describe a la neurona como una unidad fundamental del cerebro. Las neuronas se caracterizan principalmente por su capacidad para generar actividad eléctrica de una forma robusta y flexible, por su procesamiento de información no lineal dependiente de la historia previa de estímulos, por su alta plasticidad y gran adaptabilidad y su tolerancia a fallos (Rabinovich et al., 2006). Los avances en tecnología han permitido llevar la investigación en estas células un paso más allá. Los modelos computacionales sobre el comportamiento de las neuronas permiten estudiar distintas variables involucradas en la generación de la actividad eléctrica, algo difícil de realizar sobre una neurona viva, en la cual sólo se accede típicamente a su potencial de membrana (Chamorro et al., 2012). Estos modelos intentan imitar el comportamiento de una neurona utilizando sistemas de ecuaciones diferenciales acopladas o mapas iterados.

Desde este punto parte la búsqueda de modelos sencillos para diseñar neuronas electrónicas, que a su vez sean capaces de reproducir el comportamiento de una neurona biológica real (Izhikevich, 2003), en particular las oscilaciones de voltaje de la membrana. En la implementación de los circuitos híbridos se han de modelar también las conexiones entre las neuronas vivas y artificiales. Estas conexiones pueden ser de tres tipos principales: conexión eléctrica óhmica (gap junction), conexiones químicas excitadoras y conexiones químicas inhibitoras (Rabinovich et al., 2006).

En este trabajo se utilizarán los circuitos generadores centrales de patrones (CPGs) para validar varios modelos neuronales implementados en software en configuraciones híbridas en las que las neuronas artificiales están conectadas bidireccionalmente a las neuronas de estos circuitos con sinapsis eléctricas. Los CPGs se han utilizado previamente para la implementación de circuitos híbridos con neuronas electrónicas artificiales implementadas en circuitos analógicos. La implementación de las configuraciones híbridas en CPGs permite estudiar la generación, coordinación y negociación de ritmos motores, y en particular la caracterización del balance entre la robustez y flexibilidad dinámica que presentan (Selverston et al., 2000). Además, el estudio de las características de los CPGs del sistema nervioso de seres invertebrados ha permitido utilizar estos modelos en implementaciones de sistemas de osciladores neuronales acoplados para el control de la locomoción de robots modulares (Herrero-Carrón et al., 2011; Ijspeert, 2008).

La comunicación bidireccional con una neurona viva requiere gestionar la adquisición del potencial de membrana de estas células, su conversión en una señal digital que puede ser introducida como estímulo a un modelo de neurona artificial y la correspondiente generación de una respuesta online o en tiempo real que pueda ser enviada a la neurona viva con las restricciones temporales necesarias para simular esta interacción. La respuesta proviene de la integración temporal del modelo en un tiempo adecuado, adaptándose a la escala temporal y a las restricciones impuestas por la neurona viva, produciendo una señal a la velocidad de interacción de 10KHz, siendo el valor elegido para el desarrollo del proyecto. Los valores seleccionados para las simulaciones pueden variar dependiendo del sistema en el que se lleven a cabo las pruebas.

1.2.OBJETIVOS

Este proyecto tiene por objetivo principal reunir en un documento el resultado de la implementación y validación en software de modelos neuronales que presentan un comportamiento en ráfagas, susceptibles de ser utilizados en circuitos híbridos construidos en CGPS, llevándose a cabo la validación del funcionamiento del modelo final elegido en tiempo real o tiempo online.

Una vez que los modelos elegidos sean validados en cuanto a su escalado temporal e integración en tiempo real, se podrá llevar a cabo distintas pruebas de conexión creando una comunicación bidireccional a partir de una sinapsis eléctrica generada artificialmente. Los circuitos a ser implementados y validados son:

- Conexión entre una neurona software y una neurona electrónica, como primera instancia de validación de comunicación online y en tiempo real.
- Conexión entre una neurona software y una neurona viva de un CPG, creando un circuito híbrido.

La adquisición de las señales biológicas, además de la construcción de los ciclos cerrados estímulo-respuesta necesarios para la implementación de circuitos híbridos están sujetos a una precisión temporal. Para llevar a cabo la comunicación en tiempo real será necesario la utilización de una Tarjeta de Adquisición de Datos (DAQ), en este proyecto se usará la tarjeta del laboratorio del Grupo de Neurocomputación Biológica (GNB) modelo National Instruments NI PCI-6521.

También es necesario recurrir a las librerías COMEDI que permiten el uso de funciones para la realización de los registros y el envío de datos a la DAQ.

Para la validación de la interacción se realizarán las pruebas en un PC con un sistema operativo de tiempo real bajo RTAI (<https://www.rtai.org>).

El equipamiento hardware y software utilizados para el desarrollo del proyecto será explicado con mayor detalle más adelante (*véase capítulo 3*).

1.3. ORGANIZACIÓN DE LA MEMORIA

La memoria consta de los siguientes capítulos:

- **Estado del arte:** Se lleva a cabo un resumen teórico de los principales temas en los que se basa y se desarrolla el proyecto, tales como los conceptos básicos sobre el sistema nervioso, centrandó la atención en la descripción, características y comunicación (sinapsis) de las neuronas. Se repasan los modelos matemáticos de neuronas más representativos para el proyecto, además de una explicación básica de los ciclos cerrados y los circuitos híbridos.
- **Equipamiento:** En este capítulo se hace un repaso de las principales herramientas, hardware y software, utilizadas para llevar a cabo el desarrollo y las distintas pruebas a lo largo del proyecto.
- **Diseño y desarrollo:** Se realiza una explicación más detallada de los modelos matemáticos de neuronas, con el objetivo de elegir aquellos modelos que presenten el comportamiento deseado para realizar las pruebas. Se detallan las ecuaciones, parámetros y características de los modelos, se realizan distintas simulaciones y se comparan los diferentes comportamientos y resultados obtenidos.
- **Pruebas y resultados:** A partir de los modelos matemáticos de neuronas elegidos e implementados en software se lleva a cabo las pruebas conectándolos a una neurona electrónica, como primera validación del cumplimiento de las restricciones temporales necesarias para la comunicación bidireccional. Obteniéndose resultados satisfactorios en el acoplamiento de ambas neuronas, se pasa a la conexión de la neurona software con la neurona viva de un CPG (circuitos híbridos), sincronizando la actividad y por lo tanto consiguiendo un acoplamiento efectivo entre ambas neuronas. Este capítulo alberga los resultados obtenidos de las pruebas entre las distintas conexiones y modelos.
- **Conclusiones y trabajos futuros:** Este capítulo contiene las conclusiones generales de todo el proyecto habiéndose cumplido los objetivos marcados en su inicio. Además, contiene los posibles trabajos a desarrollar en un futuro con el objeto de ampliar el desarrollo de este trabajo.

2. ESTADO DEL ARTE

En este capítulo se lleva a cabo un resumen sobre las bases en las que se fundamenta este proyecto, las cuales son principalmente las características esenciales de la neurona biológica, los modelos neuronales susceptibles de implementarse en software y los circuitos híbridos, con objeto de aportar la información necesaria al lector para la mejor comprensión de este estudio.

Los experimentos llevados a cabo utilizan un control de ciclo cerrado con métodos como el de la fijación de voltaje o de corriente, ya que el uso de un circuito abierto en sistemas con características biológicas (sistemas no lineales y estocásticos) presentan unos resultados de dinámica inestables que se manifiestan en salidas complejas del sistema (Chamorro et al., 2012; Wallach, 2013).

2.1. EL SISTEMA NERVIOSO

2.1.1. INTRODUCCIÓN

Antes de que Santiago Ramón y Cajal propusiera a la neurona como la unidad funcional del sistema nervioso (Ramón y Cajal, 1909), se pensaba que éste estaba constituido por un retículo continuo, teoría reticular. Esta teoría sostenía que la función del soma de las neuronas era principalmente proporcionar alimento al sistema y fue defendida incluso después de que la teoría de la célula saliera a la luz a mediados del siglo XIX (Kandel et al., 2000; Wilson and Keil, 2001).

Gracias a la técnica de tinte desarrollada por Camillo Golgi que permitía un análisis celular muy preciso, incluso en tejidos densos como el cerebral, a finales del siglo XIX Santiago Ramón y Cajal logró demostrar que el tejido nervioso está formado por células. Su teoría consideró al sistema nervioso como un conjunto de unidades independientes, entidades discretas que, intercomunicándose, establecían una especie de red mediante conexiones especializadas o espacios (Grant, 2007; López-Muñoz et al., 2006)

Actualmente se conoce que el sistema nervioso está formado por dos tipos principales de células:

- **Las neuronas:** Unidades estructurales y funcionales.
- **Las células de glía:** Actúan como soporte de las neuronas. Además, intervienen en el procesamiento cerebral de la información (Kandel et al., 2000).

En este proyecto, la neurona será la unidad de estudio en la que nos centraremos.

2.1.2. LA NEURONA

Las neuronas son células especializadas en la recepción de estímulos y en la conducción del impulso nervioso. Las neuronas conforman e interconectan los tres componentes del sistema nervioso: el sistema sensorial, el sistema nervioso central y el sistema motor. El número de neuronas que componen el cerebro varía según la especie observada (Williams and Herrup, 1988).

Morfológicamente, una neurona típica está compuesta por el soma, el axón y las dendritas (Paniagua et al., 2002), ver figura 2.1.

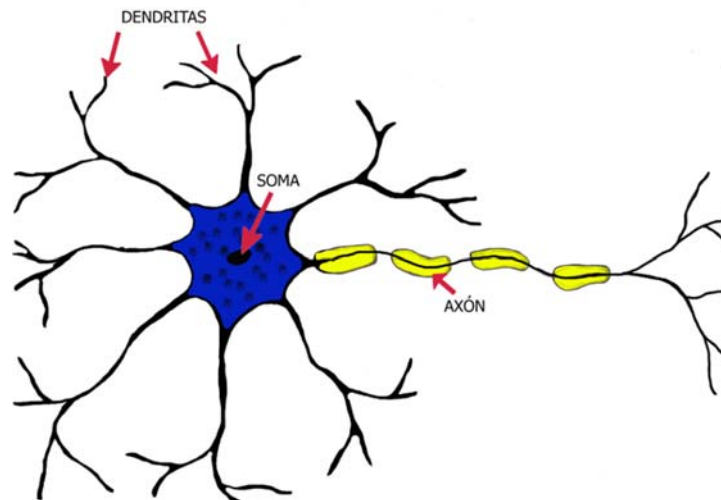


Figura 2.1. Diagrama que presenta la estructura básica de una neurona, indicando las tres partes más relevantes de la misma.

- **El soma:** es el cuerpo celular de la neurona que contiene el núcleo y los orgánulos principales.
- **Dendritas:** son ramificaciones que proceden del soma neuronal. Forman los denominados árboles dendríticos que consisten en proyecciones citoplasmáticas envueltas por una membrana plasmática. Son los principales receptores de señales nerviosas procedentes de los receptores sensoriales o de otras neuronas.
- **El axón:** es una larga prolongación del soma neuronal que actúa como principal conductor de las señales desde el cuerpo celular hasta los botones terminales.
 - **Los botones terminales:** son pequeños engrosamientos que se encuentran en ramificaciones finas al final de los axones. Secretan unas sustancias químicas llamadas neurotransmisores, los cuales excitan o inhiben a la neurona que los recibe y pueden generar o no un potencial de acción en su axón (Chen et al., 1997; Herreras, 1990; Paniagua et al., 2002; Stuart et al., 1997).

2.1.3.COMUNICACIÓN NEURONAL

Cada célula nerviosa está conectada a otras células nerviosas a través de sus axones y dendritas. Se comunican las unas con las otras mediante la generación de señales eléctricas denominadas impulsos nerviosos o potenciales de acción, los cuales producen la liberación de los neurotransmisores. En muchas ocasiones esta comunicación ocurre entre células que se encuentran muy alejadas entre sí. *La conexión entre una neurona y otra se denomina sinapsis.*

2.1.3.1. MEMBRANA CELULAR

La célula posee una membrana citoplasmática que la separa del medio extracelular. Esta membrana presenta una permeabilidad selectiva a determinadas moléculas, permitiéndolas salir y entrar de la célula a través de los canales iónicos, los cuales pueden ser:

- **Canales pasivos:** Siempre están abiertos.
- **Canales activos:** Pueden abrirse o cerrarse en respuesta a señales eléctricas, mecánicas o químicas.

La carga eléctrica a cada lado de la membrana celular es diferente, lo que produce una diferencia de potencial entre el interior y el exterior (potencial de membrana). Es la evolución temporal de este potencial de membrana lo que caracteriza el comportamiento de una neurona.

Se dice que una neurona se encuentra en reposo cuando no recibe ningún estímulo. El interior de la célula respecto al exterior presenta una carga negativa. Aunque su valor es distinto entre unas neuronas y otras, se caracteriza por el valor típico de -65mV (mili voltios), tomándose 0mV como potencial de referencia en el exterior. En este caso, la mayoría de los canales activos se encontrarán cerrados (Hille, 1978; Kandel et al., 2000; Paniagua et al., 2002).

2.1.3.2. POTENCIAL DE ACCIÓN

Se conoce como potencial de membrana al generado al producirse un movimiento de cargas eléctricas a través de los canales de la membrana celular.

Este potencial puede variar en determinadas circunstancias, produciendo una despolarización si aumenta, o una hiperpolarización en el caso contrario. Para rangos cercanos al potencial de reposo se habla de actividad subumbral, que se refiere a pequeñas despolarizaciones e hiperpolarizaciones, ver figura 2.2.

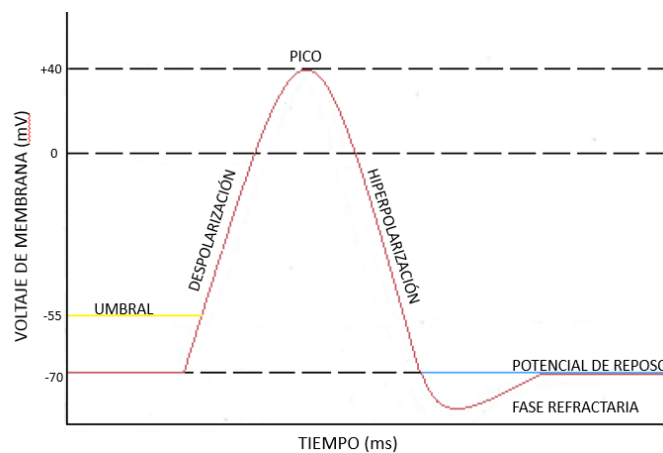


Figura 2.2. Cuando el potencial de membrana aumenta su valor superando el umbral elegido, comienza una fase de despolarización hasta alcanzar un valor pico a partir del cual disminuye su valor, entrando en la fase de hiperpolarización hasta llegar a la zona de reposo.

Cuando el potencial de membrana se incrementa de manera muy rápida en un intervalo de tiempo corto, y supera un determinado umbral de despolarización, se produce un potencial de acción o spike. La alternación de potenciales de acción y estados de relajación en una neurona se conoce como actividad spiking, ver figura 2.3.

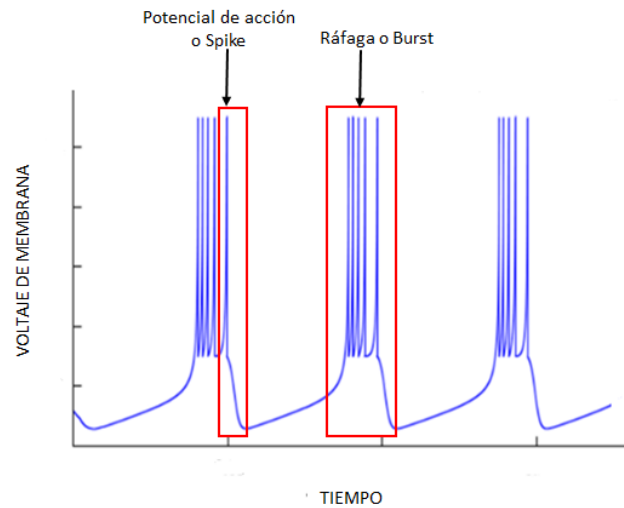


Figura 2.3. El cambio rápido desde despolarización a hiperpolarización produce un potencial de acción o spike. El conjunto de varios potenciales de acción en un corto intervalo de tiempo es denominada ráfaga o burst.

A su vez, la agrupación de dos o más potenciales de acción en un corto intervalo de tiempo, se conoce como ráfagas o burst. Es muy común que no se produzcan de manera aislada, por lo que se alternan periodos de gran actividad con periodos de actividad subumbral, conocido este comportamiento como actividad bursting (Kandel et al., 2000; Paniagua et al., 2002).

2.1.3.3. SINAPSIS

Se define sinapsis como una unión (funcional) intercelular especializada entre neuronas (Hormuzdi et al., 2004).

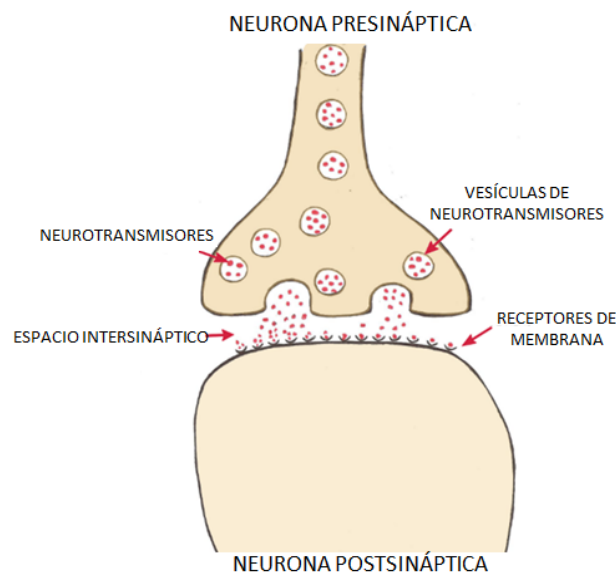


Figura 2.4. Componentes de una sinapsis química

Se pueden distinguir dos tipos principales de sinapsis, la sinapsis eléctrica y la sinapsis química.

- **Sinapsis eléctrica:** la proximidad de las neuronas en comunicación permite que una corriente fluya directamente entre sus membranas, por lo que no se produce la secreción de un neurotransmisor, sino el paso de iones entre las células a través de uniones gap.

Este intercambio de comunicación se puede realizar de manera simétrica y simultáneamente en ambos sentidos con la misma intensidad (**transmisión bidireccional**). Es una comunicación muy rápida y sin apenas retardos en comparación con la sinapsis química (Kandel et al., 2000; Paniagua et al., 2002). En algunos tipos de sinapsis eléctrica la corriente fluye en una única dirección (**sinapsis rectificadora**).

Que la corriente fluya en una única dirección o en ambas depende de la resistencia al flujo de corriente que oponen las uniones gap. La corriente que cruza las uniones gap es proporcional a la diferencia de potencial entre las dos neuronas.

La expresión de la corriente de la sinapsis eléctrica desde la neurona software (neurona 1) hacia la neurona electrónica o la neurona viva (neurona 2) en los circuitos híbridos que se han desarrollado en este proyecto es de la forma: $(g_{21}(x_1(t) - x_2(t)))$. Existiendo otra corriente en el sentido contrario, desde la neurona 2 hacia la neurona 1 con una expresión similar a la anterior: $(g_{12}(x_2(t) - x_1(t)))$.

- **Sinapsis química:** Es una transmisión entre neuronas producida por la secreción de neurotransmisores que se encuentran almacenados en los botones terminales presinápticos en unos pequeños sacos de membrana que reciben el nombre de vesículas sináptica. Estos neurotransmisores sólo son segregados por la neurona presináptica en el momento que produce un potencial de acción. La sinapsis química puede ser **gradual**, lo que supone que se liberan neurotransmisores incluso antes de que se produzca el potencial de acción.

La sinapsis química es un proceso asimétrico tanto en estructura como en funcionamiento. En esta comunicación a diferencia de la sinapsis eléctrica, las neuronas no se encuentran en contacto físico. Los neurotransmisores se unen a los receptores transmembrana de la célula postsináptica. Gracias a la unión entre los receptores transmembrana de la célula postsináptica con los neurotransmisores, se abren los canales iónicos permitiendo el flujo de los iones hacia o desde el interior, lo que cambia el potencial de membrana postsináptica. Existen neurotransmisores excitadores e inhibidores. Las corrientes que provocan los primeros despolarizan la neurona; las que producen los inhibidores la hiperpolarizan. Que se produzca un caso u otro depende del tipo de iones canalizados y de los receptores y neurotransmisores que finalmente intervienen. Se produce un retardo en la transmisión del orden de 1 a 5 milisegundos desde que se produce el potencial de acción en la neurona presináptica (Kandel et al., 2000; López-Muñoz et al., 2006; Paniagua et al., 2002).

2.1.4. MODELO ELÉCTRICO DE UNA MEMBRANA NEURONAL

El medio intracelular de una neurona se encuentra separado del medio extracelular por una membrana plasmática de 6 a 8 nm de espesor. En el interior de estas células se encuentra una mayor concentración de iones de potasio K^+ , en el exterior los iones de sodio Na^+ y cloro Cl^- están presentes en mayor cantidad. El flujo de estos iones a través de la membrana plasmática genera corrientes eléctricas. A causa de la impermeabilidad que presentan las bicapas a estos iones, su recorrido a través de la membrana se lleva a cabo exclusivamente por canales iónicos específicos, teniendo que vencer los potenciales de Nernst asociados e independientes para cada uno de ellos.

Las observaciones experimentales, principalmente del axón gigante de calamar, llevaron a Cole y Curtis (Cole and Curtis, 1939) a proponer un circuito eléctrico sencillo para representar el comportamiento de las membranas excitables, denominado como circuito equivalente, ver figuras 2.5 y 2.6.

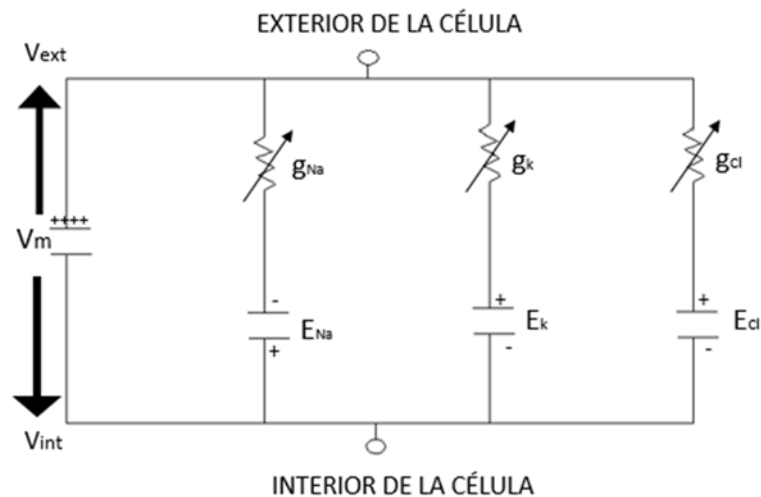


Figura 2.5. Modelo eléctrico del comportamiento pasivo de una membrana neuronal con canales de Na^+ , K^+ y Cl^-

Este circuito equivalente pretende representar las propiedades pasivas más destacadas de la membrana neuronal:

- **Resistencias variables:** Representan a los canales iónicos.
- **Fuentes de Tensión:** Representan los gradientes de concentración de las especies iónicas relevantes.
- **Condensadores:** Representan la capacidad de almacenamiento de carga eléctrica por parte de la membrana.

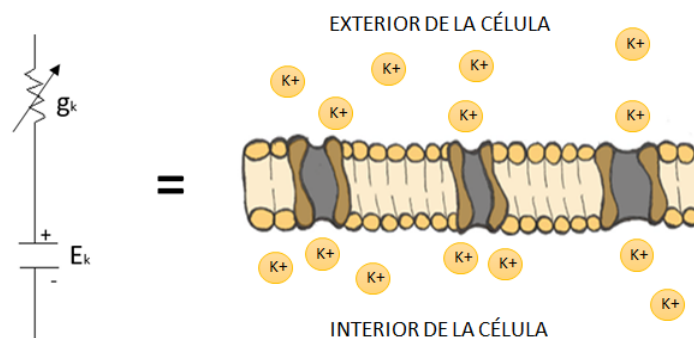


Figura 2.6. Representación del canal de potasio de una neurona

La resistencia eléctrica que ofrece la unidad de área de la membrana representa la capacidad de permitir o no el paso de iones que la atraviesen. Esta resistencia depende de la clase y del número de canales iónicos abiertos en la membrana. Su conductancia representa la capacidad de la membrana para oponerse al flujo de una corriente iónica y depende de la concentración iónica a uno y otro lado de la misma, además de la dependencia de las propiedades de la membrana. Por lo tanto, estos canales iónicos se comportan como una resistencia variable que representa una conductancia específica.

2.2. MODELOS NEURONALES

Puede tomarse al movimiento de iones de la sinapsis como un desplazamiento de carga, así la corriente iónica y la corriente electrónica pueden ser estudiadas bajo las mismas leyes de la física (Ley de Ohm, Kirchoff, etc), permitiendo desarrollar modelos matemáticos que posibilitan la descripción de la dinámica del voltaje a través de la membrana. En el contexto del proyecto, se han escogido modelos que presentan un comportamiento en ráfagas, lo que permita su simulación online o tiempo real con el fin de la implementación de circuitos híbridos en CPGs a través de una comunicación bidireccional.

Los primeros trabajos de modelos neuronales se deben a **McCulloch** (Neurofisiólogo) y **Walter Pitts** (matemático) quienes, en 1943, propusieron una teoría del procesamiento de información neuronal (McCulloch and Pitts, 1943). En esta misma época surgieron los primeros algoritmos de aprendizaje (Hebb, 1949); además del desarrollo de las técnicas de fijación o control de voltaje a partir de las demostraciones llevadas a cabo por **Kenneth Cole**.

Kenneth S. Cole fue un físico que estudió las propiedades de las membranas de una gran variedad de células, uno de los primeros en trabajar con el axón gigante del calamar. Cole junto a Curtis observaron que la conductancia de las membranas celulares aumentaba durante la ocurrencia de un potencial de acción de manera drástica (Cole and Curtis, 1939; Cole, 1968). Colocaron un axón entre dos electrodos que formaban parte de un circuito puente y balancearon el circuito hasta obtener una mínima señal. A continuación registraron tanto las corrientes como el cambio en impedancia de la membrana durante el paso del potencial de acción, al estimular uno de los nervios, encontraron que aunque la resistencia disminuía no se producían cambios en la capacitancia.

Todos estos trabajos contribuyeron para que en el año 1952 los neurofisiólogos **A. Hodgkin** y **A. Huxley** realizaran una serie de experimentos con el axón gigante de calamar, midiendo los potenciales extracelulares originados por el paso de un potencial de acción; utilizando la diferencia de potencial entre una región despolarizada y otra intacta como potencial de reposo. Diseñaron así un modelo matemático (véase apartado 4.1.1), base de los modelos realistas actuales, para explicar la dinámica del potencial de membrana de una neurona ante una corriente de iones inyectada (Hodgkin and Huxley, 1952).

El modelo presentado por **A. Hodgkin** y **A. Huxley** contiene una ecuación diferencial para cada una de las variables de estado (el potencial de membrana V , y las variables de conductancia m , n y h). Posteriormente se propusieron conjuntos de ecuaciones con un número menor de variables de estado, por ejemplo de tensión y otra variable de estado, que pueden recrear una serie de propiedades del modelo Hodgkin-Huxley. Algunos de estos modelos simplificados son:

- El modelo Fitzhugh Nagumo: En 1961 **Richard FitzHugh** propuso una simplificación de 2 dimensiones al modelo de Hodgkin-Huxley, llamándolo modelo de Bonhoeffer-van der Pol (Fitzhugh, 1961). Dicho modelo fue implementado mediante un circuito electrónico diseñado por el ingeniero japonés Jin-Ichi Nagumo (véase apartado 4.1.3), pasando a llamarse al conjunto de ecuaciones como el modelo FitzHugh Nagumo (Nagumo et al., 1962).
- El modelo de Morris-Lecar: **Cathy Morris** y **Harold Lecar**, en el año 1981, desarrollaron un modelo matemático de neurona biológica (Morris and Lecar, 1981). Se considera también una reducción del modelo cuatridimensional de Hodgkin-Huxley. Este sistema de ecuaciones describe de forma simplificada la relación entre la activación de los canales iónicos y el potencial de membrana: el potencial depende de la actividad de los canales iónicos, y la actividad de los canales de iones depende de la tensión. El modelo fue desarrollado originalmente para describir la fibra muscular gigante del percebe, pero el modelo se ha aplicado a otros sistemas, tales como neuronas del ganglio estomatogástrico de la langosta (Skinner et al., 1993) y las neuronas sensoriales espinales de mamíferos (Prescott et al., 2008).
- El modelo Kepler: este modelo se presentó como un esquema para la reducción sistemática del número de ecuaciones diferenciales de los modelos neuronales realistas. Se basa en la reducción de la conductancia de los modelos neuronales. Como ejemplo los autores presentan la reducción del modelo Hodgkin-Huxley y el modelo A-current de Connor (Connor et al., 1977; Kepler et al., 1992).

Hindmarsh y **Rose** modificaron el modelo Fitzhugh Nagumo, reemplazando la función lineal por una cuadrática dotando al modelo de la capacidad de describir de forma más realista los potenciales de acción en cortos intervalos de tiempo; publicando un primer modelo en 1982 (véase apartado 4.1.4, donde se describen en más detalle los modelos utilizados en este trabajo).

En 1984 presentaron un modelo mejorado (Hindmarsh and Rose, 1984), en el que se incorporaba una tercera ecuación, dando lugar a un sistema de ecuaciones diferenciales no lineales tridimensional, capaz de simular casi al completo el comportamiento neuronal generado por el modelo Hodgkin-Huxley, pero con una estructura más sencilla.

Nikolai F. Rulkov propuso el mapa iterado de dos dimensiones para modelar el comportamiento neuronal (Rulkov, 2002). Este modelo utiliza una señal de dinámica rápida y otra de dinámica lenta para imitar de manera aproximada el comportamiento eléctrico neuronal (véase apartado 4.1.5).

En el caso de los modelos de neuronas cuadráticas (Integrate & fire) que presentan una gama amplia de comportamientos realistas sin describir el detalle de la producción del potencial de acción, este es el caso del modelo propuesto Izhikevich (Izhikevich, 2003; Izhikevich y Edelman, 2008).

En 2003 **Eugene Izhikevich** propuso un modelo de neurona pulsante simplificado (Izhikevich, 2003). El modelo basa su comportamiento en un sistema de 2 ecuaciones diferenciales ordinarias y en dos requisitos: ser un modelo simple y a la vez ser capaz de reproducir el comportamiento neuronal real. Así, Izhikevich presentó un modelo para la simulación a gran escala de un cerebro sin el coste computacional que presentaba el modelo Hodgkin-Huxley (véase apartado 4.1.6).

2.3. ESTIMULACIÓN NEURONAL EN CICLO CERRADO

Puesto que los sistemas nerviosos son no lineales y adaptativos, y procesan información en regímenes transitorios, tiene sentido estudiarlos utilizando estimulación en ciclo-cerrado (Chamorro et al., 2012). Este tipo de estimulación contribuye a caracterizar su dinámica e implementar el control de su actividad normal o patológica.

En estos ciclos cerrados o de estimulación dependiente de la actividad el método de Dynamic Clamp es el método más representativo. Éste consiste en la implementación de conductancias iónicas o sinapsis artificiales en neuronas biológicas mediante la inyección de corriente en función del potencial registrado de forma instantánea (Destexhe and Bal, 2009; Sharp et al., 1993).

De forma parecida al concepto clásico de dynamic clamp, se han desarrollado nuevos protocolos para distintos contextos de investigación del sistema nervioso, manteniéndose el principio de ciclo cerrado de estimulación (van Boxtel and Gruzelier, 2014; Roth et al., 2014). Éstos son utilizados para revelar ritmos o dinámica de un amplio rango de procesos, y para lograr un control de estados patológicos o naturales, permitiendo el estudio de aspectos de la dinámica de los sistemas biológicos que aparecen ocultos en los protocolos tradicionales de estímulo-respuesta en ciclo abierto. Además, se pueden utilizar para favorecer el aprendizaje y también para automatizar experimentos. Gracias a la precisión temporal se tiene una representación y control más efectivos del nivel de intensidad, duración o periodicidad del estímulo en función del objetivo dado al ciclo cerrado (Chamorro et al., 2012; Fernandez-Vargas et al., 2013; Potter et al., 2014).

Dynamic Clamp (Fijación del voltaje): Durante más de 60 años, el control en tiempo real ha sido importante para el estudio de las células excitables. En los métodos de fijación de voltaje el potencial de membrana es controlado activamente, permitiendo la parametrización de modelos que describen la relación tensión-corriente de las membranas celulares. Estos métodos dinámicos de fijación permiten la adición de nuevos mecanismos de membrana "virtuales" a las células vivas.

La técnica de fijación de voltaje (Voltage Clamp) fue desarrollada por Cole y Marmont en los años 1940 en la Universidad de Chicago (Cole, 1968). La idea básica es fijar el potencial de membrana a un valor constante o con un perfil variable en el tiempo. Al igual que con la fijación de corriente, un electrodo se utiliza para inyectar corriente en la célula. Al mismo tiempo, un electrodo de tensión registra el potencial de membrana. El aparato ajusta la corriente inyectada continuamente para que sea justo lo suficiente para contrarrestar las desviaciones del potencial de membrana grabado desde el valor de tensión deseado. Esto asegura que la membrana permanece en el valor constante deseado o forma de onda de perfil variable en el tiempo (Graham, 2011).

2.4. CIRCUITOS HÍBRIDOS

Los conocimientos biológicos sobre la neurona, los avances en la tecnología y los estudios con el dynamic clamp o pinzamiento dinámico (Prinz et al., 2004; Sharp et al., 1993, 1996) han permitido implementar modelos de simulación en tiempo real (Real Time), que aplicados a los circuitos híbridos en los que neuronas biológicas interactúan con neuronas electrónicas, posibilitan llevar a cabo experimentos imposibles de realizar con las técnicas neurofisiológicas tradicionales. Estos experimentos proporcionan el realismo biológico necesario para el estudio del comportamiento neuronal (Manor and Nadim, 2001; Le Masson et al., 1999; Szücs et al., 2000; Yarom, 1991).

Entre los estudios pioneros en este campo se encuentran los llevados a cabo por Yarom, "Rhythmogenesis in a hybrid system - Interconnecting an olivary neuron to an analog network of coupled oscillators" (Yarom, 1991), quien utilizó la conexión entre una célula olivar y un simulador analógico compuesto por cuatro unidades oscilantes idénticas interconectadas, para estudiar los mecanismos por los cuales las oscilaciones en estas neuronas biológicas provocan oscilaciones sincronizadas en el potencial de membrana en un gran número de neuronas. También se puede observar los desarrollos de neuronas electrónicas analógicas con el fin de reproducir el patrón de descarga de las neuronas CPG de la langosta pilórica, "Interacting biological and electronic neurons generate realistic oscillatory rhythms" (Szücs et al., 2000). La conexión bidireccional entre la neurona electrónica y la neurona biológica creando un circuito híbrido permitió estudiar que las oscilaciones en ráfagas periódicas generadas en esta conexión dependían de la fuerza y el sentido de la señal del acoplamiento eléctrico. Se observó que el circuito híbrido permitía restaurar los ritmos característicos de las neuronas CPG regulares o irregulares, en los casos en los que estos ritmos habían sido alterados de forma aislada en cada neurona. Hay que destacar que los modelos en circuitos electrónicos garantizan las restricciones temporales de la integración del modelo.

Manor y Nadim utilizaron un circuito híbrido, "Frequency regulation demonstrated by coupling a model and a biological neuron" (Manor and Nadim, 2001), observando que si la sinapsis entre la neurona marcapaso biológica y el modelo neuronal en tiempo real disminuye se produce biestabilidad en el sistema, estando una de las salidas controlada por las propiedades intrínsecas de la neurona biológica, mientras la otra salida es controlada por la disminución de la sinapsis. En otros estudios llevados a cabo, como por ejemplo "Feedback inhibition controls spike transfer in hybrid thalamic circuits" (Le Masson et al., 2002), se han utilizado modelos de neuronas en tiempo real para investigar fenómenos como el papel de la retroalimentación en circuitos del tálamo, a través del cual la información sensorial llega a la corteza cerebral.

Para ello se diseñó una red híbrida *in vitro* que simulaba el circuito tálamo-cortical, observando que la combinación del control de ganancia de la inhibición de la retroalimentación y la modulación de la excitabilidad de la membrana permite que circuitos talámicos se coordinasen para la transmisión desde los órganos de los sentidos a la corteza.

En artículos como “Experimental and theoretical analysis of neuron-transistor hybrid electrical coupling: The relationships between the electro-anatomy of cultured *Aplysia* neurons and the recorded field potentials” (Cohen et al., 2006), se habla de la importancia de la comprensión de los mecanismos que generan los potenciales de campo (FPS) de las neuronas implementadas en chips semiconductores para poner en funcionamiento dispositivos de neuro-electrónica. El estudio se llevó a cabo sobre neuronas de la *Aplysia*, pequeño molusco, y se observó que el potencial de campo generado por estas neuronas es el resultado del flujo de corriente longitudinal entre los compartimentos neuronales eléctricamente distantes. Los resultados obtenidos demuestran que la forma y la amplitud del potencial de campo están relacionadas con la complejidad morfológica de una neurona dada, y sus propiedades biofísicas inherentes. Además, el estudio de circuitos híbridos ha permitido el desarrollo de software tal como el utilizado en los experimentos de “Single electrode dynamic clamp with StpC” (Nowotny et al., 2006; Samu et al., 2012). El uso del software de fijación dinámica StpC permite presentar nuevas características del generador de onda, observar sinapsis, además de permitir el desarrollo de un método de compensación de electrodo activo (AEC) a partir de su uso. El software StpC es el primer sistema implementado para ser usado por usuarios no expertos, y ha permitido superar algunas de las limitaciones que presentan los experimentos de pinzamiento dinámico, en los cuales se utilizan típicamente dos electrodos separados en la misma célula, uno para la grabación de potencial de membrana y otro para la inyección de corrientes. El requisito de dos electrodos independientes ha sido un factor limitante para el uso de fijación dinámica en aplicaciones en las que grabaciones duales de este tipo son difíciles o imposibles de lograr. A partir de varios ejemplos se ilustra que con software como StpC, el pinzamiento dinámico se ha desarrollado más allá de la mera introducción de las sinapsis artificiales o conductancias iónicas en las neuronas y ha pasado a ser una herramienta de investigación estándar de la electrofisiología moderna.

También existen experimentos utilizando una sinapsis artificial comunicando muestras vivas. El uso de una sinapsis artificial permitió la aplicación de acoplamientos eficaces variables, cambiando la magnitud y polaridad de la conductancia entre las neuronas, observándose que los procesos subcelulares lentos podrían ser responsables de los mecanismos implicados en la sincronización y la regularización de las actividades caóticas individuales. En “Synchronous Behavior of Two Coupled Biological Neurons” (Pinto et al., 2000), se lleva a cabo el acoplamiento eléctrico entre dos neuronas PD de la GPC pilórica del STG de la langosta, presentándose los resultados obtenidos de las simulaciones experimentales de los fenómenos de sincronización en un par de neuronas biológicas que interactúan a través de acoplamiento eléctrico. Estas neuronas de manera aislada muestran un comportamiento irregular, de hecho caótico, pero en conexión con otras neuronas pueden coordinarse de forma regularizada con el fin de producir ritmos regulares y a la vez flexibles. Se observó que este patrón de actividad puede ser alterado mediante la inyección de corriente DC en las neuronas.

En paralelo a su acoplamiento natural, se añade acoplamiento artificial mediante un dispositivo de pinzamiento dinámico de corriente (dynamic current clamp), que permite estudiar las transiciones entre sincronía y asincronía en oscilaciones lentas y picos rápidos; lo cual no era posible observarse utilizando acoplamientos naturales, pues sólo se producía sincronización en las oscilaciones lentas del potencial de membrana y no en los picos rápidos. En el trabajo expuesto en el artículo “Dynamics of two electrically coupled chaotic neurons: experimental observations and model analysis” (Varona et al., 2001), se desarrollaron modelos basados en la conductancia de la neurona del CPG pilórico de la langosta, con el objeto de entender el comportamiento caótico de las neuronas individuales. Para ello se llevó a cabo la sinapsis eléctrica artificial entre dos neuronas PD, comparando los resultados con los obtenidos en modelos biofísicos de estas neuronas. En “Feedback control of variability in the cycle period of a central pattern generator” (Hooper et al., 2015), en este trabajo, utilizando un oscilador y un elemento PIR se implementó una retroalimentación entre una neurona AB (Anterior Buser)/una neurona PD (Pyloric Dilator) y una neurona LP (Lateral Pyloric). La retroalimentación virtual genera la entrada a la PD en un tiempo marcado por la dinámica de respuesta de fase ajustable, que imita los intervalos de ráfaga generados por la neurona LP cuando está en la red pilórica. La neurona elegida como marcapasos recibe la retroalimentación virtual a través del pinzamiento dinámico. Este circuito permite medir la dependencia de variabilidad periódica de la red en la dinámica de la respuesta de fase de los elementos retroalimentados, observándose mínimas variaciones en los intervalos de respuesta constantes.

3. HARDWARE Y SOFTWARE UTILIZADO

En este proyecto se ha llevado a cabo la implementación en software de tiempo real de varios modelos de neuronas que presentan comportamientos en ráfagas. Ha sido necesario realizar las simulaciones teniendo en cuenta las restricciones de tiempo impuestas por la neurona viva y la velocidad de adquisición de la tarjeta DAQ, por lo que ha sido muy importante que los diseños de los modelos se pudieran ejecutar sin retraso o adelanto en distintos equipos y que siguiera cumpliendo con el comportamiento esperado. Como podrá observarse en siguientes capítulos, se han realizado varias pruebas de los modelos implementados sin y con la comunicación con otra neurona (neurona electrónica, neurona viva), así se ha podido comprobar que realmente los programas mantienen su correcto funcionamiento con unos márgenes de diferencia en tiempos que no afectan a los resultados.

3.1. CARACTERÍSTICAS DE LOS PC'S UTILIZADOS

3.1.1. PC PERSONAL PARA DESARROLLOS INICIALES

El diseño inicial y las primeras pruebas para observar el comportamiento de cada modelo de neurona elegido se realizó sobre un PC con las siguientes características:

Procesador	Intel® Core™ i3-4150 CPU @ 3.50GHz
Memoria instalada (RAM)	4,00 GB (3,81 GB utilizable)
Tipo de sistema	Sistema operativo de 64 bits, procesador x64
Sistema Operativo	sobre Linux Versión 3.13.0-44-generic

3.1.2. PC DEL LABORATORIO DEL GNB

Para probar el correcto funcionamiento de los modelos realizados en otro equipo, se utilizó un PC del laboratorio del Grupo de Neurocomputación Biológica (GNB) con las siguientes características. Los resultados obtenidos se pueden observar en siguientes capítulos (*véase capítulos 4 y 5*).

Procesador	Intel® Pentium® 4 cpu 3,20 GHz
Memoria instalada (RAM)	32 bits (2Gbits)
Sistema Operativo	Ubuntu 10.04.4 LTS

Este mismo PC se utilizó para llevar a cabo las pruebas finales, ya que tenía instalado el sistema operativo para las simulaciones en tiempo real (RTAI) y las librerías de COMEDI utilizadas para la comunicación con la Tarjeta de Adquisición (DAQ). Por lo tanto, fue el sistema sobre el que se llevaron a cabo la comunicación bidireccional con la neurona electrónica y las neuronas vivas.

3.2. TARJETAS DE ADQUISICIÓN DE DATOS (DAQ)

La DAQ o tarjeta de adquisición es la interfaz encargada de la traducción de señales eléctricas analógicas a muestras digitales, y viceversa, permitiendo la comunicación entre el mundo analógico con el digital.

Este proceso de adquisición de datos mide o cuantifica un fenómeno eléctrico o físico como puede ser un voltaje, corriente, presión o cualquier otro cambio que pueda ser detectado por un sensor el cual genera una señal analógica que refleja la alteración ocurrida en el fenómeno bajo observación.

Un sistema de adquisición se compone típicamente de una serie de sensores o transductores, encargados de convertir fenómenos físicos en señales eléctricas que se puedan medir. Presentan funcionalidades como la capacidad de emisión de señales analógicas para automatizar/controlar sistemas de medida, permitiendo implementar sistemas de control y sistemas de experimentación basados en ciclos cerrados de estimulación conducida por objetivo. Pueden ser utilizadas para una gran variedad de sistemas de control o estimulación de diversos tipos como pueden ser robots, maquinaria industrial, motores de paso y otros. El hecho de incorporar un PC en estos sistemas de adquisición hace que se aumente la potencia y sobre todo la flexibilidad del sistema de medida ya que los procesadores actuales trabajan a velocidades muy altas.

Tres componentes claves caracterizan a los mecanismos de adquisición de datos:

- Circuito de acondicionamiento de Señal.
- Conversor Analógico Digital (ADC).
- Bus de PC

Actualmente se pueden encontrar tarjetas de adquisición las cuales pueden ser dotadas de nuevas funcionalidades, con distintas prestaciones tanto en la velocidad de transmisión de datos entre la tarjeta y el PC, como en las tasas de adquisición, en el tipo y rango de señales.

La tarjeta de Adquisición utilizada en este proyecto es la **NI PCI-6521**, ver figura 3.1. Esta tarjeta presenta una frecuencia máxima de 20kHz y una resolución de 16bits, haciéndose uso de ella en este proyecto a la velocidad de 10kHz.



Figura 3.1. Tarjeta de Adquisición de datos similar a la utilizada en el desarrollo del proyecto. Esta imagen corresponde a la tarjeta de un PC fuera de servicio del laboratorio del GNB

A continuación se describen algunas de las características de la tarjeta de adquisición datos, las características completas de la tarjeta utilizada se pueden consultar en la hoja de datos de su fabricante: <http://www.ni.com/pdf/manuals/371607b.pdf>

- **Características generales:**
 - Familia de productos: E / S digital
 - Tipo de Medición: Digital
 - Factor de forma: PCI
 - Sistema operativo: Linux/Windows
 - Tipo Aislamiento: Ch-Ch Aislamiento
- **E / S digital:**
 - Canales de sólo entrada: 8
 - Canales de sólo salida: 8
 - Sincronización: Software
 - Niveles lógicos: 24V
 - Filtros de entrada programables: Si
- **Entrada digital:**
 - Tipo de entrada: Sinking
Sourcing
 - Rango de Tensión máxima: -30 V - 30 V
- **Salida digital:**
 - Tipo de salida: Relay, Forma A
Relay, Forma C
Sinking
Sourcing
 - Rango de Tensión máxima: 0 V - 150 V
- **Contadores / Temporizadores:**
 - Temporizador de vigilancia: Si
- **Especificaciones físicas:**
 - Largo: 17,5 cm
 - Anchura: 9.9 cm

3.3.REAL TIME APLICATION INTERFACE (RTAI)

Los sistemas operativos en tiempo real ofrecen un control temporal estricto en la escala de milisegundos necesario para llevar a cabo las pruebas del proyecto, ya que es uno de los requerimientos del mismo para llevar a cabo la comunicación bidireccional con otros sistemas como son la neurona electrónica o la neurona viva si se da el caso.

Los sistemas operativos de propósito general, como Windows o Linux a pesar de ofrecer una variedad de herramientas y recursos, no presentan esta característica del control temporal de manera estricta para las simulaciones en tiempo real. Por esta razón, es necesario el uso de la extensión del kernel de propósito general, en el caso de este proyecto del sistema Linux utilizado.

Es el uso de Real Time Application Interface (RTAI), instalado en el PC del laboratorio de GNB y utilizado para llevar a cabo las pruebas de los distintos modelos y de la comunicación bidireccional con la neurona electrónica, el que se ha utilizado como una solución que permite el control de tiempo estricto. En el laboratorio de GNB se utiliza este sistema operativo en tiempo real como soporte para la mayoría de la investigación (Muñiz et al., 2009). A través del Hardware Abstraction Layer, el RTAI controla los componentes hardware del PC utilizado, dando prioridad a la ejecución de las instrucciones que se encuentran en una posición prioritaria frente a tareas y procesos que maneja Linux. Por lo tanto, RTAI permite la simulación de tareas soft y hard real time, además del desarrollo en modo usuario de distintas aplicaciones.

Con el uso de RTAI se tiene el control, tanto en espacio usuario como en espacio kernel, de una gran variedad de drivers que pertenecen a COMEDI, los cuales se instalan y compilan junto a RTAI. También se provee de un API de funciones, KCOMEDILIB. Todo esto permite el control y manipulación de distintas tarjetas de adquisición. Para el desarrollo del sistema de transmisión y recepción en tiempo real se ha utilizado el siguiente sistema de temporización de RTAI:

```
rt_set_oneshot_mode();
task_period_count = nano2count(task_period_ns);
timer_period_count = start_rt_timer(task_period_count);
```

Además, la instrucción `rt_task_make_periodic` hace periódica la tarea desde un origen. La descripción completa de las funciones utilizadas en este proyecto para llevar a cabo las simulaciones en tiempo real o tiempo online, de manera estricta, se pueden consultar en la documentación albergada en las webs de sus correspondientes desarrollos.

- Proyecto RTAI: <https://www.rtai.org/>
- Proyecto COMEDI: <http://comedi.org/>

3.4.COMEDI

The Control and Measurement Device Interface (COMEDI) es un proyecto de software libre en el que se desarrollan controladores para tarjetas de adquisición de datos (DAQ), herramientas y librerías para varias formas de adquisición, lectura y escritura analógica o digital, generación de pulsos, etc. Por tanto, COMEDI funciona como una interfaz entre la tarjeta de adquisición y el Sistema Operativo (SO) a partir de la combinación de un API de funciones genéricas, de una colección de controladores o drivers para un amplio rango de tarjetas y una librería de funciones para su uso y manejo, siempre de manera independiente de la tarjeta a la que pretende controlar. Una de las características más destacables de COMEDI es que puede ser integrada y utilizada en sistemas operativos de tiempo real como RTAI. El proyecto COMEDI distribuye su código en varios paquetes:

- **COMEDI:** es la colección de los controladores para un amplio número de tarjetas o devices. La funcionalidad del controlador o driver se realiza por medio de la combinación de dos módulos del sistema, el módulo COMEDI y uno específico a la tarjeta que maneje.

- **COMEDILIB:** es la librería en espacio usuario que contiene el API de funciones para la manipulación de las tarjetas. Establece un interfaz para el desarrollo de aplicaciones en espacio usuario y realiza una comunicación con los módulos de COMEDI que actúan a espacio kernel y manipula y gobierna el hardware de sistema.
- **KCOMEDILIB:** es la librería que realiza la función de interfaz amigable para el desarrollo de aplicaciones sobre tareas de tiempo real en espacio kernel de manera similar a la función realizada por COMEDILIB pero con pérdida de funcionalidad, por ejemplo las funciones de conversión de datos a valores de señal analógica no son aplicables en modo kernel.

3.5.LA NEURONA ELECTRÓNICA

Para las pruebas de comunicación bidireccional en tiempo real llevadas a cabo en este proyecto se utilizó una neurona electrónica (Pinto et al., 2000; Stiesberg et al., 2007) desarrollada en el laboratorio de GNB, ver figura 3.2. La cual implementa el modelo Hindmarsh-Rose (Hindmarsh and Rose, 1984) (Véase capítulo 4 para observar el desarrollo completo del modelo).

$$\frac{dx(t)}{dt} = ay(t) + bx(t)^2 - cx(t)^3 - dz(t) + I \quad (3.1)$$

$$\frac{dy(t)}{dt} = e - fx(t)^2 - y(t) \quad (3.2)$$

$$\frac{dz(t)}{dt} = \mu(-z(t) + S(x(t) + h)) \quad (3.3)$$

En este modelo la variable x representa el potencial de membrana, la variable y interviene en la descripción de la dinámica rápida de los potenciales de acción y la variable z en la onda lenta de la actividad en ráfagas. En estas ecuaciones a , b , c , d , e , f , μ , S y h son parámetros del modelo e I la corriente de entrada. La neurona tiene una salida analógica donde se representa el potencial de la membrana, una entrada analógica y de un potenciómetro (ver figura 3.2), ambos componentes pueden ser utilizados para derivar corriente externa en el sistema y así provocar que la neurona entre en otro régimen de excitación. Ésta puede pasar de regímenes en forma de spikes (potenciales de acción) a estados en forma de burst o grupo de spikes (ráfagas).

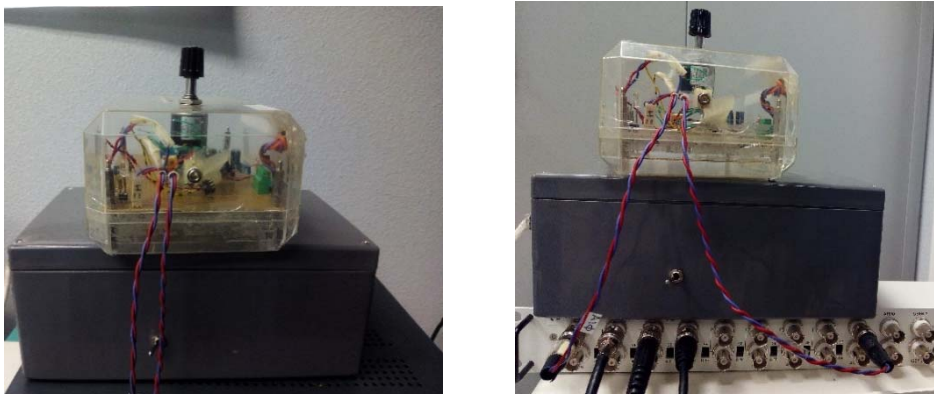


Figura 3.2. Neurona electrónica del laboratorio GNB conectada a la neurona software a través de la tarjeta de adquisición (DAQ)

3.6.NEURONA VIVA

Para llevar a cabo la validación de los circuitos híbridos se han utilizado muestras del sistema nervioso del *Carcinus maenas* (cangrejo común, ver figura 3.3).



Figura 3.3. Dibujo del *Carcinus maenas* (cangrejo común), el cangrejo utilizado para extraer muestras del sistema nervioso para las validaciones del modelo de Hindmarsh-Rose y del modelo de Izhikevich, llevado a cabo en el laboratorio del GNB

- **Preparación:**

Se ha realizado la preparación *in vitro* (figura 3.4) de las muestras obtenidas del *Carcinus maenas*.



Figura 3.4.Preparación *in vitro* de la muestra del sistema nervioso obtenida del *Carcinus Maenas*.

Se insertaron, con la ayuda del microscopio de electrofisiología, electrodos en el soma de las neuronas en busca de aquellas neuronas que presentan actividad. En los experimentos de este trabajo se utilizó la neurona PD.



Figura 3.5. La imagen de la izquierda presenta la preparación in vitro de la muestra del sistema nervioso extraída del Carcinus maenas bajo el microscopio de electrofisiología. Se observan dos electrodos conectados a la muestra. La imagen de la derecha muestra el microscopio de electrofisiología del laboratorio del GNB, bajo él se encuentra la muestra del sistema nervioso conectada a los electrodos.

A continuación se ha pasado a observar la actividad de la neurona viva, en el software disponible en el laboratorio del GNB, registrando la actividad en tiempo real de las señales procedentes del amplificador, para la selección de la neurona a utilizar y para observar el comportamiento de la neurona viva durante las pruebas de validación de los circuitos híbridos.

4. DISEÑO Y DESARROLLO

El proyecto utiliza la simulación en código C de modelos neuronales que presentan una actividad en ráfagas, con el fin de implementar circuitos híbridos llevando a cabo una interacción bidireccional online o en tiempo real entre la neurona simulada en software y una neurona implementada en hardware (neurona electrónica) o una neurona viva. Estos modelos son necesarios para estudiar la generación y coordinación de ritmos motores en estos circuitos.

Este capítulo está compuesto por dos partes principales. La primera parte contiene una pequeña introducción de cada uno de los modelos neuronales candidatos a ser implementados en software para llevar a cabo las pruebas de simulación. A demás, se presentan los dos métodos de integración numérica utilizados para la resolución de las ecuaciones diferenciales de los modelos. En la segunda parte se explica la implementación en software de los modelos expuestos anteriormente y la descripción de su comportamiento. Las simulaciones iniciales se llevan a cabo en dos PC diferentes con objeto de comprobar que se han implementado los modelos de tal manera que su simulación en diferentes equipos no afecta significativamente a los resultados finales; es decir, se podrá observar que se obtienen resultados dentro de las restricciones de tiempo deseadas.

4.1. MODELOS NEURONALES CANDIDATOS A SER IMPLEMENTADOS EN SOFTWARE

Los modelos neuronales pueden ser clasificados en:

- **Modelos abstractos:** los cuales permiten llevar a cabo análisis teóricos, ya que no simulan parámetros reales, aunque permiten observar algunos comportamientos como aprendizaje con plausibilidad biológica, oscilaciones o conectividad realista de las unidades o redes neuronales.
- **Modelos de Integrate and Fire (I&F):** en los que se simula el potencial de membrana dependiente de estímulos recibidos hasta superar un umbral, a partir del cual se genera un disparo instantáneo, para posteriormente volver a un estado subumbral. Son modelos menos complejos que los sistemas no lineales, y más adecuados siempre que no sea necesaria la simulación con detalle biofísico del potencial de acción.
- **Modelos dinámicos simplificados:** como es el caso de los modelos I&F, reducen las no linealidades presentes en el modelo Hodgkin-Huxley disminuyendo el coste computacional de simulación. Estos modelos pueden presentar un comportamiento de estados de reposo y disparos en ráfagas, repetitivos. Se caracterizan por la simplicidad en sus sistemas de ecuaciones. Aunque no son los modelos más adecuados para reproducir la forma exacta de los potenciales de acción, permiten reproducir las bifurcaciones de la dinámica y comportamientos representativos del modelo Hodgkin-Huxley.

- **Modelos realistas:** como es el caso del modelo Hodgkin-Huxley, los cuales están descritos por ecuaciones diferenciales no lineales para el potencial de membrana y las variables de conductancia, por lo que requieren de un mayor número de requisitos computacionales que los modelos anteriores, pero permiten representar de forma más exacta el comportamiento de los potenciales de acción.

En este trabajo se estudian tres modelos: el modelo de Hindmarsh-Rose, el mapa iterado de Rulkov, el modelo de Eugene Izhikevich y el modelo de conductancias tipo Hodgkin-Huxley de Thomas Nowotny. Con objeto de completar la información de este capítulo, se han llevado a cabo distintas simulaciones de los modelos implementados en lenguaje C.

4.1.1. MODELO HODGKIN-HUXLEY

El modelo Hodgkin Huxley fue presentado en 1952 por los neurofisiólogos Alan Lloyd Hodgkin y Andrew Huxley (Hodgkin and Huxley, 1952). Dicho modelo parte del desarrollo matemático basado en ecuaciones diferenciales que tenía como objetivo describir y entender la generación y propagación del potencial de acción.

Alan Lloyd Hodgkin y Andrew Huxley caracterizaron los mecanismos iónicos en el axón gigante del calamar, identificando la contribución independiente de tres tipos de iones diferentes, sodio (Na^+), potasio (K^+), y una corriente de pérdida que consiste específicamente en iones de cloro (Cl^-). En estos estudios se demostró que el sodio (Na) y el potasio (K) representan un papel importante en generación de los potenciales de acción.

El modelo Hodgkin Huxley, por lo tanto, se basa en las propiedades eléctricas de la membrana semipermeable de la neurona, que separa el interior del líquido extracelular. Las propiedades eléctricas pueden ser representadas con un circuito equivalente como el mostrado en la figura 4.1.

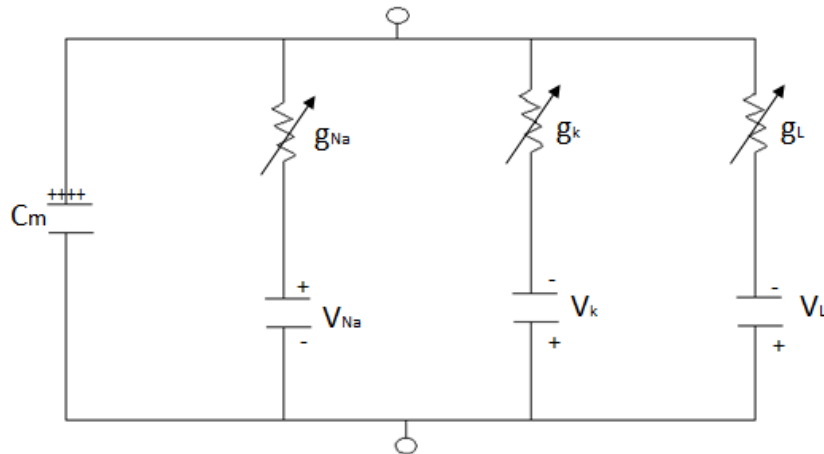


Figura 4.1. Circuito eléctrico equivalente al modelo propuesto por Hodgkin y Huxley

El modelo original Hodgkin-Huxley es un sistema de 4 ecuaciones diferenciales, en el que una de ellas modela la variación de potencial de membrana y las otras las variables de conductancia de los canales de sodio y de potasio que toman valores entre 0 y 1:

$$C_m \frac{dV}{dt} = I_{ext} - \bar{g}_L(V - V_L) - \bar{g}_{Na} m^3 h (V - V_{Na}) - \bar{g}_K n^4 (V - V_K) \quad (4.1)$$

$$\frac{dn}{dt} = \alpha_n(V)(1 - n) - \beta_n(V)n \quad (4.2)$$

$$\frac{dm}{dt} = \alpha_m(V)(1 - m) - \beta_m(V)m \quad (4.3)$$

$$\frac{dh}{dt} = \alpha_h(V)(1 - h) - \beta_h(V)h \quad (4.4)$$

4.1.1.1. DESARROLLO DEL MODELO HODGKIN-HUXLEY

Para entender mejor el desarrollo del conjunto de ecuaciones que componen el modelo Hodgkin-Huxley, a continuación se llevará a cabo un desglose de dichas ecuaciones.

- Las corrientes que fluyen a través de la membrana tienen dos componentes principales, una asociada con las propiedades capacitivas de la membrana y la otra con el flujo de tipos específicos de iones que circulan por los canales. Es decir, las corrientes iónicas de la membrana se pueden expresar como:

$$I_m = I_K + I_{Na} + I_L \quad (4.5)$$

Donde:

- I_K e I_{Na} representan las corrientes no lineales debidas al flujo de iones de potasio y sodio respectivamente.
- I_L es una corriente de fuga lineal.
- Aplicando la ley de Ohm, se calculan las conductancias iónicas individuales para dichas corrientes.

$$I_K = g_K(V - V_k) \quad (4.6)$$

$$I_{Na} = g_{Na}(V - V_{Na}) \quad (4.7)$$

Siendo:

- V el voltaje transmembrana, determinado por el circuito eléctrico (fijación de voltaje).
- V_k y V_{Na} son los potenciales de inversión de los iones de potasio y sodio medidos.

- Hodgkin y Huxley expresaron en términos de tres variables adimensionales n , m y h , que varían entre 0 y 1, la descripción de las variables de conductancia de los canales a partir de los resultados de los experimentos obtenidos con la técnica de fijación de voltaje. Suponiendo una cinética de primer orden de la forma expresada en las ecuaciones (4.1, 4.2 y 4.3), donde:
 - $\alpha_n, \alpha_m, \alpha_h$
 - $\beta_n, \beta_m, \beta_h$

Representan las tasas de cambio de la cinética de los distintos canales. Estos parámetros fueron ajustados mediante las siguientes ecuaciones empíricas:

$$\alpha_n(V) = \frac{0.01(-V + 10)}{e^{\frac{-V+10}{10}} - 1} \quad (4.8)$$

$$\beta_n(V) = 0.125e^{\frac{-V}{80}} \quad (4.9)$$

$$\alpha_m(V) = \frac{0.1(-V + 25)}{e^{\frac{-V+25}{10}} - 1} \quad (4.10)$$

$$\beta_m = 4e^{\frac{-V}{18}} \quad (4.11)$$

$$\alpha_h(V) = 0.07e^{\frac{-V}{20}} \quad (4.12)$$

$$\beta_h(V) = \frac{1}{e^{\frac{-V+30}{10}} + 1} \quad (4.13)$$

- Para ajustar las ecuaciones a los datos experimentales, se expresaron las conductancias como:

$$g_K = \bar{g}_K n^4 \quad (4.14)$$

$$g_{Na} = \bar{g}_{Na} m^3 h \quad (4.15)$$

Donde:

- \bar{g}_{Na} y \bar{g}_K Representan las conductancias máximas al sodio y al potasio.
- n y m son los parámetros de activación para las corrientes de sodio y potasio
- h parámetro de inactivación de la corriente de sodio

- También calcularon el valor de los parámetros:

$$\bar{g}_K = 36 \text{ mS/cm}^2$$

$$V_k = 12 \text{ mV}$$

$$\bar{g}_{Na} = 120 \text{ mS/cm}^2$$

$$V_{Na} = -115 \text{ mV}$$

$$\bar{g}_L = 0.3 \text{ mS/cm}^2$$

$$V_L = -10.6 \text{ mV}$$

- Por último, incorporando la capacitancia de la membrana C_m , la expresión de la corriente a través de la membrana de la ecuación (5) queda expresada como:

$$I_m = C_m \frac{dV}{dt} + I_K + I_{Na} + I_L \quad (4.16)$$

Siendo:

- I_m la corriente transmembrana
- $C_m = 1\mu\text{F/cm}^2$

Aunque este modelo ha contribuido a la neurobiología convirtiéndose en la base de los modelos matemáticos realistas, presenta una limitación al no permitir la representación de los importantes fenómenos en ráfagas de potenciales de acción.

4.1.2. MODELO DE CONDUCTANCIA DE THOMAS NOWOTNY et al.

El modelo de neurona tipo Hodgkin-Huxley desarrollado por Thomas Nowotny, Rafael Levi y Allen I. Selverston, presentado en el artículo "Probing the Dynamics of Identified Neurons with a Data-Driven Modeling Approach" (Nowotny et al., 2008), es un modelo de conductancia detallado tipo HH. Está basado en la conductancia de las células de la langosta (*Panulirus interruptus*) pilórica lateral (LP) a partir de técnicas de estimación de parámetros llevando a cabo el bloqueo seleccionado de las corrientes iónicas. El modelo es capaz de reproducir las bifurcaciones en el comportamiento observadas en esta neurona en función de diferentes estímulos. Los resultados obtenidos permiten indicar que los modelos de datos impulsados son herramientas útiles para el análisis en profundidad de la dinámica neuronal (Nowotny et al., 2008).

Es así que este modelo permite la obtención de los valores del potencial de membrana de los compartimientos del soma y del axón, los cuales se describen como:

$$\frac{dV_{axon}}{dt} = \frac{1}{C_a} (-I_{Na} - I_{Kd} - I_M - I_{leak,a} + I_{vv}) \quad (4.17)$$

$$\frac{dV_{soma}}{dt} = \frac{1}{C_s} (-I_{Ca} - I_{KCa} - I_A - I_h - I_{leak,s} - I_{vv} + I_{scale} (I_{Dc} - I_{offset} + I_{syn})) \quad (4.18)$$

Las corrientes están agrupadas en dos compartimientos, I_{Ca} , I_{KCa} , I_A y I_h , además de la corriente de fuga $I_{leak,a}$, en un compartimiento denotado como 'soma'; y las corrientes I_{Na} , I_{Kd} , I_M y la corriente de fuga $I_{leak,a}$, en un compartimiento denotado como 'axón'. La corriente de calcio, I_{Ca} , está compuesta a su vez por otras dos corrientes, en particular, por la corriente transitoria de calcio, I_{CaT} , y la corriente lenta de calcio, I_{CaS} (Nowotny et al., 2008). La descripción en dos compartimientos permite separar las variables lentas y rápidas y reproducir mejor la dinámica observada en la neurona LP.

4.1.2.1. DESARROLLO DEL MODELO DE CONDUCTANCIA DETALLADO DE NOWOTNY et al.

A continuación se presentan las ecuaciones desarrolladas para este modelo de neurona, las cuales permiten obtener los valores del potencial de membrana del axón y del soma anteriormente presentados llevando a cabo el bloqueo seleccionado de las corrientes.

Las corrientes I_{Na} , I_{Kd} , I_M , I_{KCa} , y I_h están descritas por:

$$I_x = g_x m^p h^q (V - V_x) \quad (4.19)$$

La activación y desactivación de las corrientes presentes en el compartimiento axón (I_{Na} , I_{Kd}) está controlada por la siguiente dinámica:

$$\frac{dm_x}{dt} = (m_{\infty x}(V) - m_x) k_{mx} \quad (4.20)$$

$$\frac{dh_x}{dt} = (h_{\infty x}(V) - h_x) k_{hx} \quad (4.21)$$

El control de las conductancias de las corrientes (I_{CaT} , I_{CaS} , I_{KCa} , I_A , I_h y I_M) es llevado a cabo por las siguientes ecuaciones:

$$\frac{dm}{dt} = a_m(1 - m) - b_m m \quad (4.22)$$

$$\frac{dh}{dt} = a_h(1 - h) - b_h h \quad (4.23)$$

El resto de las corrientes se hallan a partir de las siguientes definiciones:

- $I_{offset} = 2nA$
- I_{syn} representa la corriente sináptica de entrada total
- I_{Dc} toma valores variables

$$I_{leak} = g_{leak}(V - V_{leak}) \quad (4.24)$$

$$I_{vv} = g_{vv}(V_{soma} - V_{axon}) \quad (4.25)$$

$$I_{Ca} = (g_{CaT}m_{CaT}h_{CaT} + g_{CaS}m_{CaS}) \frac{[Ca] \exp\left(\frac{V_{soma}}{RT/F}\right) - [Ca]_{out}}{\exp\left(\frac{V_{soma}}{RT/F}\right) - 1.0} P_{Ca} V_{soma} \quad (4.26)$$

Donde,

$$\frac{d[Ca]}{dt} = -C_{ica}I_{Ca} - K_{Ca}([Ca] - [Ca]_o) \quad (4.27)$$

$$I_A = g_A m_A (a_{hA1} + (1 - a)h_{A2})(V_{soma} - V_A) \quad (4.28)$$

Siendo,

$$a = \frac{1}{(1 + \exp\left(\frac{V_{soma} - V_{aA}}{S_{aA}}\right))} \quad (4.29)$$

Las variables como $m_{\infty x}(V)$, $h_{\infty x}(V)$ además de ser dependientes del voltaje, representan las tasas de cambio de la cinética de los distintos canales. Los valores de todos los parámetros que anteriormente no se han especificado se encuentran detallados en el artículo (Nowotny et al., 2008).

4.1.3.MODELO FITZHUGH-NAGUMO

En 1961 el fisiólogo Richard FitzHugh propuso el modelo llamado “Bonhoeffer-van der Pol”, basado en la ecuación del oscilador de Van der Pol (FitzHugh, 1961). Un año después, este modelo fue implementado mediante un circuito electrónico diseñado por el ingeniero japonés J.Nagumo (Nagumo et al., 1962).

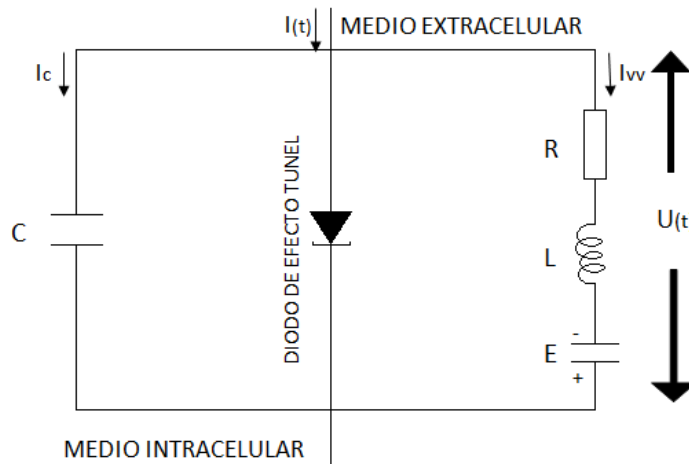


Figura 4.2. Circuito eléctrico equivalente de Nagumo

Este comportamiento fue modelado en las siguientes ecuaciones:

$$\frac{du(t)}{dt} = f(u(t)) - w(t) - I(t) \quad (4.30)$$

$$\frac{dw(t)}{dt} = a(b * u(t) + d - c * w(t)) \quad (4.31)$$

Donde:

$$f(u(t)) = (u(t)) - \frac{u(t)^3}{3} \quad (4.32)$$

- $u(t)$ es una variable rápida que modela la tensión de membrana mediante un polinomio de tercer orden.
- w es una variable lenta o de recuperación.
- $I(t)$ representa una corriente externa aplicada al sistema
- $a = 0.08$
- $b = 1$
- $c = 0.8$
- $d = 0.7$

Como resultado, se obtuvo un modelo simplificado compuesto por 2 ecuaciones (una lineal y otra no lineal), del modelo propuesto por Hodgkin-Huxley original que se componía de 4 ecuaciones, en el que se eliminan las variables m y n que modelan la apertura de los canales iónicos del sodio y potasio.

4.1.4. MODELO HINDMARSH-ROSE

El modelo Hindmarsh-Rose (Hindmarsh and Rose, 1984), surgió en 1982 como una modificación del modelo propuesto por FitzHugh y Nagumo. Reemplazaron la función lineal por una cuadrática. En 1984 se modificó el modelo inicial, convirtiéndose en un sistema de 3 ecuaciones diferenciales no lineales capaz de reproducir distintas actividades en ráfaga incluyendo un régimen caótico similar al observado en las neuronas reales.

$$\frac{dx(t)}{dt} = ay(t) + bx(t)^2 - cx(t)^3 - dz(t) + I \quad (4.33)$$

$$\frac{dy(t)}{dt} = e - fx(t)^2 - y(t) \quad (4.34)$$

$$\frac{dz(t)}{dt} = \mu(-z(t) + S(x(t) + h)) \quad (4.35)$$

Donde:

- $x(t)$ Representa el potencial de membrana neuronal
- $y(t)$ es una corriente rápida
- $z(t)$ es una corriente lenta ($\mu \ll 1$)
- $a, b, c, d, e, f, \mu, S, h$ Son parámetros adimensionales ajustables, utilizados para determinar el comportamiento de la neurona simulada.

Para la simulación de una neurona electrónica a partir del modelo Hindmarsh-Rose se asignaron los siguientes valores a los parámetros anteriores:

- $a = 1, b = 3, c = 1, d = 1, e = 1, f = 5, h = 1.6, \mu = 0.0021, S = 4$

Dependiendo del valor elegido para el parámetro I , el modelo puede presentar un comportamiento en régimen regular ($I=3.0$) o un comportamiento en régimen no regular/caótico ($I=3.281$) (Rodríguez et al., 2001).

En conclusión, el modelo Hindmarsh-Rose teniendo una estructura más sencilla es capaz de demostrar una amplia variedad de comportamientos y bifurcaciones observadas en las neuronas (Szücs et al., 2000).

4.1.5. MAPA DE RULKOV

El modelo propuesto por Nikolai F. Rulkov (Rulkov, 2002), consiste en un mapa de las dos variables más representativas del comportamiento neuronal. Se trata de un modelo matemáticamente simple que permite estudiar tres regímenes estables a partir de la combinación de sus parámetros:

- **Silencioso:** El potencial de membrana se mantiene en un estado de descanso constante
- **Disparo tónico:** La neurona emite disparos a una frecuencia constante

- **Ráfaga tónica:** Se emiten ráfagas de disparo a una frecuencia constante entre intervalos de silencio.

Este modelo puede ser definido mediante las siguientes ecuaciones:

$$x_{n+1} = f(x_n, y_n + eI_n) \quad (4.36)$$

$$y_{n+1} = y_n - \mu(x_n + 1) + \mu\sigma + \mu\sigma_e I_n \quad (4.37)$$

$$f(x, y) = \begin{cases} \frac{\alpha}{1-x} + y, & x \leq 0 \\ \alpha + y, & 0 \leq x < \alpha + 1 \\ -1, & x > \alpha + y \end{cases} \quad (4.38)$$

En este modelo bidimensional, la variable x_n representa el voltaje de la membrana de la neurona, y_n se asigna a la variable de dinámica lenta sin significado biológico directo.

La combinación de α y σ selecciona el régimen de funcionamiento del modelo, además de varias propiedades neuronales en el caso de estar en el régimen de ráfagas (Rulkov, 2002).

4.1.6. MODELO IZHKEVICH

En 2003 Eugene Izhikevich presentó un nuevo modelo neuronal compuesto por dos ecuaciones (Izhikevich, 2003). Este modelo representa un tipo de neurona pulsante que emite un pulso cuando la tensión de membrana supera un determinado nivel de disparo. Se presentó como una alternativa a los modelos de tipo integrate and fire.

$$\frac{dv(t)}{dt} = 0.04v^2 + 5v + 140 - u + I \quad (4.39)$$

$$\frac{du(t)}{dt} = a(bv - u) \quad (4.40)$$

En el caso en el que se prefije el nivel de disparo a 30mV, las variables u y v toman los siguientes valores:

$$\text{if } v \geq 30\text{mV} \quad \text{Entonces } \begin{cases} v \rightarrow c \\ u \rightarrow u + d \end{cases} \quad (4.41)$$

Donde:

- $u(t)$ y $v(t)$ Son variables:
 - $u(t)$ Representa la variable de recuperación de la membrana. Provee una retroalimentación negativa para $v(t)$.

- a, b, c y d son dos parámetros del modelo:
 - a Describe la escala de tiempo para la variable de recuperación $u(t)$. Valores pequeños significan una recuperación lenta. Parámetro adimensional.
 - b describe la sensibilidad de la variable de recuperación $u(t)$ a las fluctuaciones subumbrales del potencial de membrana $v(t)$. Parámetro adimensional.
 - c describe el valor de retorno del potencial de membrana $v(t)$ después del potencial de acción, causado por las conductancias de potasio rápidas y de alto umbral. Es un parámetro con unidades mV.
 - d describe el valor de retorno de la variable de recuperación $u(t)$ después del potencial de acción, causado por las conductancias de sodio y potasio, lentas y de alto umbral. Parámetro adimensional.

La elección de diferentes valores para los parámetros adimensionales permite representar distintos patrones de disparo, obteniéndose los siguientes tipos de representación de neuronas:

- **Neuronas Corticales Excitatorias:**

Tabla 4.1. Los modelos RS, IB y CH corresponden a neuronas corticales excitatorias:

Tipo de neurona	Valor de parámetros
RS (Regular Spiking)	a=0,02 b=0,2 c= -65 mV d=8
IB (Intrinsically Bursting)	a=0,02 b=0,2 c= -55 mV d=4
CH (Chattering)	a=0,02 b=0,2 c= -50 mV d=2

- **Interneuronas Corticales Inhibitorias:**

Tabla 4.2. Los modelos FS y LTS a interneuronas corticales inhibitorias

Tipo de neurona	Valor de parámetros
FS (Fast Spiking)	a=0,1 b=0,2 c= -65 mV d=2
LTS (Low-Threshold Spiking)	a=0,02 b=0,25 c= -65 mV d=2

- **Neuronas Tálamo-Corticales:**

Tabla 4.3. Neuronas tálamo-corticales:

Tipo de neurona	Valor de parámetros
TC (Thalamo-Cortical)	a=0,02 b=0,25 c= -65 mV d=0,05
RZ (Resonator)	a=0,1 b=0,25 c= -65 mV d=0,05

4.2.MÉTODOS DE INTEGRACIÓN NUMÉRICA DE LOS MODELOS

Para la simulación de los modelos matemáticos explicados a lo largo del apartado 4.1 ha sido necesario utilizar los siguientes métodos numéricos para su resolución (Simmons, 2007; Zill, 2008), pues la mayoría de los modelos descritos anteriormente corresponden a ecuaciones diferenciales ordinarias (EDOs) acopladas, a excepción del mapa iterado de Rulkov cuya resolución es directa.

Estos métodos matemáticos utilizan integración numérica para la resolución de las EDOs, por lo que es imprescindible elegir un paso de integración adecuado. Para el desarrollo de este proyecto se ha utilizado un paso fijo en lugar de uno variable, debido a las restricciones que presenta llevar a cabo un muestreo de datos en tiempo real, de forma que la neurona software sincronice su comportamiento con el de la neurona viva. La elección de cada valor en un paso variable añadiría un tiempo de espera o retraso en la generación de las muestras. Por lo tanto, para la resolución de las EDOs de los modelos de neuronas implementados en este proyecto se ha utilizado los métodos de Euler y Runge-Kutta de cuarto orden, ambos con un paso de integración fijo. En apartados posteriores se puede observar la comparación de los resultados de las simulaciones utilizando uno u otro método de resolución (*véase apartado 4.3.2.2*).

4.2.1.MÉTODO DE EULER

El método de Euler (Simmons, 2007; Zill, 2008), es un procedimiento de integración numérica para la resolución de ecuaciones diferenciales ordinarias a partir de unos valores iniciales dados. Fue llamado así en honor a Leonhard Paul Euler. Se basa en la pendiente estimada de la función objeto de estudio, para extrapolar desde un valor anterior a uno nuevo.

Suponiendo un problema descrito por una EDO y de un valor inicial:

$$y' = f(x, y) \quad (4.42)$$

$$y(x_0) = y_0$$

Se puede integrar de x_0 a $x_1 = x_0 + h$, obteniendo

$$y(x_1) - y(x_0) = \int_{x_0}^{x_1} f(x, y) dx \quad (4.43)$$

En este punto, se observa que la función desconocida y está presente en el integrando de la derecha $f(x, y)$, por lo que se hace necesario un método de aproximación de la integral, es a partir de esta técnica que se obtiene el método de Euler.

Suponiendo que el intervalo $[x_0, x_1]$ no varía demasiado, se obtendrá un valor pequeño al sustituir $f(x, y)$ por su valor en el punto extremo izquierdo.

$$y(x_1) = y(x_0) + \int_{x_0}^{x_1} f(x, y) dx \quad (4.44)$$

$$\approx y(x_0) + \int_{x_0}^{x_1} f(x, y) dx \approx y(x_0) + hf(x_0, y_0) \quad (4.45)$$

Siendo h la longitud del intervalo $[x_{j-1}, x_j]$

Sobre la base de estos cálculos se define:

$$y_1 = y_0 + h \cdot f(x_0, y_0) \quad (4.46)$$

Estableciendo: $x_2 = x_1 + h$, se puede definir:

$$y_2 = y_1 + hf(x_1, y_1) \quad (4.47)$$

De manera general, estableciendo $x_j = x_{j-1} + h$, se define:

$$y_{j+1} = y_j + hf(x_j, y_j) \quad (4.48)$$

Fijando $h \neq 0$, es posible obtener aproximaciones de la solución del problema de valores iniciales en los puntos x_1, x_2, \dots, x_n donde $x_i = x_{i-1} + h, i=1,2,\dots, n$, mediante el método recurrente:

$$y_i = y_{i-1} + h \cdot f(x_{i-1}, y_{i-1}), \quad i=1,2,\dots,n \quad (4.49)$$

4.2.2.MÉTODO DE EULER MEJORADO

A partir de la primera parte de la Ecuación (4.42), con el objeto de mejorar el método de Euler (Simmons, 2007; Zill, 2008), se reemplaza el integrando por:

$$[f(x_0, y_0) + f(x_1, y(x_1))]/2 \quad (4.50)$$

Se obtiene así

$$y_1 = y_0 + \frac{h}{2}[f(x_0, y_0) + f(x_1, y(x_1))] \quad (4.51)$$

Representando el nuevo valor por medio de $z_1 = y_0 + h \cdot f(x_0, y_0)$ la ecuación se convierte en:

$$y_1 = y_0 + \frac{h}{2}[f(x_0, y_0) + f(x_1, z_1)] \quad (4.52)$$

En general, el esquema de recurrencia queda de la siguiente forma:

$$y_{j+1} = y_j + \frac{h}{2}[f(x_j, y_j) + f(x_{j+1}, z_{j+1})] \quad (4.53)$$

Siendo

$$z_{j+1} = y_j + h \cdot f(x_j, y_j), \quad j=0, 1, 2... \quad (4.54)$$

4.2.3.MÉTODO RUNGE-KUTTA

Los métodos de Runge-Kutta (RK) son un conjunto de métodos iterativos (implícitos y explícitos) enfocados en la resolución de ecuaciones diferenciales ordinarias. Fueron desarrollados por los matemáticos Carl Runge y Martin Wilhelm Kutta, consiste en aproximar la integral sustituyendo el integrando por una parábola.

Existen variantes del método de Runge-Kutta, como Runge-Kutta de segundo, tercer y cuarto orden, se dice que el método de Euler es un método de Runge Kutta de primer orden, y las parejas de métodos Runge-Kutta (o métodos Runge-Kutta-Fehlberg). Este último consiste en ir aproximando la solución de la ecuación mediante dos algoritmos Runge-Kutta de órdenes diferentes, para así mantener el error acotado y hacer una buena elección de paso.

Para llevar a cabo las pruebas de los modelos implementados en este proyecto se ha utilizado el método de **Runge-Kutta de cuarto orden**.

En el caso de Runge-Kutta de cuarto orden, fijando $h \neq 0$, es posible obtener aproximaciones de la solución del problema de valores iniciales de la Ecuación 18 en los puntos x_1, x_2, \dots, x_n donde $x_i = x_{i-1} + h, i=1, 2, \dots, n$, mediante el método recurrente:

$$y_i = y_{i-1} + \frac{1}{6}(p_i + 2q_i + 2r_i + s_i), \quad i=1,2,\dots,n \quad (4.55)$$

Donde:

$$p_i = f(x_{i-1}, y_{i-1})h \quad (4.56)$$

$$q_i = f\left(x_{i-1} + \frac{1}{2}h, y_{i-1} + \frac{1}{2}p_i\right)h \quad (4.57)$$

$$r_i = f\left(x_{i-1} + \frac{1}{2}h, y_{i-1} + \frac{1}{2}q_i\right)h \quad (4.58)$$

$$s_i = f(x_{i-1} + h, y_{i-1} + r_i)h \quad (4.59)$$

4.3. DISEÑO INICIAL DE LOS MODELOS NEURONALES

Como se ha comprobado en el capítulo del Estado del Arte, existen varios modelos neuronales propuestos. En los apartados anteriores se han expuesto los modelos matemáticos que presentan más interés para este proyecto, los cuales son Hodgkin-Huxley, Nowotny et al., Fitzhugh Nagumo, Hindmarsh-Rose, Izhikevich y el Mapa de Rulkov.

Para cumplir con el objetivo de este proyecto sólo se requiere utilizar aquellos modelos que presenten actividades en ráfagas, en las que se simula el potencial de membrana neuronal, pues se pretende que la implementación de estos modelos sirva de base para futuros desarrollos de circuitos híbridos en los que la comunicación entre la neurona biológica y la electrónica será más sencilla si se lleva a cabo mediante transmisiones de potenciales en ráfagas. También es importante que las distintas implementaciones aquí desarrolladas puedan ser simuladas en distintos PC. Para esto, se ha llevado a cabo el desarrollo inicial en un PC con las características expuestas en el capítulo 3 de Equipamiento (véase apartado 3.1.1). Estas mismas pruebas se han realizado nuevamente en el PC del laboratorio de GNB, cuyas características pueden observarse también en el capítulo 3 de Equipamiento (véase apartado 3.1.2).

En este proyecto además de las simulaciones que se presentan a continuación, también se ha llevado a cabo la implementación de los modelos de Hindmarsh-Rose e Izhikevich utilizando el sistema RTAI, COMEDI y la Tarjeta de Adquisición. Por lo tanto, estos nuevos diseños se han utilizado para realizar la interacción bidireccional con la neurona electrónica y la neurona viva. Estos diseños y los resultados se describen en los siguientes capítulos (véase capítulo 5).

4.3.1. PLANTEAMIENTOS INICIALES

Para el desarrollo inicial de los modelos neuronales se ha tenido en cuenta las posibles limitaciones a cumplir con el objeto de posibilitar el diseño de futuros circuitos híbridos, estas limitaciones son:

- El uso de la DAQ (Tarjeta de Adquisición de datos), a partir de la cual todo el proceso de adquisición de señales de las neuronas vivas y el envío de estímulos presenta una frecuencia de muestreo determinada.
- Posibilidad de simulación en distintos equipos con velocidades de procesamiento distintas.
- Uso de modelos neuronales cuyo comportamiento se lleva a cabo en ráfagas con una duración en tiempo real que ha de ser entorno a un segundo.

Teniendo en cuenta los puntos anteriores, la implementación de los modelos neuronales debe realizarse de tal manera que se genere una ráfaga de potenciales con un número de puntos determinado por la frecuencia de transmisión de la tarjeta de adquisición (DAQ) utilizada. En este proyecto se utilizará la tarjeta de adquisición del laboratorio de GNB, la cual presenta una frecuencia de 10kHz (véase apartado 3.2 para observar las características que presenta la tarjeta). Las simulaciones generarán 10.000 puntos por segundo, los cuales contienen el valor de potenciales de membrana.

Es importante recordar que en este capítulo se lleva a cabo la implementación de los modelos neuronales sin realizar la interacción bidireccional con la neurona electrónica o la neurona viva, las simulaciones generan un único archivo en el que se guardan los **datos de salida** correspondientes a los potenciales de membrana producidos por la neurona simulada.

Para que se puedan realizar las simulaciones en distintos equipos se impone la limitación de tiempo de ejecución de una ráfaga en un tiempo aproximado a un segundo (tiempo real). Como podrá observarse más adelante, los modelos neuronales elegidos se ejecutan de manera muy rápida, en tiempos inferiores a un segundo, como es el caso del Mapa de Rulkov; o en tiempos muy superiores al segundo deseado, en este caso están enmarcados los modelos que usan muchas exponenciales en su descripción, lo que hace que sus simulaciones tomen varios segundos. La mayor parte de los modelos neuronales planteados, a excepción del Mapa de Rulkov, presentan un sistema de ecuaciones diferenciales cuya resolución no es trivial. Esto hace necesario el uso de métodos numéricos de integración como el de Euler o el de Runge Kutta, en los que es muy importante la correcta elección del paso de integración.

Teniendo esto en cuenta, se plantean dos opciones para elegir como método numérico de resolución de los modelos:

- **El método de Euler** permite un desarrollo con menos operaciones, por lo que la carga operacional sobre el sistema en cada paso es menor, pero limita el proceso a usar pasos de integración pequeños, generándose más valores.

- **El método de Runge Kutta** permite el uso de paso de integración mayor a costa de producir una mayor carga en el sistema debido al mayor número de operaciones que este método necesita.

Se ha intentado desarrollar un código sencillo para obtener una simulación optimizada evitando en la medida de lo posible el coste en tiempo de ejecución. Además, en el momento de compilación se ha recurrido al uso de las opciones de optimización (-O2 y -march) que presenta el compilador utilizado.

4.3.2.PROCESO DESARROLLADO

Teniendo en cuenta las limitaciones planteadas para la implementación de los modelos, el desarrollo inicial de los mismos se ha llevado a cabo en varias etapas y utilizando el **PC para desarrollos iniciales** cuyas características pueden ser observadas en el apartado 3.1.1.

4.3.2.1. ETAPA 1- SIMULACIÓN DE LA ACTIVIDAD DE LOS MODELOS NEURONALES DE INTERÉS

En esta etapa se ha implementado en código C los modelos neuronales de Hodgkin-Huxley, Nowotny et al., Fitzhugh- Nagumo, Eugene Izhikevich, Hindmarsh-Rose y el Mapa de Rulkov y **se han representado sólo los modelos que presentan un comportamiento en ráfaga en dicho potencial en un tiempo inferior a un segundo (modelos rápidos), siendo los adecuados para implementar circuitos híbridos con neuronas que presentan actividad también en ráfagas.**

Los resultados de este estudio se resumen en la tabla 4.4, donde se presentan las representaciones de los valores del potencial de membrana obtenidos de las simulaciones de los modelos neuronales propuestos. **Se toma como referencia el modelo de Hindmarsh-Rose y los resultados obtenidos de su simulación**, por lo que se realizan todas las simulaciones utilizando los mismos valores que en éste modelo. Es decir, en el modelo Hindmarsh-Rose se consigue representar una ráfaga completa utilizando un paso de integración fijo de 0,001 (el paso mínimo considerado adecuado para el método de integración de Euler en este modelo) con 280.000 valores por ráfaga (el número de puntos necesarios para representar una ráfaga es diferente para cada modelo). El tiempo aproximado de ejecución en un ordenador con un procesador Intel® Core™ i3-4150 CPU @ 3.50GHz es de 0,2 segundos. Para permitir su comparación, la integración de los otros modelos se muestran en la misma escala temporal de 0,2 segundos, utilizando el método numérico de Euler y con el mismo paso de integración fijo.

Para la resolución de las ecuaciones de dichos modelos se ha utilizado el **PC para desarrollos iniciales** (véase apartado 3.1.1) y el **método de Euler**.

Tabla 4.4. Representación del potencial de membrana de la simulación de los modelos neuronales: Hindmarsh-Rose, Eugene Izhikevich y el Mapa de Rulkov; utilizando para su resolución el método de Euler, con un paso de integración fijo de 0,001. Estas representaciones permiten comprobar que dichos modelos presentan un comportamiento de ráfaga en su potencial de acción.

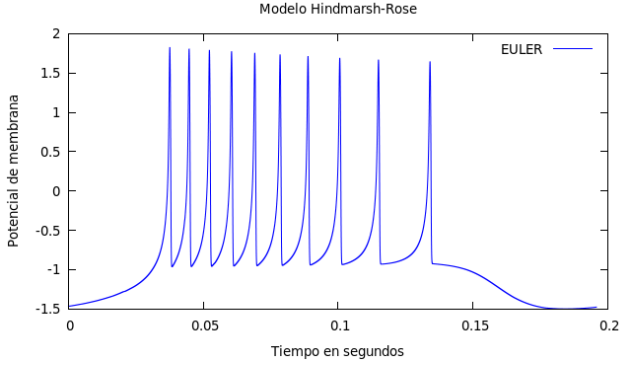
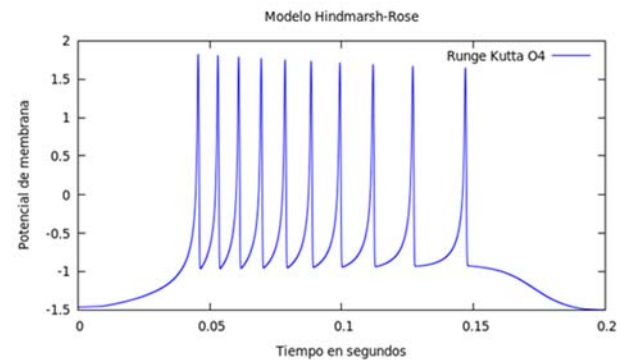
Modelo	Variables	Representación
HINDMARSH-ROSE	$E=3,0$ (Regular)	
IZHIKEVICH	CH $a=0,02$ $b=0,2$ $c=-50$ $d=2$	
MAPA DE RULKOV		

El modelo planteado por E.Izhikevich presenta distintos comportamientos, dependiendo de los valores de las variables a , b , c y d elegidos. Para el desarrollo de este proyecto ha sido necesario elegir el comportamiento **CHATTERING (CH)**, el cual presenta actividad en ráfagas, siendo $a=0.02$, $b=0.2$, $c=-50$ y $d=2$. Destacar que en el caso de la simulación del Mapa de Rulkov, éste no necesita el uso de ninguno de los modelos matemáticos expuestos en el apartado 4.2 (Euler, Runge-Kutta), ni depende del paso de integración, por lo que se considera como caso especial.

4.3.2.2. ETAPA 2- ELECCIÓN DEL MÉTODO MATEMÁTICO A UTILIZAR: EULER O RUNGE-KUTTA

Es importante tener en cuenta que el uso del método matemático de Runge Kutta incrementa la carga computacional en cada paso de integración de cada modelo debido al aumento del número de operaciones que tiene que realizar la simulación. El método de Euler por otro lado impone la necesidad del uso de pasos de integración pequeños, por lo que lleva a generar un mayor número de puntos por simulación. Teniendo en cuenta la precisión, el comportamiento de la integración con diferentes modelos y el coste computacional de cada método, en este trabajo se ha optado por utilizar el método de Euler para integrar las ecuaciones.

Tabla 4.5. Representación del potencial de membrana generado por el modelo Hindmarsh-Rose y el tiempo de ejecución total en segundos. La representación superior muestra los resultados obtenidos de resolver el modelo de Hindmarsh-Rose utilizando el método de Euler con un paso de integración fijo de 0,001. En la representación inferior se ha resuelto el modelo utilizando el método de Runge-Kutta de cuarto orden con paso de integración fijo de 0,001. Ambas representaciones permiten comparar la diferencia entre los tiempos de simulación según el modelo matemático utilizado en la implementación del modelo Hindmarsh-Rose con paso de integración fijo 0,001.

Modelo matemático utilizado	Representación
EULER	
RUNGE KUTTA (ORDEN 4)	

Los modelos elegidos para el desarrollo de este proyecto son modelos de generación rápida, por lo que el uso de uno u otro método matemático de los que se han expuesto anteriormente no producen importantes cambios en aspectos temporales de las simulaciones. Se ha elegido el método de Euler por las características que presenta, y que han sido expuestos en párrafos anteriores, pensando en los futuros trabajos en los que se propone la implementación de modelos lentos.

4.3.2.3. ETAPA 3- PASO DE INTEGRACIÓN

A partir de los resultados obtenidos en el apartado anterior, se ha modificado el código para el modelo de Hindmarsh-Rose y el modelo de Izhikevich, los cuales presentan actividad en ráfagas, eligiendo el paso de integración que mejor ajusta la simulación a tiempos cercanos al tiempo de simulación elegido. En este caso, el objetivo es ajustar los resultados obtenidos a un segundo para que los modelos puedan cumplir con las restricciones de tiempo real impuestas por la neurona viva. En las siguientes tablas, se observa que la variación del paso de integración aumenta o disminuye el tiempo que tarda el modelo en calcular cada punto.

Se ha llevado a cabo la implementación de los modelos utilizando el método de integración numérica de **Euler**, variando el valor del paso de integración fijo para cada representación. **Se ha tomado como referencia los valores conocidos de la simulación del modelo de Hindmarsh-Rose**, por lo que se conoce que en este modelo se genera una ráfaga completa de 280.000 valores en un tiempo inferior a un segundo utilizando un paso de integración fijo de valor 0,001. Disminuyendo este valor se generan menos puntos y se disminuye el tiempo en el que se produce la simulación, si se aumenta el paso ocurre lo contrario, el número de valores obtenidos de las integraciones del modelo aumenta y con ello aumenta el tiempo de simulación. Como se observará, siempre se representa una ráfaga.

Tabla 4.6. Se representa el potencial de membrana obtenido de la simulación del modelo de Hindmarsh-Rose, utilizando el método de Euler, para diferentes valores de paso de integración fijo y el tiempo de ejecución total en segundos, con el objetivo de elegir el valor de paso que mejor aproxime la simulación del modelo a un tiempo cercano pero inferior de un segundo. Se representan un total de 280.000 puntos, los cuales para el modelo de Hindmarsh-Rose supone la simulación de una ráfaga. La representación (A.) muestra los resultados del uso de un paso de integración fijo de 0,01. (B.) representa los resultados obtenidos con un paso fijo de 0,001 y (C.) muestra los resultados para el paso fijo de 0,0001. Con objeto de realizar la comparación entre gráficas, se realiza la representación temporal durante 2 segundos

	Paso de integración	Tiempo de simulación (segundos)	Representación
A.	0,01	0,023726	
B.	0,001	0,199488	
C.	0,0001	1,929467	

En el caso del modelo Hindmarsh-Rose, el paso de integración que presenta la simulación en un tiempo inferior pero no lejano a un segundo es para el valor de **0,001**.

En el caso del modelo de Izhikevich se toma la misma referencia de 280.000 valores y el paso fijo de 0,001 como punto de partida, pero en este caso se representan cuatro ráfagas, más adelante se podrá comprobar el valor correcto para representar una única ráfaga en este modelo. Se lleva a cabo el mismo proceso, aumentando y disminuyendo el valor del paso de integración hasta encontrar aquel que mejor ajusta las representaciones del modelo en valores de tiempo inferiores al segundo pero muy cercanas al mismo.

Tabla 4.7. Se representa el potencial de membrana obtenido de la simulación del modelo de Izhikevich, utilizando el método de Euler, para diferentes valores de paso de integración fijo y el tiempo de ejecución total en segundos, con el objetivo de elegir el valor de paso que mejor aproxime la simulación del modelo a un tiempo cercano pero inferior de un segundo. Se representan un total de 280 000 puntos, los cuales para el modelo de Izhikevich, a diferencia del modelo anterior, supone la simulación de cuatro ráfagas. La representación (A.) muestra los resultados del uso de un paso de integración fijo de 0,01. (B.) representa los resultados obtenidos con un paso fijo de 0,001 y (C.) muestra los resultados para el paso fijo de 0,0001. Con objeto de realizar la comparación entre gráficas, se realiza la representación temporal durante 2 segundos

	Paso de integración	Tiempo de simulación (segundos)	Representación
A.	0,01	0,026612	
B.	0,001	0,234089	

Paso de integración		Tiempo de simulación (segundos)	Representación
C.	0,0001	2,243271	

El paso de integración que mejor aproxima el tiempo de ejecución de la simulación del modelo de Izhikevich es el **0,0001**.

4.3.2.4. ETAPA 4- AJUSTE TEMPORAL DEL SISTEMA Y MUESTREO

Para las simulaciones de los modelos con actividad en ráfagas es necesario que el total de puntos generados en cada simulación suceda dentro de un tiempo determinado para que se pueda garantizar su correcta ejecución en distintos equipos y para que cumpla las restricciones de tiempo real impuestas por la futura conexión con una neurona viva, siendo elegido el valor de un segundo aproximadamente. Por lo tanto, se ha recurrido a la función **nanosleep ()** (véase ANEXO B) para realizar las paradas entre cada punto y así ajustar el tiempo total. A su vez, se han muestreado 10.000 puntos, valor determinado por la tarjeta de adquisición utilizada a 10kHz.

Es importante destacar que para las implementaciones de los modelos utilizando RTAI y COMEDI, la función **nanosleep ()** será sustituida por las funciones que el sistema en tiempo real proveen (véase capítulo 5).

4.3.2.4.1. DESARROLLO DEL MÉTODO DE PARADAS DEL SISTEMA UTILIZADO

El método matemático elegido es el método de **Euler**. Además, se ha observado que para simular una sola ráfaga en el modelo de Hindmarsh-Rose se requiere de **280.000 puntos y un paso de integración de valor 0,001**, a diferencia del modelo de Izhikevich que genera una única ráfaga con **1.000.000 puntos para el paso de integración de valor 0,0001**. El número de valores necesarios para representar una ráfaga completa puede variar dependiendo del sistema que se utilice para su simulación, por lo que se puede hablar de **n valores** en su lugar.

Los pasos de integración para que las simulaciones ocurran en un tiempo inferior a un segundo para los dos modelos seleccionados (Hindmarsh-Rose e Izhikevich) son:

Tabla 4.8. Valores de paso de integración fijo seleccionados para los modelos de Hindmarsh-Rose e Izhikevich utilizados en su resolución utilizando el método de Euler

Modelo	Paso de integración	Método matemático
HINDMARSH-ROSE	0,001	Euler
IZHIKEVICH	0,0001	Euler

En la Tabla 4.9 se puede observar el tiempo de generación en el que se espera que ocurra cada punto para obtener una simulación de todos los puntos en un segundo. Como se ha demostrado anteriormente, estos modelos se generan en tiempos muy inferiores al valor deseado por lo que es necesario llevar a cabo paradas de la simulación en el momento adecuado.

Tabla 4.9. Se calcula el tiempo de ejecución esperado para cada punto

Modelo	Tiempo de ejecución de cada punto
HINDMARSH-ROSE	$1/280000 = 3,6$ milisegundos
IZHIKEVICH	$1/1000000=1$ milisegundo

Elegidos el método matemático a utilizar y el paso de integración que hace que cada modelo se ejecute en un tiempo inferior, pero muy cercano al segundo, queda por realizar las paradas del sistema para ajustar este tiempo de ejecución a un valor más exacto.

Para realizar las paradas de las simulaciones se han planteado dos posibles métodos:

- **MÉTODO DE PARADA 1**

El número total de puntos tiene que ocurrir en un segundo aproximadamente.

En el caso del modelo de Hindmarsh-Rose su simulación utilizando el método de integración numérica de Euler, con un paso de integración de 0,001 genera **280.000 puntos**, de los cuales son necesarios **10.000 puntos** de acuerdo a la frecuencia de la tarjeta de adquisición utilizada; para esto se lleva a cabo un muestreo almacenando uno de cada **28** valores generados. Para el modelo de Izhikevich utilizando el mismo método de integración que en el caso anterior, con un paso de integración de 0,0001, se generan **1.000.000 puntos**, de los cuales sólo se almacena un valor cada **100 puntos**.

Cuando un punto ocurre en un tiempo inferior al que le corresponde se produce una llamada a la función que gestiona las paradas del sistema a través del uso de la función **nanosleep()** (véase ANEXO B), realizando una pausa correspondiente al valor de tiempo restante, de forma que la comunicación con la tarjeta cumpla la restricción temporal correspondiente a 10KHz.

Tabla 4.10. Resultados obtenidos de la simulación, utilizando el método de Euler, de los modelos de Hindmarsh-Rose e Izhikevich con el método de parada 1, generándose 10000 puntos en un tiempo de un segundo en tiempo real. Las columnas 2 y 3 de la tabla permiten comparar la diferencia de tiempos antes y después de aplicar las paradas del sistema. La imagen superior contiene la representación del modelo Hindmarsh-Rose con paso de integración fijo de 0,001 y muestreo cada 28 puntos de los valores generados. La imagen inferior representa el modelo Izhikevich con paso de integración fijo de 0,0001 y muestreo cada 100 puntos de los valores generados.

Modelo	Tiempo de simulación sin parada (segundos)	Tiempo de simulación con parada (segundos)	Representación
HINDMARSH - ROSE	0,022116	1,000003	
IZHIKEVICH	0,048387	1,000022	

- **MÉTODO DE PARADA 2**

El segundo método de parada consiste en una modificación del primero. En lugar de realizar una pequeña parada en cada punto según su tiempo de ocurrencia, las paradas se llevan a cabo solamente con los puntos seleccionados en el muestreo para ser almacenados; en el caso del modelo de Hindmarsh-Rose esto ocurre cada 28 valores, en Izhikevich ocurre cada 100 puntos.

Por lo tanto, en lugar de hacer paradas cada punto generado, se obliga al sistema que sólo los puntos a ser almacenados se generen en un tiempo determinado, para ambos modelos estos puntos deben ocurrir cada 1/10000 segundos. Este método evita que el sistema tenga que parar muchas veces en cortos espacios de tiempo.

Tabla 4.11. Resultados obtenidos de la simulación con el método de parada 2, utilizando el método de Euler para la resolución de los modelos de Hindmarsh-Rose e Izhikevich, generándose 10000 puntos en un tiempo de un segundo en tiempo real. Las columnas 2 y 3 de la tabla permiten comparar la diferencia de tiempos antes y después de aplicar las paradas del sistema. La imagen superior contiene la representación del modelo Hindmarsh-Rose con paso de integración fijo de 0,001 y muestreo cada 28 puntos de los valores generados. La imagen inferior representa el modelo Izhikevich con paso de integración fijo de 0,0001 y muestreo cada 100 puntos de los valores generados.

Modelo	Tiempo de simulación sin parada (segundos)	Tiempo de simulación con parada (segundos)	Representación
HINDMARS H- ROSE	0,022116	1,000060	
IZHIKEVICH	0,048387	1,000056	

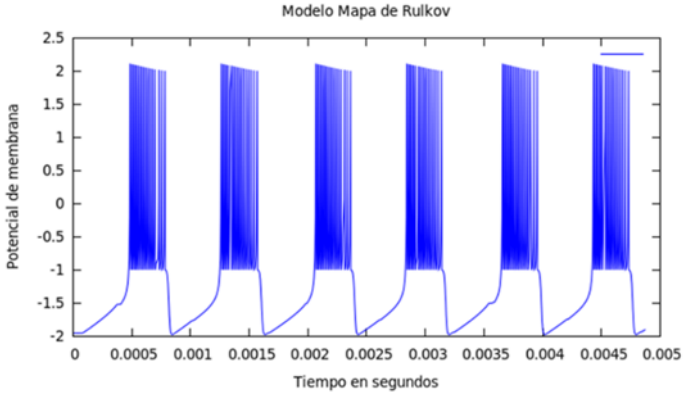
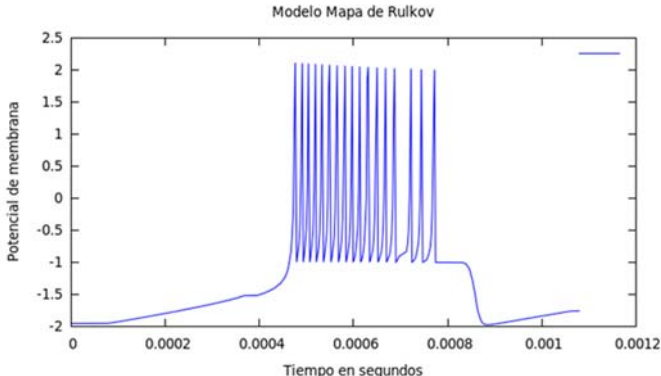
Comparando las tablas 4.11 y 4.12, se observa que el uso del método de parada 1 o el método de parada 2 es indiferente, pues se obtienen resultados similares con ambos ya que el tiempo extra acumulado debido a las paradas producidas es insignificante a causa de trabajar con órdenes de nanosegundos.

4.3.3.CASO ESPECIAL MAPA DE RULKOV

El modelo del Mapa de Rulkov se considera como caso especial pues no necesita el uso de método matemático para su resolución, ya que no contiene ecuaciones diferenciales, por lo que su desarrollo es directo y por ello tampoco necesita un paso de integración (véase **¡Error! No se encuentra el origen de la referencia.**).

Para que este modelo genere una ráfaga completa sólo necesita generar **400 valores** como puede observarse en la tabla 4.12.

Tabla 4.12. Representaciones del Mapa de Rulkov. Para la simulación de 2000 puntos se obtiene una actividad de 6 ráfagas y para 400 puntos se representa una sola ráfaga.

Total de Puntos generados	Representación
2000	 <p>Modelo Mapa de Rulkov</p> <p>Este gráfico muestra el potencial de membrana (Y-axis, -2 a 2.5) frente al tiempo en segundos (X-axis, 0 a 0.005). Se observan seis ráfagas de actividad eléctrica, cada una compuesta por un grupo de oscilaciones de alta frecuencia que alcanzan un potencial de aproximadamente 2.0 V, seguidas por un período de silencio donde el potencial cae a unos -1.5 V.</p>
400	 <p>Modelo Mapa de Rulkov</p> <p>Este gráfico muestra el potencial de membrana (Y-axis, -2 a 2.5) frente al tiempo en segundos (X-axis, 0 a 0.0012). Se observa una única ráfaga de actividad eléctrica, caracterizada por un grupo de oscilaciones de alta frecuencia que alcanzan un potencial de aproximadamente 2.0 V, precedido por un período de silencio y seguido por un período de silencio posterior.</p>

Después de demostrar que este modelo tiene actividad en ráfagas, es necesario resolver el problema del número de puntos que se necesitan generar, pues es necesario que los modelos cumplan con las restricciones impuestas por la neurona viva y por la frecuencia de 10 kHz a la que se ha decidido que trabaje la tarjeta de adquisición de datos. En la simulación de una ráfaga se ha podido observar que se generan 400 puntos, y no existe un paso de integración para variar y aumentar el número de puntos generados. Para solucionar este inconveniente es necesario recurrir a una interpolación lineal, calculándose $(10.000-400)/400 = 24$ puntos entre cada par de puntos iniciales con el fin de obtener el total de 10.000 valores de potencial de membrana.

A continuación puede observarse las líneas de código desarrolladas para realizar la interpolación lineal de acuerdo al polinomio de interpolación de grado 1 utilizado:

$$x_{interp} = x1 + \frac{x2 - x1}{t2 - t1} (t_{interp} - t1) \quad (4.60)$$

```
for ( j=0; j< (10000-TIEMPO)/TIEMPO ; j++ )
{
    gettimeofday(&fin,NULL);
    tin=((fin.tv_sec-inicio.tv_sec)*1e3+(fin.tv_usec-inicio.tv_usec)/1e3);//milisegundos
    xin=x+((x1-x)/(t1-t))*(tin-t);
    fprintf(*f, "%f %f \n", tin/1e3,xin);//segundos
}
}
```

Después de aplicar la interpolación lineal se comprueba que se genera una ráfaga completa de 10.000 valores en un tiempo muy aproximado al segundo, lo cual es el objetivo buscado, ver figura 4.3.

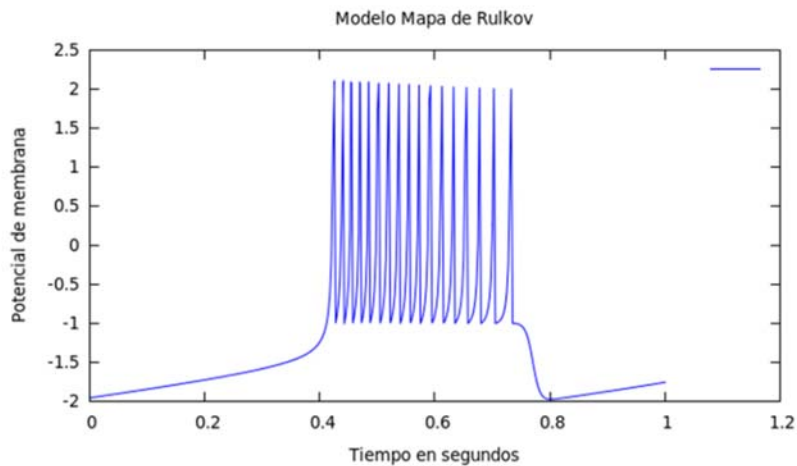


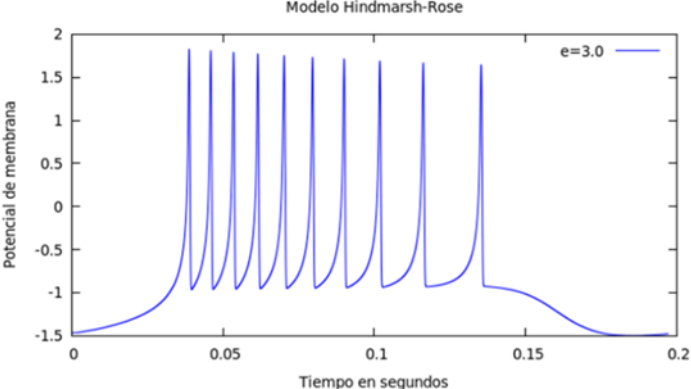
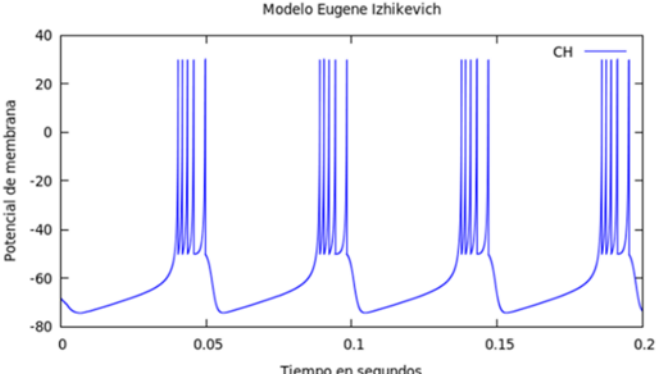
Figura 4.3. Resultados obtenidos de la simulación del Mapa de Rulkov con el método de interpolación lineal y muestreo de puntos para generar 10.000 puntos en un tiempo de un segundo

4.4.COMPARACIÓN DE LAS SIMULACIONES EN DIFERENTES EQUIPOS

Con el objetivo de comprobar que las implementaciones de los modelos neuronales funcionan efectivamente en distintos equipos, a continuación se muestran los resultados obtenidos para los dos PC cuyas características pueden observarse en el capítulo 3 Equipamiento.

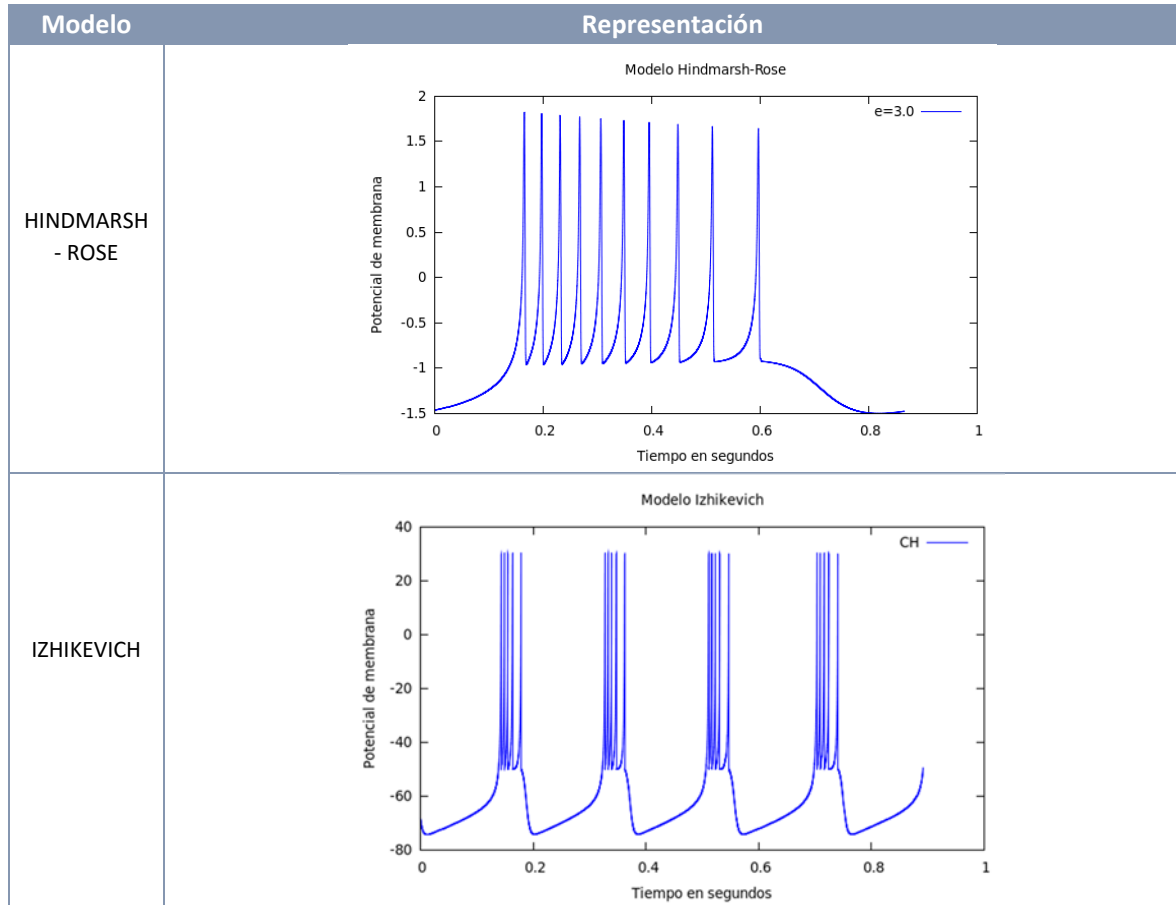
Las simulaciones iniciales sin el ajuste de paradas de tiempo para el PC de desarrollo inicial del apartado 3.1.1 para el modelo de Hindmarsh-Rose y el modelo de Izhikevich son:

Tabla 4.13. Representación del modelo de Hindmarsh-Rose y el modelo de Izhikevich con un paso de integración de 0,001 para ambos modelos, y un límite de 280.000 puntos generados en el PC para el desarrollo inicial

Modelo	Representación
HINDMARSH-ROSE	
IZHIKEVICH	

A continuación puede observarse los resultados obtenidos para la misma implementación de los modelos anteriores, pero utilizando el PC del laboratorio del GNB (véase apartado 3.1.2).

Tabla 4.14. Representación del modelo de Hindmarsh-Rose y el modelo de Izhikevich con un paso de integración fijo de 0,001 para ambos modelos, y un límite de 280000 puntos generados en el PC del laboratorio del GNB

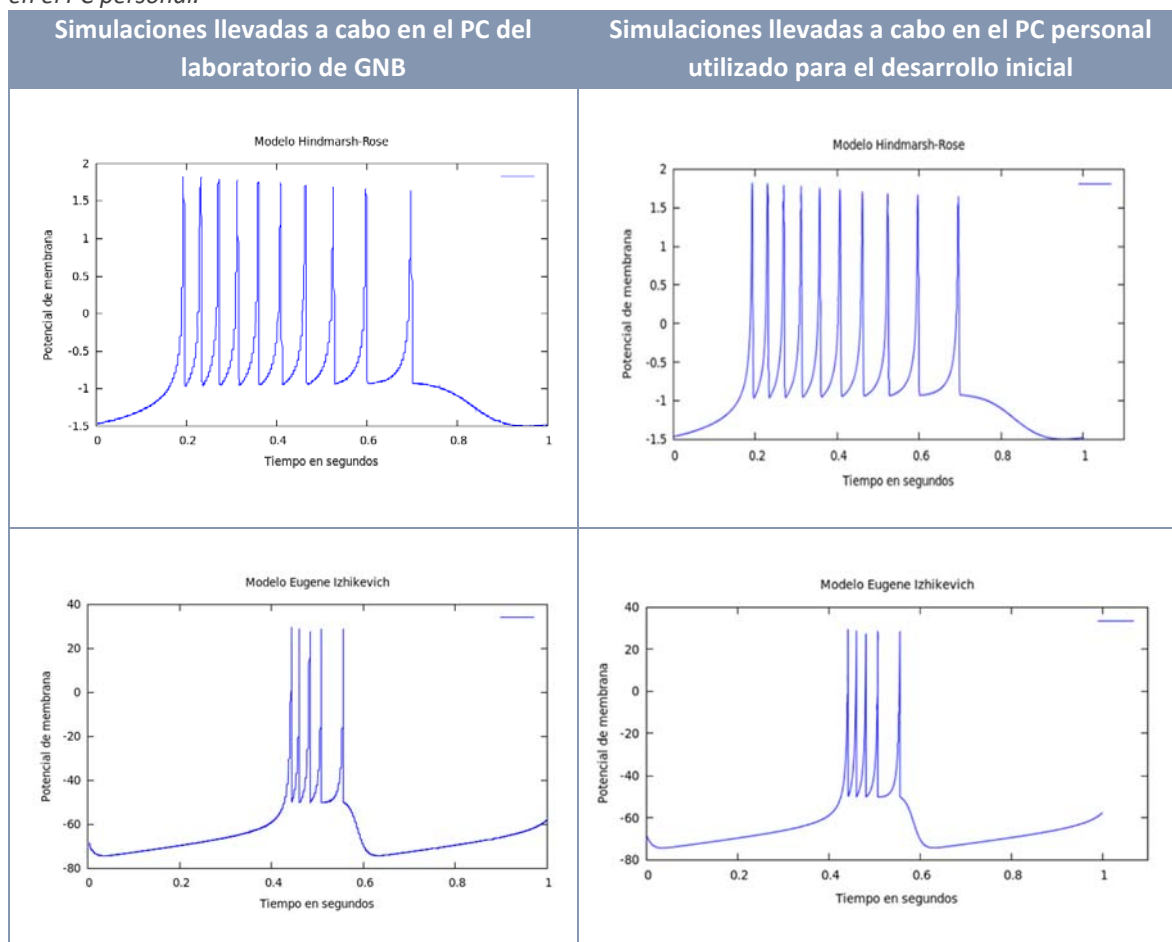


Con las representaciones de las Tablas 4.14 y 4.15 se comprueba que el comportamiento de los modelos varía según el equipo utilizado. También se deduce que el PC del laboratorio del GNB es más lento que el PC utilizado inicialmente en el desarrollo del proyecto.

Una de las principales restricciones planteadas en el desarrollo de los modelos era limitar cada ejecución en un tiempo determinado para que se pudieran llevar a cabo en diferentes equipos (véase apartado 3.3.1) y para su posterior conexión con una neurona viva. El valor de tiempo elegido fue el de un segundo, por lo que ya se ha comprobado que hasta el momento se pueden generar 10000 puntos dentro del límite de tiempo en el PC de desarrollo inicial.

Se puede comprobar en la siguiente tabla que dicha restricción de tiempo generada gracias a las paradas del sistema permite simular los modelos en el PC del laboratorio GNB obteniéndose resultados similares y aceptables.

Tabla 4.15. Comparación de los resultados obtenidos utilizando el PC del laboratorio del GNB y el PC personal de desarrollo inicial. La columna de la izquierda contiene las simulaciones llevadas a cabo en el PC del laboratorio del GNB. Las imágenes de la columna de la derecha corresponden a las simulaciones iniciales en el PC personal.



4.5. CONCLUSIONES

En este capítulo se ha llevado a cabo la implementación de los modelos de neuronas, presentados al principio del mismo, mediante el uso de métodos matemáticos (Euler y Runge-Kutta de cuarto orden) con un paso de integración fijo. Se puede observar los resultados de las simulaciones de los distintos modelos que ha permitido elegir sólo aquellos modelos neuronales que presentan un comportamiento en ráfagas adecuado para la implementación de circuitos híbridos en CPGs (Modelo Hindmarsh-Rose, modelo Izhikevich y el Mapa de Rulkov).

Estos modelos se han implementado con el objetivo de que puedan ser simulados produciendo una ráfaga de 10.000 puntos cada segundo. Al ser modelos que producían muestras muy rápido (10.000 puntos en tiempos inferiores a un segundo), se ha recurrido a adaptar el tiempo de ejecución de las simulaciones de los modelos mediante el uso de la función **nanosleep()**.

En el caso del Mapa de Rulkov, ha sido necesario utilizar interpolación lineal para aumentar el número de puntos generados por ráfaga. Se han realizado las simulaciones con los modelos aislados, sin conexión a la neurona electrónica o la neurona viva, y se ha validado sus simulaciones en distintos PCs.

5. INTEGRACIÓN, PRUEBAS Y RESULTADOS

En el capítulo anterior se implementaron los modelos de neuronas que presentaban un potencial de membrana en ráfagas utilizando lenguaje C, ver figura 5.1.

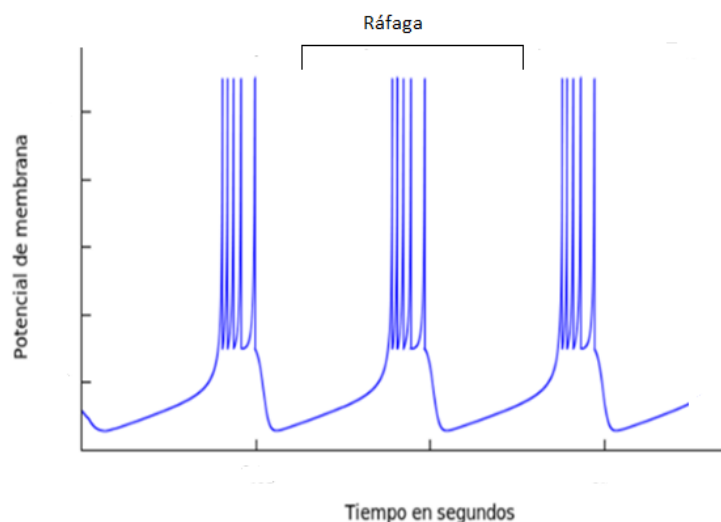


Figura 5.1. La imagen muestra el comportamiento en ráfagas que deben tener los modelos neuronales implementados en el proyecto

Se llevaron a cabo los ajustes del código necesarios para poder simular los modelos en cualquier equipo. Para ello se limitó el tiempo de ejecución de una ráfaga a un segundo en tiempo real, y a 10.000 valores de acuerdo con la frecuencia de 10kHz de la tarjeta de Adquisición (DAQ).

En este capítulo se realizarán los cambios en el código del modelo de Hindmarsh-Rose, el modelo de Izhikevich y el Mapa de Rulkov, con el fin de llevar a cabo la interacción bidireccional con la neurona electrónica en tiempo real o tiempo online. Para ello se hace necesario la adaptación del código al sistema RTAI y el control de la tarjeta de adquisición, además de la adaptación de los parámetros propios de cada modelo, necesarios para conseguir el acoplamiento entre la neurona software y la neurona electrónica del laboratorio del GNB.

Estos mismos códigos serán utilizados para llevar a cabo la implementación y validación de los circuitos híbridos; para lo cual se realizarán dos tipos de interacciones, la primera entre el modelo Hindmarsh-Rose y la neurona viva, la segunda entre el modelo Izhikevich y la neurona viva. Todas las pruebas llevadas a cabo en este capítulo se realizaron en el PC del laboratorio del GNB (véase apartado 3.1.2).

A continuación se procede a explicar los cambios realizados en el código respecto al capítulo anterior, la adaptación al sistema en tiempo real y al uso de COMEDI, y la configuración final utilizada para conseguir el acoplamiento entre ambas neuronas (neurona software y neurona electrónica en primera instancia, y neurona software y neurona viva en los experimentos finales de validación).

5.1. ADAPTACIÓN DE LOS MODELOS A RTAI Y COMEDI

El PC del laboratorio del GNB tiene instalado el sistema RTAI y las librerías para el uso de los drivers de COMEDI, por lo que su instalación y configuración quedan fuera del alcance de este proyecto.

En el apartado 3.4 del capítulo de Equipamiento se puede encontrar una descripción básica sobre COMEDI, para obtener mayor información se recomienda acceder al siguiente enlace: <http://comedi.org/>, donde se encontrarán varios ejemplos útiles para familiarizarse con su uso. En los modelos de Hindmarsh-Rose e Izhikevich se ha añadido dos nuevas funciones, la primera, *int init_comedi(void)*, contiene las llamadas a las funciones encargadas de inicializar los devices y subdevices de COMEDI:

```
int init_comedi(void)
{
```

- Se utiliza `comedi_open (filename)` para realizar la apertura de un dispositivo Comedi especificado por el nombre **filename**:

```
    dev = comedi_open(filename);
    printf("Comedi device handle: %p.\n", dev);
    if (!dev){
        printf("Unable to open %s.\n", filename);
        return 1;
    }
```

- La función `comedi_find_subdevice_by_type(comedi_t* device, int type, unsigned int start_subdevice)` intenta localizar un subdispositivo perteneciente al dispositivo Comedi, identificado en la función anterior como filename. La identificación comienza por el valor dado al campo `start_subdevice`. La opción `type` especifica los tipos posibles de subdispositivo:

COMEDI_SUBD_AI: Entrada analógica
COMEDI_SUBD_AO: Salida analógica

```
    subdevai = comedi_find_subdevice_by_type(dev, COMEDI_SUBD_AI, 0);
    if (subdevai < 0) {
        comedi_close(dev);
        printf("AI subdevice %d not found.\n", COMEDI_SUBD_AI);
        return 1;
    }
    printf("AI subdevice type: %d\n", COMEDI_SUBD_AI);
    comedi_get_krange(dev, subdevai, 0, AI_RANGE, &krangeai);
    maxdatai = comedi_get_maxdata(dev, subdevai, 0);
```

```

subdevao = comedi_find_subdevice_by_type(dev, COMEDI_SUBD_AO,1);
if (subdevao < 0) {
    comedi_close(dev);
    printf("AO subdevice %d not found.\n", COMEDI_SUBD_AO);
    return 1;
}
printf("AO subdevice type: %d \n", COMEDI_SUBD_AO);
comedi_get_krange(dev, subdevao, 0, AO_RANGE, &krangeao);
maxdatao = comedi_get_maxdata(dev, subdevao, 0);

return 0;
}

```

Otra modificación añadida afecta a la función inicial que contiene el modelo matemático de la neurona a simular. En este caso se añaden las llamadas de inicialización y configuración para el uso del sistema RTAI, lo que permite llevar a cabo pruebas en tiempo real (véase apartado 3.3).

Aprovechando las funcionalidades aportadas por las librerías de RTAI y COMEDI, se elimina la función encargada de llamar a la función **nanosleep()** y se realiza en su lugar una parada del sistema durante 0,0001ms llamando a la función **rt_task_wait_period()** cada 28 valores generados en el caso del modelo de Hindmarsh-Rose, y cada 100 valores en el caso de Izhikevich; es decir, se utiliza la idea planteada para el método de parada 2 explicada en el apartado 4.3.2.4.1. En este caso se ha comprobado que las integraciones se producen en órdenes temporales de nanosegundos pues los modelos elegidos son rápidos. Por lo que se ha optado por utilizar un tiempo de parada del sistema constante, pero se podría volver a aplicar el método que se ha utilizado en las simulaciones del capítulo anterior (antes de aplicar RTAI y COMEDI), calculando los tiempos en los que se generan los valores y deteniendo el sistema el tiempo restante hasta los 0,0001ms.

Se sustituye el uso de la función **gettimeofday()** para medir el tiempo del sistema por la función **rt_get_time()**. Ambas funciones realizan el mismo cálculo de tiempo, por lo que el uso de una función u otra es indiferente, ver figura 5.2.

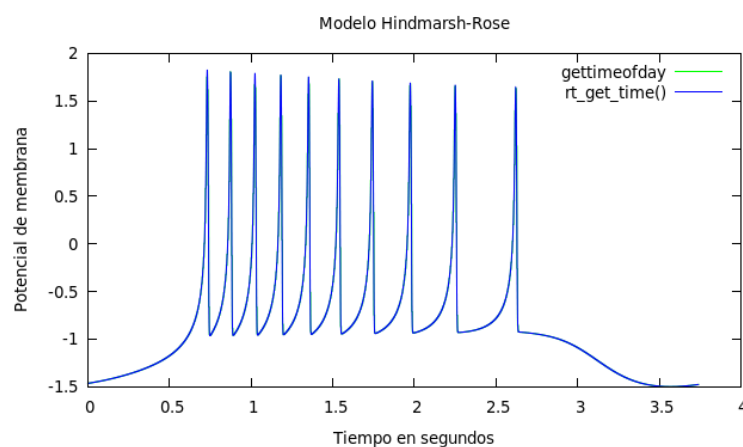


Figura 5.2. Se realiza la simulación del modelo de Hindmarsh-Rose utilizando el método de Euler con un paso de integración fijo de 0,001. La gráfica permite comparar que el uso de la función **Gettimeofday()** o la función **rt_get_time()** no cambia los resultados en el cálculo de tiempo de ejecución del modelo.

5.2.RESULTADOS OBTENIDOS DE LAS SIMULACIONES UTILIZANDO RTAI Y COMEDI

A continuación se pueden observar distintas pruebas del modelo de Hindmarsh-Rose, el cual ha sido tomado como modelo de referencia durante todo el proyecto por su sencilla implementación y por los resultados obtenidos sin requerir demasiados ajustes. En este capítulo se ha adaptado este mismo modelo para el uso con COMEDI y RTAI. **Estas pruebas se llevan a cabo sin conectar el modelo implementado a la neurona electrónica o la neurona viva**, por lo que sólo se hace la transmisión de los valores generados hacia la tarjeta de adquisición, sin recibir datos entrantes.

- **Prueba 1:** Simulación del modelo de Hindmarsh-Rose sin paradas del sistema. Se transmite a la tarjeta de adquisición n datos que permitan representar una ráfaga completa en el modelo de Hindmarsh-Rose. Para realizar una comparación con el modelo desarrollado en el capítulo anterior, se han generado 280.000 valores de potencial de membrana, ver figura 5.3.

Los resultados obtenidos permiten comprobar que el comportamiento en ráfaga se mantiene. Comparando la representación obtenida en este apartado con la obtenida anteriormente en el apartado 4.4, en la que también se llevaron a cabo las simulaciones del modelo de Hindmarsh-Rose en el PC del laboratorio del GNB, se observa que el uso de un sistema con RTAI disminuye los tiempos de simulación pues se da prioridad a esta tarea sobre las demás, realizándose en un tiempo real estricto, ver figura 5.3.

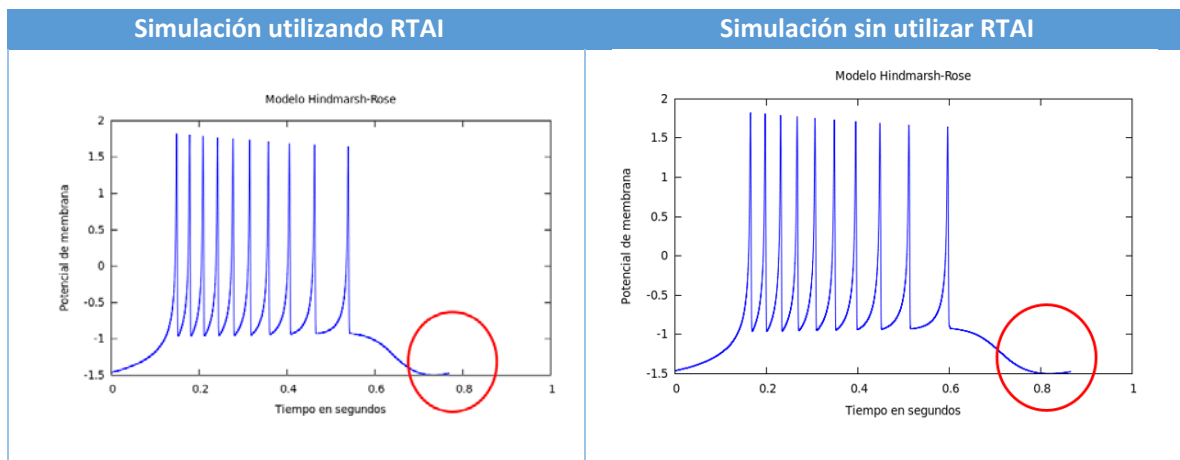


Figura 5.3. Comparación de los resultados entre la simulación de código implementado para su ejecución en sistemas RTAI, gráfica izquierda, con los resultados obtenidos de la simulación en un sistema GPOS, gráfica derecha.

- **Prueba 2:** Simulación del modelo de Hindmarsh-Rose, llevando a cabo paradas en el sistema a partir del uso de la función `rt_task_wait_period()`, transmitiéndose 10.000 puntos en un segundo a la tarjeta de adquisición, ver figura 5.4.

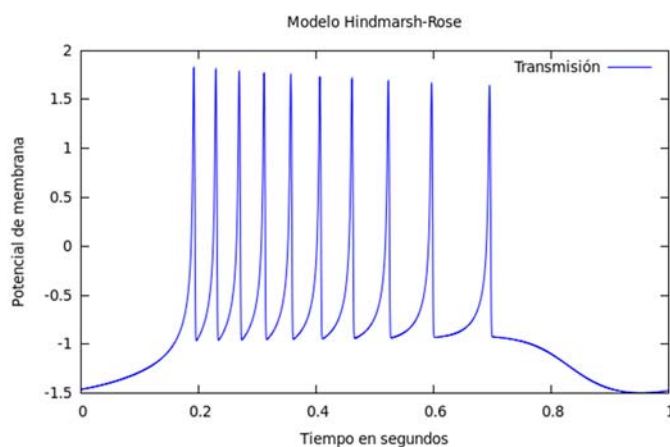


Figura 5.4. Resultado de la simulación del modelo Hindmarsh-Rose de una ráfaga de 10.000 puntos en un tiempo determinado de un segundo

Una vez que se ha comprobado que el uso de RTAI y de COMEDI no ha alterado los resultados esperados, sino que los ha mejorado, se ha ajustado nuevamente el código para su simulación en un tiempo determinado. Tal y como puede recordarse, el tiempo elegido para producir una ráfaga de 10.000 puntos era un segundo. Aunque el código de los modelos implementados en el capítulo anterior ha sido modificado, se acaba de comprobar que el comportamiento de dichas implementaciones se mantiene correctamente, obteniéndose los mismos resultados que se habían obtenido con los anteriores códigos, pero con la ventaja del uso de un sistema de RTAI que garantiza la prioridad de ejecución del modelo simulado sobre cualquier otra tarea pendiente en el sistema en el momento de la ejecución, lo que permite la comunicación bidireccional en tiempo real con una neurona externa (neurona electrónica o neurona viva). También se ha validado el comportamiento del modelo de Izhikevich y del Mapa de Rulkov, obteniéndose nuevamente buenos resultados, por lo que también se puede trabajar con ellos.

5.3. INTERACCIÓN BIDIRECCIONAL ENTRE LA NEURONA SOFTWARE Y LA NEURONA ELECTRÓNICA. COMUNICANDO NEURONAS

Uno de los objetivos del proyecto era llevar a cabo la comunicación entre la neurona implementada en software y la neurona electrónica mediante una **sinapsis eléctrica artificial**. Todos los pasos desarrollados en los capítulos y apartados anteriores han permitido llegar a este punto.

Se han elegido tres modelos de neuronas, **Hindmarsh-Rose**, **Izhikevich** y el **Mapa de Rulkov** por su comportamiento en ráfagas y su sencilla implementación en lenguaje C, lo que ha permitido realizar distintas pruebas dentro del tiempo asignado para el desarrollo del proyecto.

Se han utilizado las prestaciones que aporta el uso de COMEDI y RTAI para la simulación en tiempo real de forma estricta; además de métodos de optimización, tanto en el código como con el uso de los parámetros.

Desde este punto todo está preparado para llevar a cabo la comunicación con la neurona electrónica del laboratorio del GNB, ver figura 5.5.

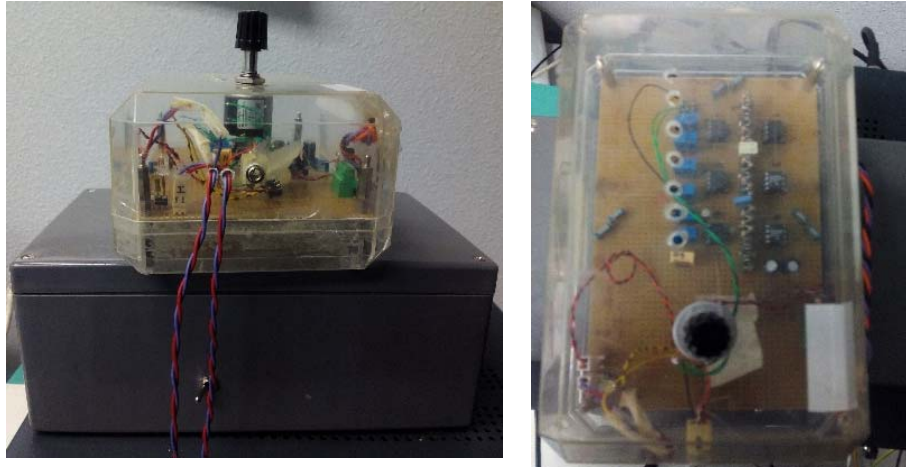


Figura 5.5. Neurona electrónica del laboratorio del GNB desarrollada a partir del modelo matemático de neurona de Hindmarsh-Rose

5.3.1. MODELO MATEMÁTICO PARA LA COMUNICACIÓN ENTRE NEURONAS REPRESENTADAS POR EL MODELO HINDMARSH-ROSE

A partir del modelo de ecuaciones de Hindmarsh-Rose, presentado en el apartado 4.1.4, se puede establecer una comunicación entre la neurona implementada en software y la neurona electrónica; es decir, se puede llevar a cabo una **sinapsis eléctrica artificial**. El nuevo sistema de ecuaciones en el modelo (en la neurona software) queda descrito por:

$$\frac{dx_1(t)}{dt} = y_1(t) + 3x_1^2(t) - x_1^3(t) - z_1(t) + e_1 + g_{12}(x_2(t) - x_1(t)) \quad (5.1)$$

$$\frac{dy_1(t)}{dt} = 1 - 5x_1^2(t) - y_1(t) \quad (5.2)$$

$$\frac{dz_1(t)}{dt} = \mu(-z_1(t) + S(x_1(t) + 1.6)) \quad (5.3)$$

- El índice 1 representa a la neurona software y el índice 2 a la neurona electrónica.
- $g_{12}(x_2(t) - x_1(t))$ representa el valor total de la corriente que llega a la neurona software denotada por el índice 1.
- g_{12} representa el acoplamiento eléctrico entre la neurona software (1) y la neurona electrónica (2).
- $(x_2(t) - x_1(t))$ Representa la diferencia entre el potencial generado por la neurona implementada en software (1) y el potencial recibido desde la neurona electrónica (2).

Para llevar a cabo la comunicación entre las dos neuronas se transmite hacia la neurona electrónica el valor $(g_{21}(x_1(t) - x_2(t)))$, siendo $x_2(t)$ el valor que devuelve dicha neurona a la neurona software; es decir, desde la neurona electrónica recibe una corriente de valor $(g_{12}(x_2(t) - x_1(t)))$.

El valor de acoplamiento electrónico g_{12} se ha elegido después de realizar varias pruebas entre aquellos que presentaban el mejor acoplamiento entre las neuronas.

5.3.1.1. PASOS LLEVADOS A CABO PARA REALIZAR LA COMUNICACIÓN ENTRE LA NEURONA SOFTWARE Y LA NEURONA ELECTRÓNICA

En resumen, se tienen dos neuronas representadas ambas con el modelo de Hindmarsh-Rose, la neurona desarrollada en software (NS) y la neurona desarrollada en hardware o neurona electrónica (NE). La neurona electrónica produce actividad a la misma velocidad que una neurona viva y un experimento bidireccional con esta neurona supone una primera validación en la que no se requiere realizar una preparación con un ser vivo.

Iniciada la comunicación entre ambas neuronas, la neurona electrónica envía el valor de su potencial de membrana, el cual permite al código calcular el valor de corriente del acoplamiento eléctrico y el valor del potencial de membrana que en ese momento la neurona software ha generado. A la neurona electrónica se le enviará un valor de corriente sináptico calculado desde el código de la neurona software.

A continuación se pueden observar los diagramas que intentan explicar el proceso llevado a cabo para realizar la comunicación y correcto acoplamiento entre las dos neuronas elegidas en el desarrollo de este proyecto:

1. Se registra la actividad de las dos neuronas sin utilizar un valor de acoplamiento, por lo tanto, en ambas neuronas se registra el valor de potencial de membrana generado en ese momento de muestreo:
 - a. La neurona software inicializa la simulación del modelo elegido, en este caso será Hindmarsh-Rose, generando valores de potencial de membrana, $x_1(t)$.
 - b. Realiza el muestreo, enviando 1 de cada n valores (véase apartado 4.3). Este valor se obtiene de la necesidad de muestrear sólo 10.000 valores, impuesto por la frecuencia elegida para utilizar la tarjeta de adquisición (10kHz) y por las restricciones de velocidad de la neurona electrónica, por lo que se ha elegido 280.000 puntos para representar una ráfaga en el modelo de Hindmarsh-Rose, muestreándose 1 de cada 28 puntos integrados, y 1.000.000 para su representación en el modelo de Izhikevich, muestreándose en este caso 1 de cada 100 valores. Se recibe dentro de la etapa de muestreo el valor enviado desde la neurona electrónica, $x_2(t)$.

- Se añade el uso de un acoplamiento eléctrico, g_{21} en el sentido de la neurona software (1) hacia la neurona electrónica (2) y g_{12} para la comunicación en el sentido contrario. Por lo tanto, en el momento en el que el modelo implementado en software genera un valor de potencial de membrana, **se añade** el valor de la corriente calculada por $g_{12}(x_2(t) - x_1(t))$, siendo g_{12} el valor de acoplamiento eléctrico elegido, $x_1(t)$ el valor de potencial de membrana de la neurona software y $x_2(t)$ el valor de potencial de membrana recibido desde la neurona electrónica.

Con el nuevo valor de potencial de membrana calculado utilizando el valor de acoplamiento, se calcula un nuevo valor de corriente para enviar a la corriente eléctrica.

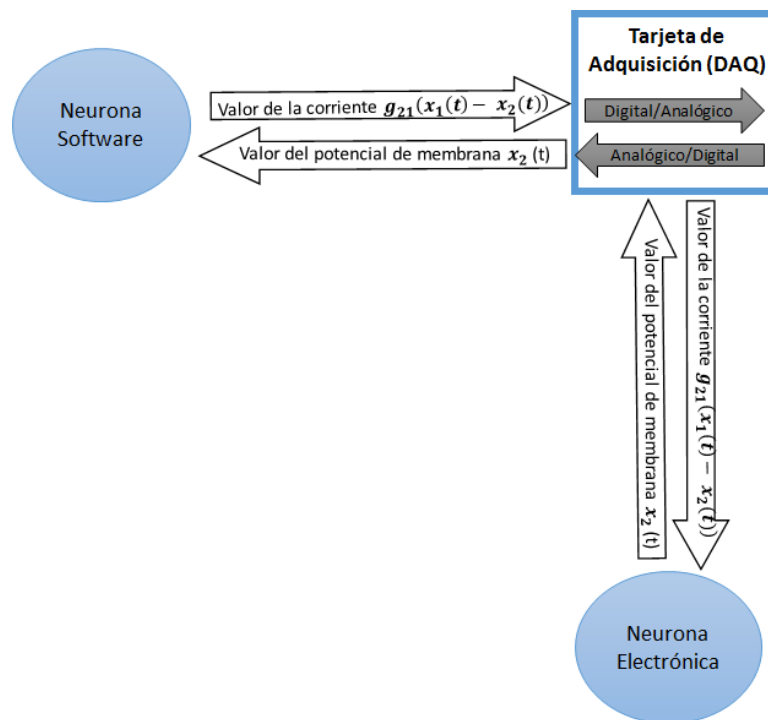


Figura 5.6. Diagrama que representa el circuito entre la neurona software y la neurona electrónica, conectadas a través de la tarjeta de adquisición de datos del laboratorio del GNB. Se añade al sistema el valor de una corriente, por lo tanto se genera una sinapsis eléctrica artificial entre ambas neuronas.

5.3.2. RESULTADOS OBTENIDOS DE LA INTERACCIÓN BIDIRECCIONAL UTILIZANDO EL MODELO DE HINDMARS-ROSE

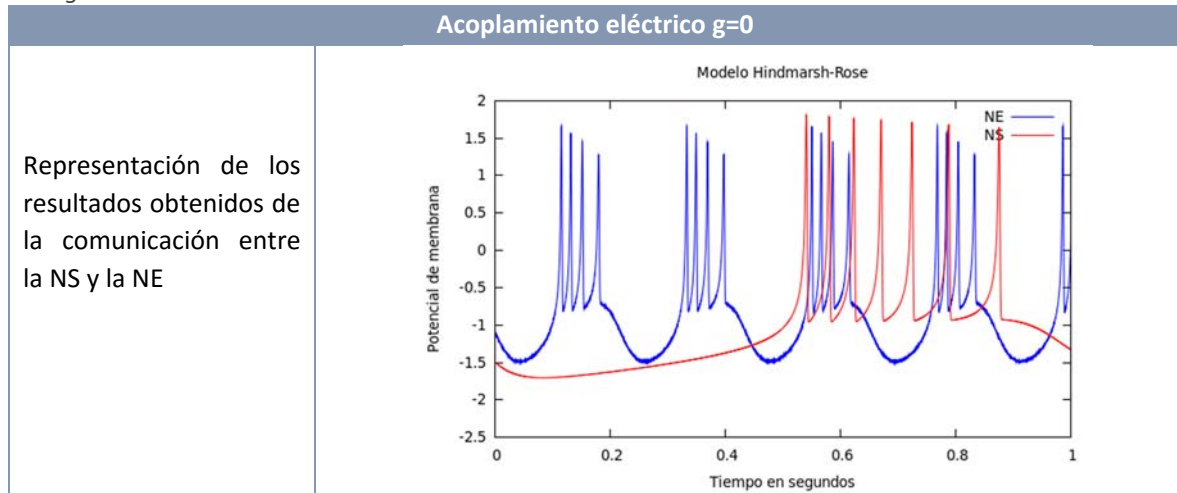
En este apartado se pueden observar los resultados obtenidos en el proceso de comunicación entre la neurona software y la neurona electrónica, expuesto en el apartado anterior.

Se han llevado a cabo varias pruebas cambiando el valor del acoplamiento electrónico. Para obtener buenos resultados y poder comparar en la misma escala las representaciones de los potenciales de membrana de ambas neuronas ha sido necesario disminuir la amplitud de la corriente enviada a la neurona electrónica, dividiendo dicho valor entre un valor **Ampli** adecuado,

obtenido después de realizar varias simulaciones: $\frac{g(x_s(t) - x_E(t))}{Ampli}$

Eligiéndose el valor para la variable designada como *Ampli* se ha comprobado que con su uso se obtiene el mejor resultado de acoplamiento en las representaciones. A continuación pueden observarse los distintos resultados obtenidos:

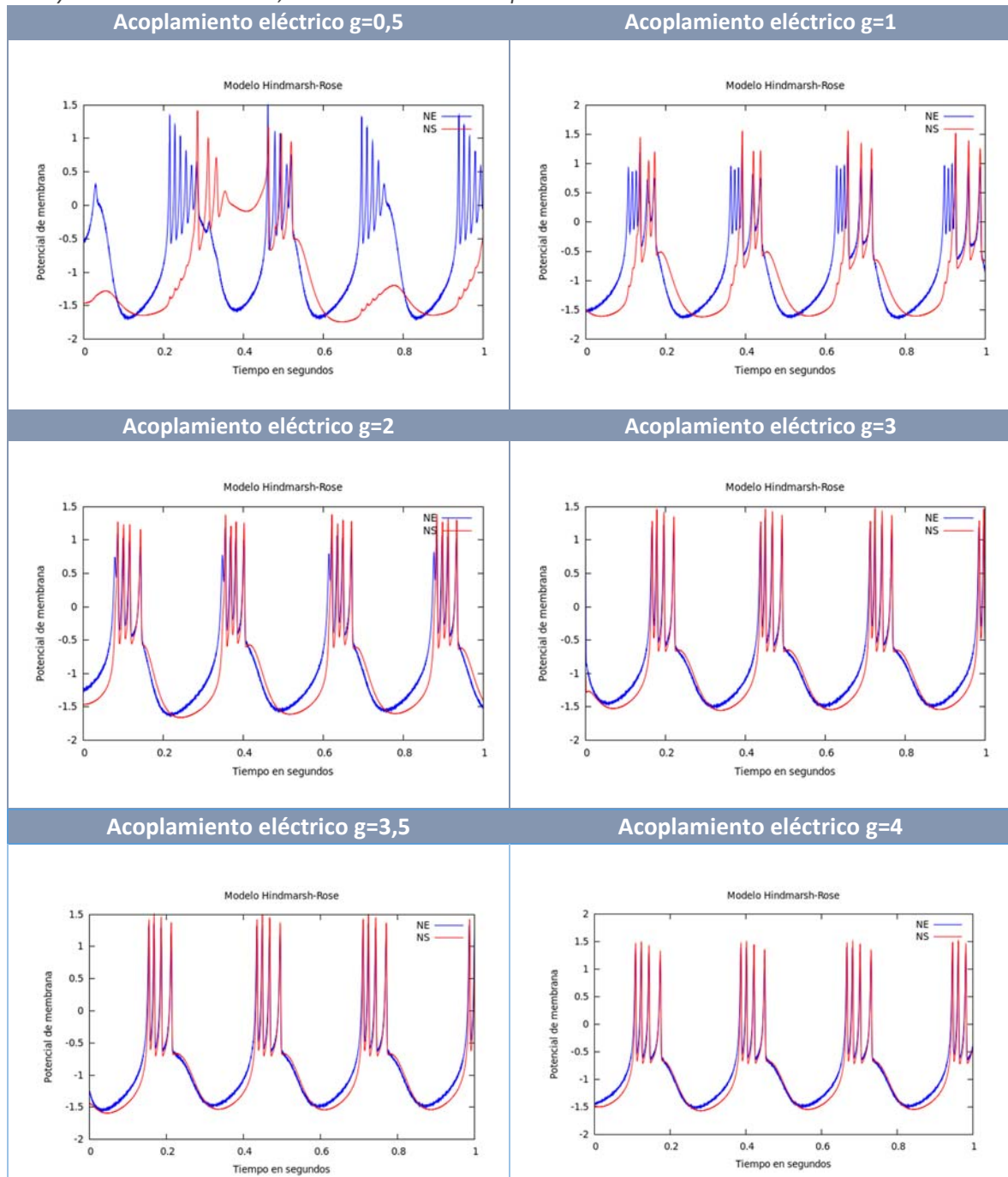
Tabla 5.1. Circuito híbrido con el modelo Hindmarsh-Rose sin ajustes de amplitud y acoplamiento $g=0$; es decir, sin utilizar valor de corriente. La imagen azul corresponde al valor del potencial que genera la neurona electrónica, la imagen roja al potencial generado por la neurona software. Las imágenes se representan en un segundo.

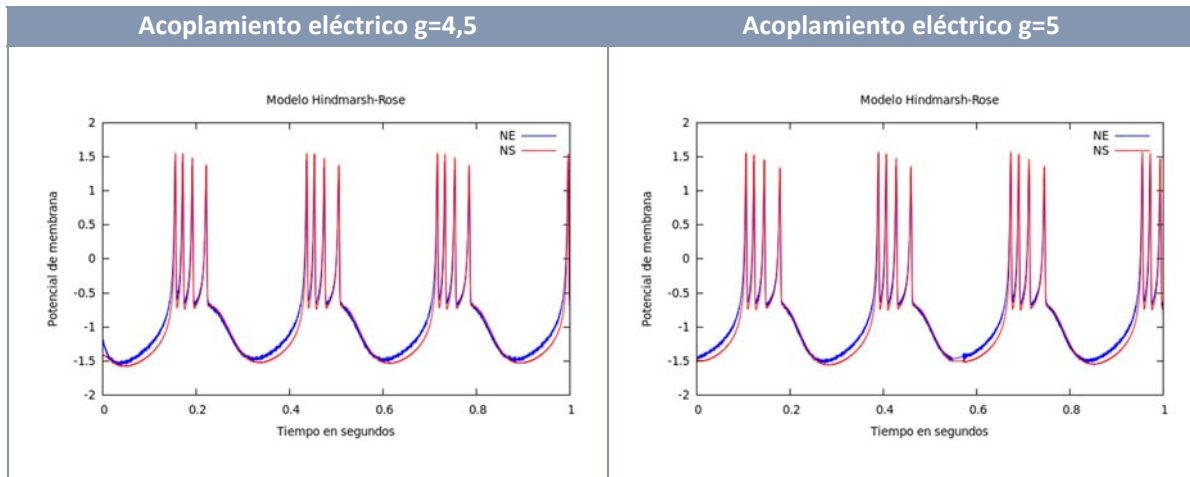


Para valores de acoplamiento, g , distintos de cero se intercambia corriente a través de la sinapsis eléctrica entre la neurona software y la neurona electrónica. Ha sido necesario realizar varias simulaciones con distintos valores de acoplamiento g para obtener el valor que permite sincronizar ambas neuronas.

En las siguientes tablas se puede observar las distintas simulaciones llevadas a cabo con diferentes valores de acoplamiento. La primera fila corresponde al potencial de membrana que genera la neurona electrónica, la segunda fila al potencial de la neurona software, a ambas neuronas se les está aplicando el valor de una corriente con el objetivo de sincronizar sus comportamientos. La última fila muestra la superposición de las dos filas anteriores, donde se puede observar el nivel de sincronismo que se obtiene en cada momento para el valor g utilizado. **Se ha utilizado el mismo valor de g (conductancia) para la corriente enviada desde la neurona software hacia la neurona electrónica y para la corriente enviada en el sentido opuesto.**

Tabla 5.2. Resultados obtenidos para el circuito formado por la neurona software con el modelo Hindmarsh-Rose y la neurona electrónica, variando el valor de acoplamiento





Después de las simulaciones anteriores, se observa que se obtiene el mejor acoplamiento para los valores de $g=3, 4, \text{ y } 4.5$. Para valores superiores las señales se ven afectadas por el ruido. Por lo tanto, se elige $g=3$ para el valor de acoplamiento eléctrico entre la neurona software y la neurona electrónica del modelo Hindmarsh-Rose, este valor se aplica para la conductancia en ambas direcciones, desde la neurona software hacia la neurona electrónica y viceversa, ver figura 5.7.

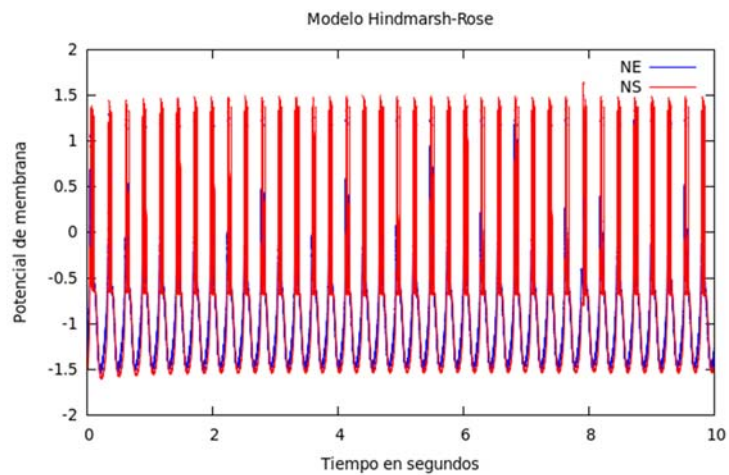


Figura 5.7. Representación de la sincronización resultado de la comunicación bidireccional entre la neurona implementada en software (gráfica roja) y la neurona electrónica (gráfica azul) durante 10 segundos utilizando el mismo valor de conductancia, $g=3$, en ambas direcciones de la comunicación.

5.3.3.INTERACCIÓN BIDIRECCIONAL UTILIZANDO EL MODELO DE IZHIKEVICH

Anteriormente se ha comprobado que se puede establecer una comunicación bidireccional entre dos neuronas que presentan el mismo modelo matemático. El otro modelo implementado en este capítulo para su simulación en sistemas RTAI es el modelo de Izhikevich.

El acoplamiento se ha llevado a cabo de la misma manera que en el caso de Hindmarsh-Rose, restando el valor de la corriente total al del potencial de membrana calculado; por lo tanto, las explicaciones del apartado 5.3.1 y del subapartado 5.3.1.1 también se aplican para este modelo, sólo es necesario sustituir las ecuaciones del modelo anterior por el nuevo a simular. A continuación se muestra el conjunto de **ecuaciones utilizadas para la implementación de la neurona software** y así poder realizar la misma comunicación que se llevó a cabo en el apartado anterior.

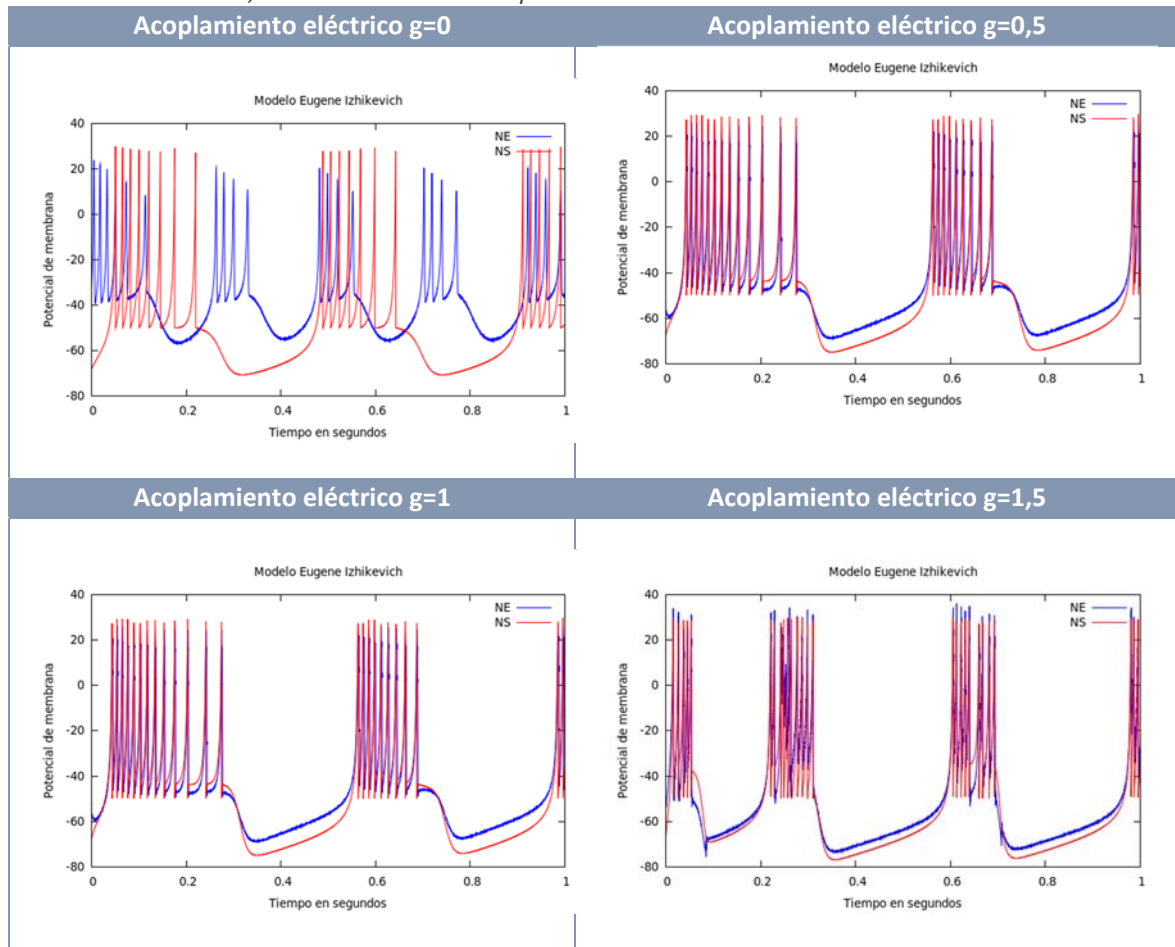
$$\frac{dv(t)}{dt} = 0.04v^2 + 5v + 140 - u + I + g_{12}(v_2(t) - v_1(t)) \quad (5.4)$$

$$\frac{du(t)}{dt} = a(bv - u) \quad (5.5)$$

De la misma manera, se ha tenido que realizar varias simulaciones para buscar el mejor valor de acoplamiento eléctrico. A diferencia del modelo de Hindmarsh-Rose en el que fue necesario disminuir 4 veces la amplitud de la corriente enviada a la neurona eléctrica, en el modelo Izhikevich se reduce la amplitud 6 veces y se desplazada hacia valores negativos, con objeto de poder comparar las representaciones de los potenciales de membrana de las neuronas que intervienen en la comunicación: $\left(\frac{g(x_s(t)-x_E(t))}{Ampli}\right) - offset$

Nuevamente, las siguientes tablas albergan las distintas simulaciones llevadas a cabo con diferentes valores de acoplamiento en el circuito formado por la neurona software implementada con el modelo de Izhikevich, y la neurona electrónica. Cada imagen muestra los valores superpuestos de los potenciales de membrana de ambas neuronas, en ellas se puede observar el cambio de comportamiento que se produce al variar el valor de la corriente utilizada. También se observa el valor de g que produce una mejor sincronización. Nuevamente, se ha utilizado el mismo valor para la conductancia en ambas direcciones, desde la neurona software hacia la neurona electrónica y viceversa.

Tabla 5.3. Resultados obtenidos para el circuito formado por la neurona software con el modelo Izhikevich y la neurona electrónica, variando el valor de acoplamiento



Queda comprobado que es posible establecerse una comunicación bidireccional entre neuronas que presentan diferente modelo de desarrollo matemático. Se ha conseguido acoplar la neurona software del modelo de Izhikevich a la neurona electrónica del modelo Hindmarsh-Rose utilizando el valor de acoplamiento eléctrico $g=1$, aplicando el mismo valor en ambas direcciones, y un ajuste de amplitud de la corriente enviada a la neurona eléctrica, ver figura 5.8.

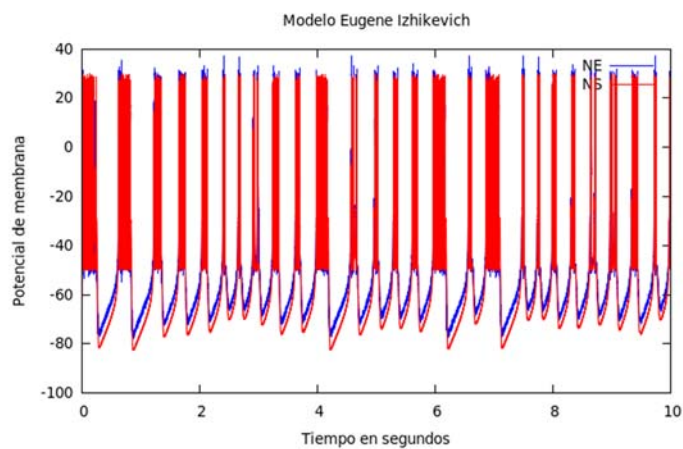


Figura 5.8. Representación del acoplamiento resultado de la comunicación bidireccional entre la neurona implementada en software (gráfica roja) y la neurona electrónica (gráfica azul) durante 10 segundos con $g=1$, se utiliza el mismo valor de conductancia en ambas direcciones.

5.3.4. INTERACCIÓN BIDIRECCIONAL UTILIZANDO EL MAPA DE RULKOV

Para la validación de las pruebas de comunicación bidireccional entre la neurona software y la neurona electrónica también se ha adaptado el código del Mapa de Rulkov (véase apartado 4.1.5):

$$x_{n+1} = f(x_n, y_n + eI_n) + g_{12}(x_2(t) - x_1(t)) \quad (5.6)$$

$$y_{n+1} = y_n - \mu(x_n + 1) + \mu\sigma + \mu\sigma_e I_n \quad (5.7)$$

$$f(x, y) = \begin{cases} \frac{\alpha}{1-x} + y, & x \leq 0 \\ \alpha + y, & 0 \leq x < \alpha + 1 \\ -1, & x > \alpha + y \end{cases} \quad (5.8)$$

En este caso se ha encontrado un único valor de acoplamiento claramente válido para la comunicación, $g=0,5$ y se ha modificado la corriente enviada a la neurona electrónica, reduciendo su amplitud y desplazándola, al igual que se hizo en los modelos anteriores:

$$\frac{g(x_s(t) - x_E(t))}{Ampli} - offset$$

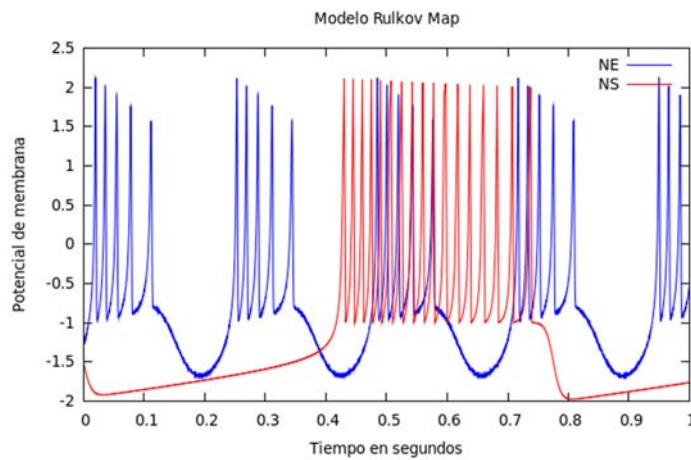
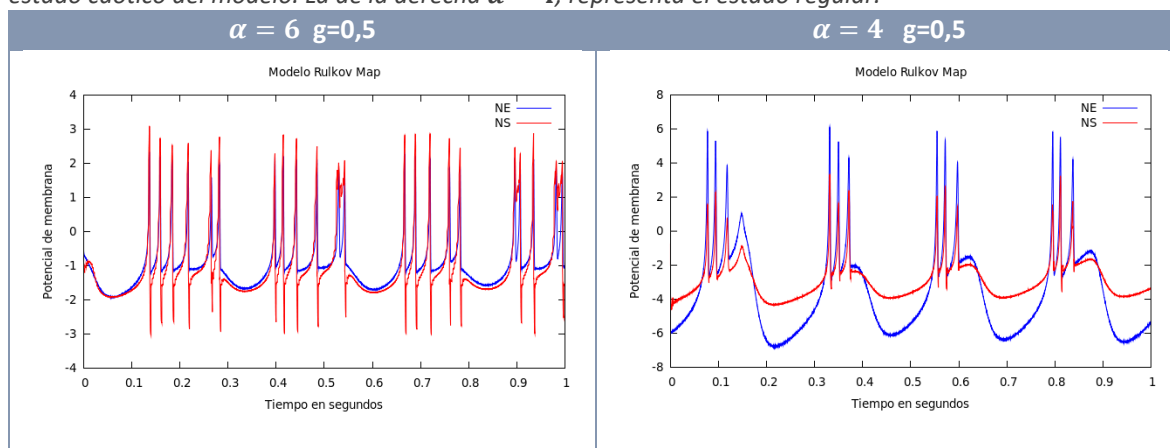


Figura 5.9. Representación inicial del circuito híbrido con el Mapa de Rulkov. En la imagen se observan los resultados con valor de acoplamiento $g=0$. Se han ajustado los valores de *Ampli* y *offset* de la neurona software con objeto de sincronizarla con la neurona electrónica. En rojo se representa el potencial de membrana del modelo y en azul el de la neurona electrónica

Se han llevado a cabo dos pruebas de acoplamiento, la primera con $\alpha = 6$ (estado caótico del modelo) y la otra prueba en el estado regular del modelo, $\alpha = 4$. Los resultados obtenidos de las simulaciones se pueden observar a continuación:

Tabla 5.4. Representación de los potenciales de membrana de la implementación del Mapa de Rulkov (gráfica roja) y la neurona electrónica (gráfica azul), generados en el circuito establecido mediante la sinapsis eléctrica con valor de acoplamiento $g=0,5$. La imagen de la izquierda con $\alpha = 6$ representa el estado caótico del modelo. La de la derecha $\alpha = 4$, representa el estado regular.



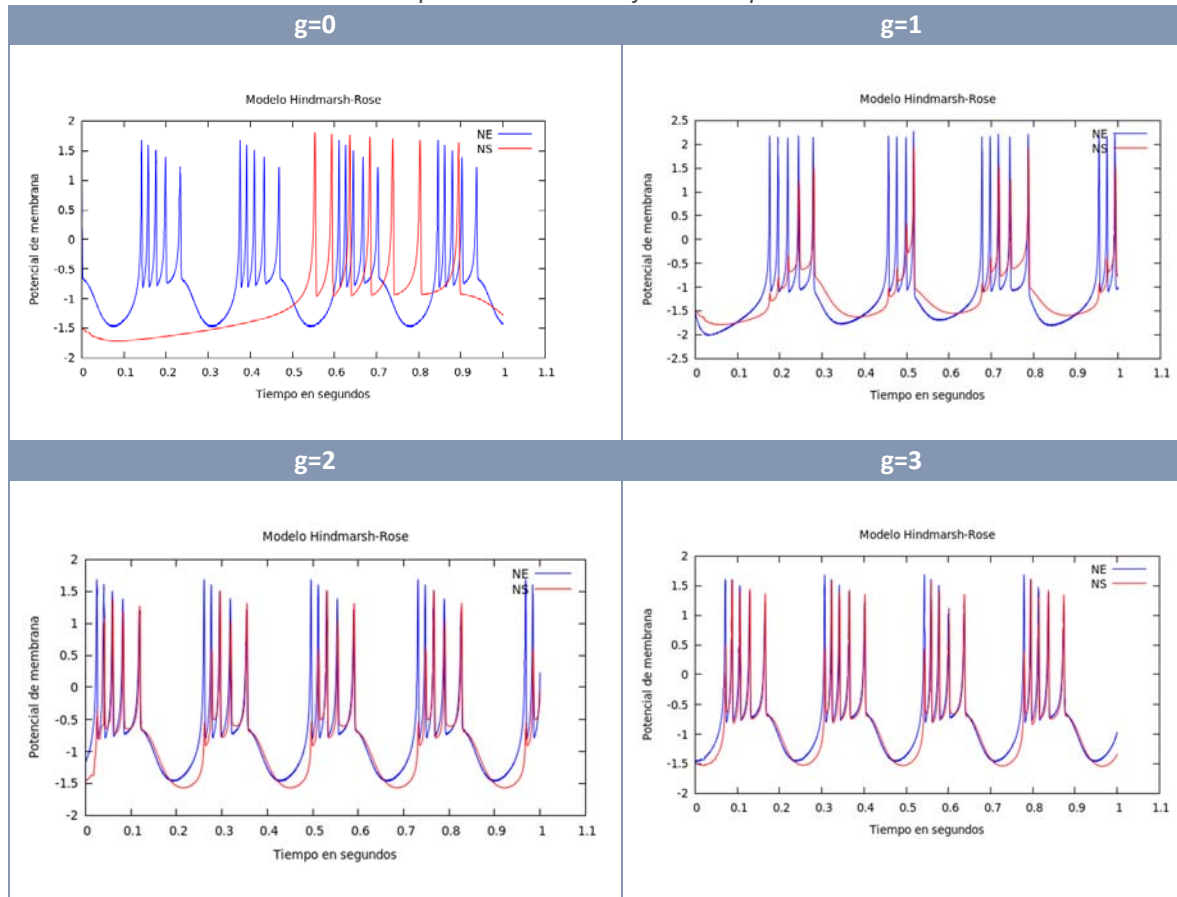
En las imágenes anteriores se puede observar que el comportamiento de ambas neuronas cambia según el valor dado a la variable designada como α , pero el sincronismo entre ambas neuronas se consigue con el mismo valor de acoplamiento para ambos casos. Es decir, también es importante tener en cuenta las variables propias del modelo para intentar llevar a cabo el sincronismo entre ambas neuronas; por ejemplo, en el caso en el que $\alpha = 4$ se observa que no se producen valores similares en la zona de reposo de las representaciones.

5.4.LA COMUNICACIÓN BIDIRECCIONAL SE MANTIENE SIN EL USO DE RTAI

Por último, se ha vuelto a modificar el código para el modelo de Hindmarsh-Rose, eliminando las funciones que hacen uso del sistema RTAI, pero manteniendo la configuración de COMEDI que permite el uso de la tarjeta de adquisición. En este caso no se puede recurrir a las funciones `rt_task_wait_period()` y `rt_get_time()`, pues pertenecen a las librerías de RTAI, por lo que se vuelven a utilizar las funciones `nanosleep()` y `gettimeofday()`, tal y como se hizo en el código implementado en el capítulo 4.

En el apartado 4.4 se comprobó que el código desarrollado es flexible, y se puede realizar su simulación en equipos diferentes al utilizado para su desarrollo. En este apartado se comprueba que las implementaciones siguen funcionando de la manera deseada, aunque no se utilice un sistema en tiempo real como RTAI; permitiendo establecer la comunicación bidireccional con un buen acoplamiento entre la neurona software y la neurona electrónica. Para llevar a cabo las pruebas se ha comunicado nuevamente el modelo de la neurona software con la neurona electrónica del laboratorio del GNB, se generaron 10.000 valores de potencial de membrana en un segundo y se realizaron varias pruebas con valores de conductancia $g=0, 1, 2$ y 3 . El valor de la conductancia utilizado en la comunicación desde la neurona software hacia la neurona electrónica es el mismo valor que se utiliza en el sentido contrario de la comunicación. A continuación se pueden observar los resultados obtenidos de las simulaciones:

Tabla 5.5. Representaciones de los resultados obtenidos de la comunicación bidireccional entre la neurona software y la neurona electrónica para varios valores de conductancia con el objeto de comprobar que las implementaciones llevadas a cabo siguen su correcto funcionamiento sin el uso de las prestaciones del RTAI. Se ha utilizado el mismo valor de conductancia g en ambos sentidos de la comunicación y se mantiene el uso de las librerías de COMEDI necesarias para el uso de la tarjeta de adquisición de datos.



Como se puede observar, se comprueba que utilizando las funciones aplicadas en los desarrollos iniciales del proyecto, en los que no se llevaba a cabo la comunicación bidireccional; es decir, sin el uso de RTAI, se consigue el acoplamiento en la comunicación bidireccional entre la neurona software y la neurona electrónica con una precisión temporal aceptable. Por lo que se deduce que en el caso de no tener acceso al uso de un sistema operativo ampliado con RTAI se puede conseguir la comunicación entre ambas neuronas en tiempo real, pues las implementaciones de los modelos se han llevado a cabo de manera flexible, se ha cumplido con las restricciones impuestas por la neurona real utilizando un modelo de bajo coste computacional, imponiéndose la integración de 10.000 valores (se utiliza la tarjeta de adquisición de datos con una frecuencia de valor 10kHz) dentro de un segundo. En modelos de mayor coste computacional, el uso de un sistema operativo de tiempo real puede resultar necesario.

5.5. INTERACCIÓN BIDIRECCIONAL ENTRE LA NEURONA SOFTWARE Y LA NEURONA VIVA. CIRCUITOS HÍBRIDOS

El último objetivo a cumplir en este proyecto es la implementación de **circuitos híbridos**, creando una comunicación bidireccional, mediante una sinapsis eléctrica artificial, en tiempo real entre el modelo de neurona software y una neurona viva del sistema nervioso del *Carcinus maenas* (Véase apartado 3.6).

A partir del código desarrollado a lo largo de todo el proyecto, desde las primeras pruebas implementadas en el PC personal hasta las modificaciones del código para su adaptación al uso de RTAI y COMEDI, se ha comprobado que se producen simulaciones en tiempo real, que se ejecuta una ráfaga de 10.000 puntos cada segundo, y que es posible el acoplamiento entre la neurona software a la neurona electrónica consiguiéndose sincronismo entre las mismas.

En este apartado se validarán las pruebas de simulación de los circuitos híbridos implementados. Las pruebas de validación se han realizado con **el modelo de Hindmarsh-Rose y el modelo de Izhikevich**.

5.5.1. IMPLEMENTACIÓN DEL CIRCUITO HÍBRIDO

El diseño del circuito híbrido se basa en la conexión de la neurona software (modelo Hindmarsh-Rose o modelo Izhikevich) a la neurona viva (muestra tomada del *Carcinus maenas*), a través de dos electrodos que conectan a la neurona viva a un amplificador de señal. Uno de estos electrodos se encarga de aplicar la corriente a la muestra viva, el otro mide los potenciales de membrana que genera la misma. Desde el amplificador, ver figura 5.10, la neurona viva y la neurona software se conectan utilizando la tarjeta de adquisición de datos.



Figura 5.10. Amplificador de señal del laboratorio del GNB

Para llevar a cabo las pruebas ha sido necesario adaptar nuevamente el código utilizado, en este caso ha sido muy importante tener en cuenta que la conexión de la neurona software se ha realizado hacia una neurona viva, por lo que es un sistema cuyo comportamiento varía en cada simulación a causa de diferentes factores como puede ser la las propiedades del electrodo y de la preparación. Cualquier valor de corriente muy alto inyectado a la muestra viva puede llevar a perder su utilidad y provocar el fin de las pruebas con las muestras.

Por lo tanto, se ha utilizado un valor de acoplamiento diferente para cada neurona; g_{VM} corresponde al acoplamiento asignado a la corriente que se supone que se recibe desde la

neurona viva a la neurona software; g_{MV} es el caso contrario, se aplica a la corriente desde la neurona software a la neurona viva.

También ha sido necesario ajustar el potencial de membrana generado por el modelo, escalando y/o desplazando la corriente aplicada a la neurona software, según sea necesario, con el objetivo de ajustar los comportamientos de ambas neuronas y conseguir mayor sincronismo.

Los cambios realizados en los modelos a validar se pueden observar a continuación:

- **Hindmarsh-Rose correspondiente a la neurona software:**

$$\frac{dx_1(t)}{dt} = y_1(t) + 3x_1^2(t) - x_1^3(t) - z_1(t) + e_1 - g_{12}((x_1(t) + \mathbf{offset}) - x_2(t)) \quad (5.9)$$

$$\frac{dy_1(t)}{dt} = 1 - 5x_1^2(t) - y_1(t) \quad (5.10)$$

$$\frac{dz_1(t)}{dt} = \mu(-z_1(t) + S(x_1(t) + 1.6)) \quad (5.11)$$

- Corriente aplicada a la neurona software del modelo HR:

$x_t = x + (DT*(y + 3*x*x - x*x*x - z + e - (\mathbf{gmv} * ((x + \mathbf{offsetV}) - \mathbf{physical_value}))));$

$y_t = y + DT*(1 - 5*x*x - y);$

$z_t = z + DT*(MU*(-z + S*(x + 1.6)));$

- Corriente enviada hacia la neurona viva desde el modelo:

$\mathbf{datao} = \mathbf{from_phys}(\mathbf{gmv} * ((x + \mathbf{offsetV}) - \mathbf{physical_value}));$

- **Izhikevich correspondiente a la neurona software:**

$$\frac{dx_1(t)}{dt} = y_1(t) + 3x_1^2(t) - x_1^3(t) - z_1(t) + e_1 - g_{12}((x_1(t) + \mathbf{offset}) - x_2(t)) \quad (5.12)$$

$$\frac{dy_1(t)}{dt} = 1 - 5x_1^2(t) - y_1(t) \quad (5.13)$$

- Corriente aplicada a la neurona software del modelo EI:

```
if(v>=30)
{
    v=c;
    u=u+d;
}
v_t= v+ DT*(0.04*v*v + 5*v + 140 - u + I-(gvm*(((v*ampli)+offsetV) - physical_value)));
u_t= u+DT*a*(b*v - u);
```

- Corriente enviada hacia la neurona viva desde el modelo:

```
datao=from_phys(gmv*(((v*ampli) + offsetV)-physical_value));
```

El objetivo de estos cambios aplicados, es encontrar aquellos valores que mejor sincronizan los potenciales de membrana que ambas neuronas generan al estar afectadas por una corriente $g_{21}(x_1(t) - x_2(t))$ en el sentido desde la neurona software hacia la neurona viva, y otra corriente de expresión $g_{12}(x_2(t) - x_1(t))$ en el sentido contrario.

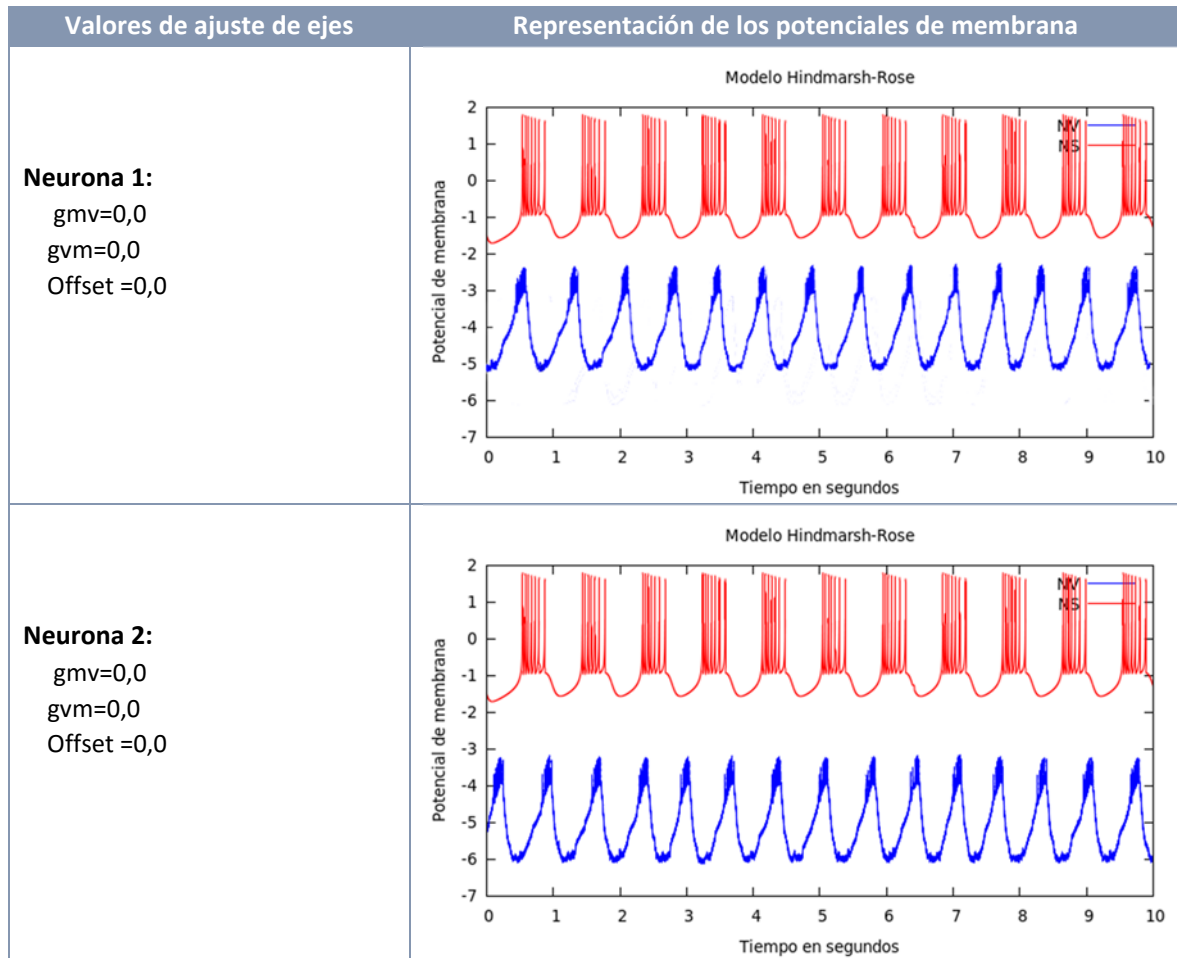
El uso de esta corriente (sinapsis eléctrica artificial) permite que la neurona software interfiera en el comportamiento de la neurona viva y viceversa; por lo tanto, se comprueba que existe una comunicación en ambos sentidos entre ambas neuronas. En el siguiente apartado se puede observar los resultados de las distintas pruebas llevadas a cabo con los circuitos híbridos y los cambios producidos al variar los valores de acoplamiento y los ajustes de ejes.

5.5.2.RESULTADOS OBTENIDOS EN LAS PRUEBAS DEL CIRCUITO HÍBRIDO FORMADO POR EL MODELO HINDMARSH-ROSE Y NEURONA VIVA

Las pruebas con el modelo de Hindmarsh-Rose en el circuito híbrido se llevaron a cabo en dos días diferentes, por lo que a continuación se pueden observar los resultados utilizando neuronas diferentes.

Como se ha comentado en apartados anteriores, al utilizarse muestras vivas es necesario realizar ajustes iniciales en cada prueba; es decir, los ajustes de amplitud y desplazamiento vertical de los potenciales de membrana utilizados en las pruebas con la primera neurona fueron diferentes a los ajustes realizados en las pruebas con la segunda neurona.

Tabla 5.6. Resultados de las simulaciones iniciales del circuito híbrido entre el modelo de Hindmarsh-Rose y la neurona utilizando una sinapsis eléctrica artificial, sin valores de acoplamiento ni offset. Se ha realizado dos pruebas con dos neuronas diferentes. En rojo se representa el potencial de membrana del modelo y en azul el de la neurona viva



Se observa que las ráfagas de los potenciales de membrana producidos por la neurona 1 están situados en valores mayores que en la neurona 2, pero mantienen la misma amplitud; por lo que el offset en ambos casos debe ser diferente.

Después de llevar a cabo varias pruebas se eligieron los siguientes valores de desplazamiento u offset que mejor ajustaban las ráfagas del modelo con las ráfagas de la neurona viva. Nótese que se desplaza la representación de la neurona software hacia valores negativo:

- Pruebas con neurona viva, día 1:

Tabla 5.7. Resultados obtenidos utilizando un único electrodo conectado a la neurona viva. Este electrodo se encarga de aplicar la corriente a la neurona y de tomar los valores producidos por la misma. Se lleva a cabo la variación de los valores de acoplamiento y de ejes con el objeto de sincronizar el comportamiento de ambas neuronas. En rojo se representa el potencial de membrana del modelo y en azul el de la neurona viva

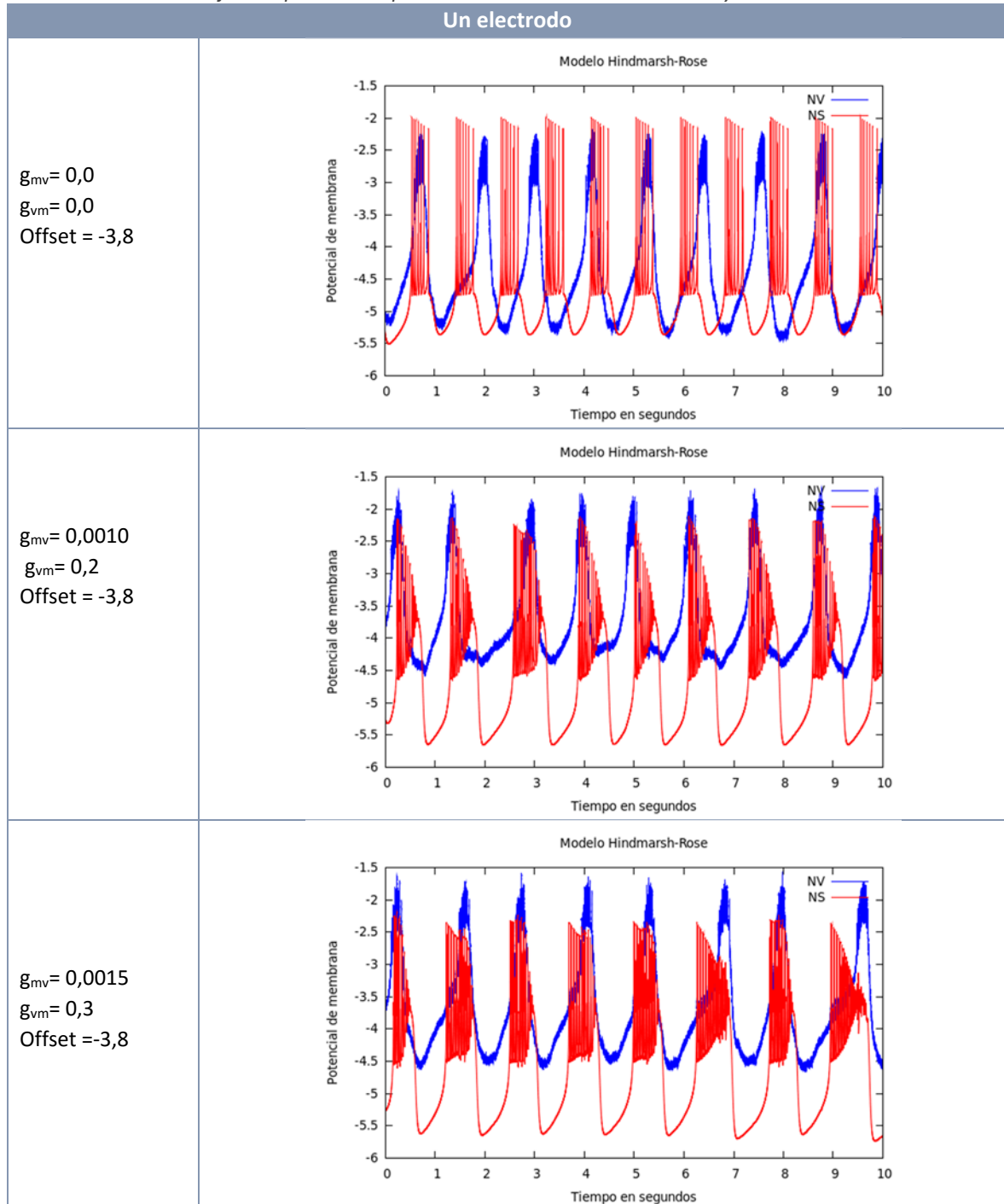


Tabla 5.8. En esta comunicación se utilizan dos electrodos conectados a la neurona viva. Un electrodo se encarga de aplicar corriente a la neurona y el otro de capturar las señales de membrana que produce la misma. En rojo se representa el potencial de membrana del modelo y en azul el de la neurona viva

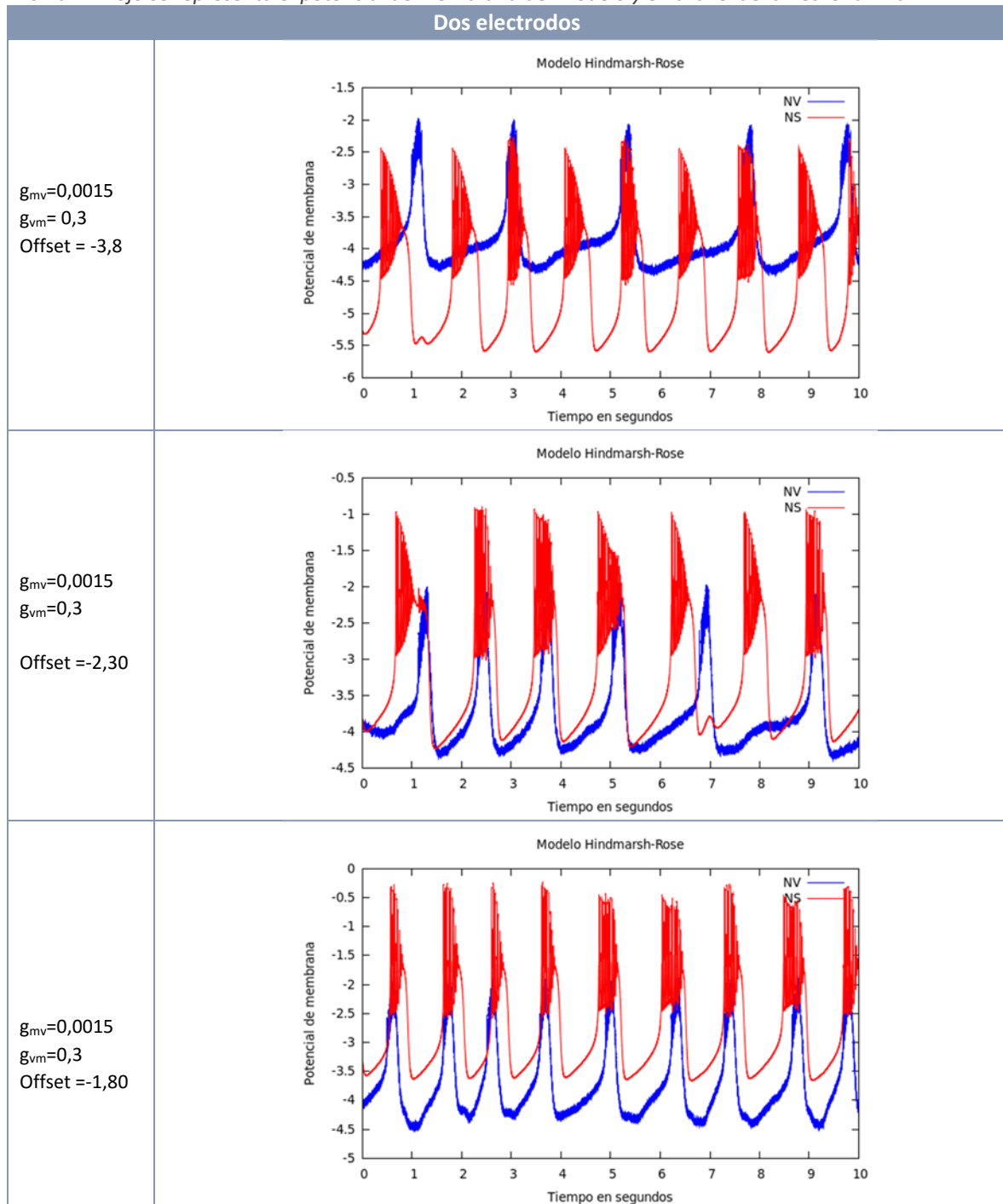
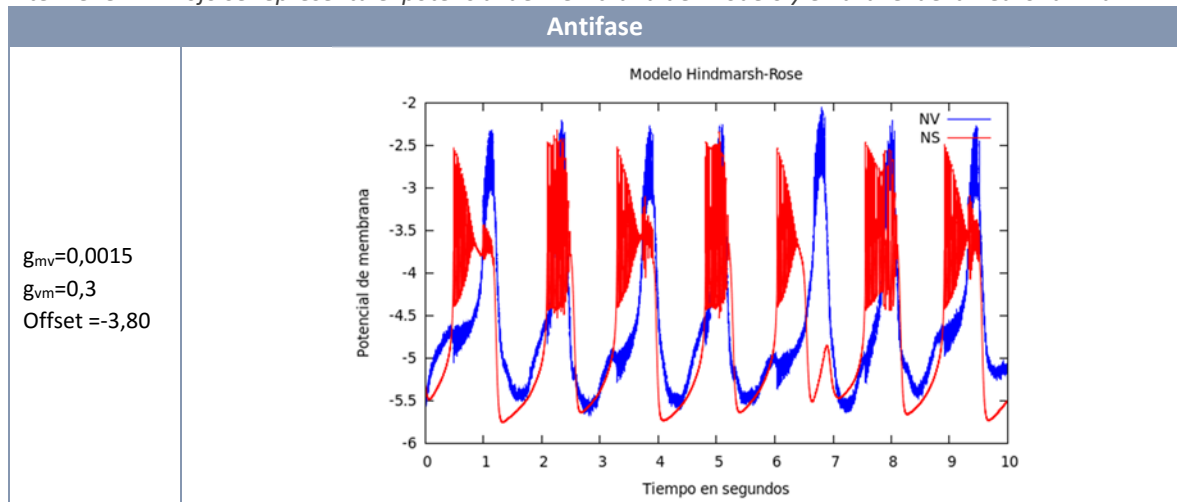
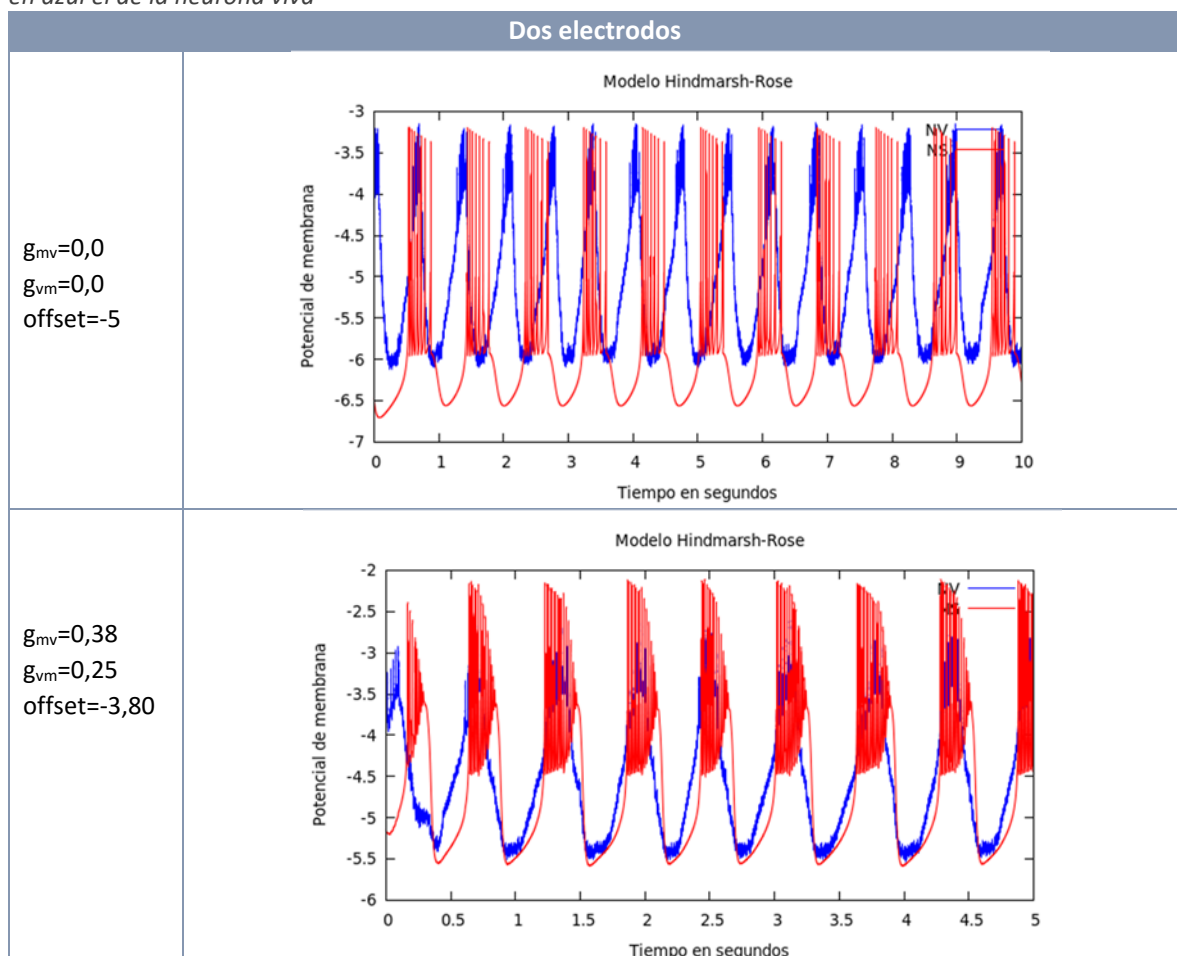


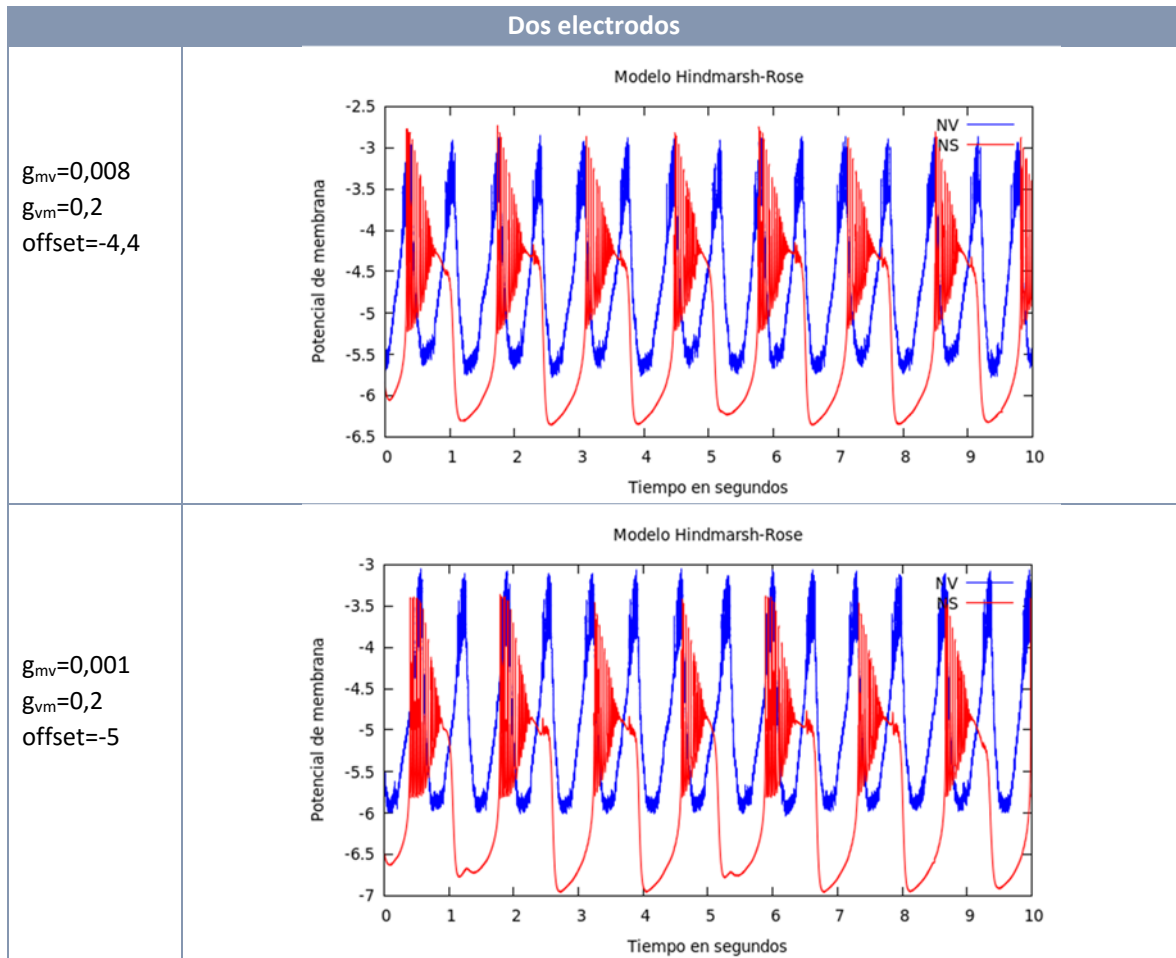
Tabla 5.9. Se comprueba que el cambio de signo en uno de los valores de acoplamiento (transmisión de una corriente negativa) cambia el comportamiento de los potenciales de membrana de las neuronas que intervienen. En rojo se representa el potencial de membrana del modelo y en azul el de la neurona viva



- Pruebas con neurona viva, día 2:

Tabla 5.10. Resultados de la comunicación utilizando dos electrodos conectados a la neurona viva del segundo día de pruebas. Un electrodo se encarga de aplicar corriente a la neurona y el otro de capturar las señales de membrana que produce la misma. En rojo se representa el potencial de membrana del modelo y en azul el de la neurona viva





5.5.3. RESULTADOS OBTENIDOS EN LAS PRUEBAS DEL CIRCUITO HÍBRIDO FORMADO POR EL MODELO IZHKEVICH Y NEURONA VIVA

Validado el circuito híbrido utilizando el modelo de Hindmarsh-Rose se ha decidido llevar a cabo las pruebas del circuito sustituyendo el modelo por el de **Izhkevich**. Con este modelo se actúa de la misma manera que en el apartado anterior, ajustando los parámetros para conseguir el sincronismo entre ambas señales.

Se inician las pruebas con todos los valores a ajustar iguales a cero. En este modelo puede observarse que además del ajuste de la posición vertical de las ráfagas, también es necesario un cambio en la amplitud.

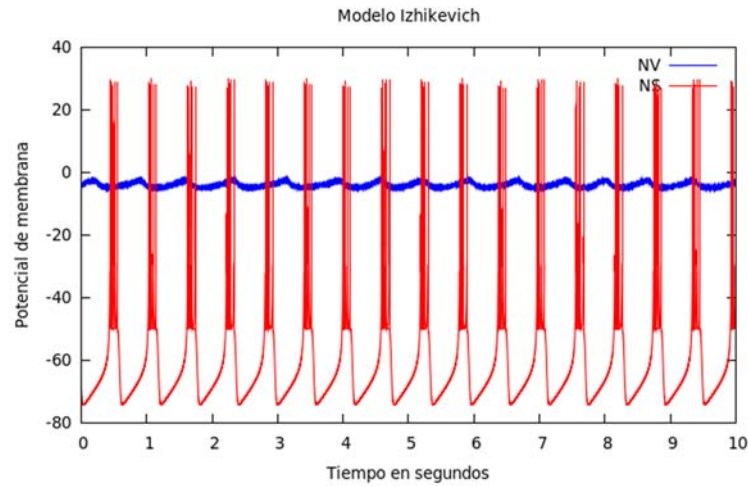


Figura 5.11. Valores iniciales de la simulación del circuito híbrido con el modelo de Izhikevich, sin ajustes de parámetros ni cambio de ejes. En rojo se representa el potencial de membrana del modelo y en azul el de la neurona viva. $g_{mv}=0,0$ $g_{vm}=0,0$ Offset=0,0 Ampli=0,0

Tabla 5.11. Resultados obtenidos de la implementación del circuito híbrido entre el modelo de Izhikevich y la neurona viva conectada a dos electrodos. En rojo se representa el potencial de membrana del modelo y en azul el de la neurona viva. Las imágenes permiten observar el cambio de comportamiento de ambas neuronas al variar el valor de acoplamiento y el ajuste de ejes. La imagen inferior presenta mayor sincronismo que la superior

Dos electrodos	
$g_{mv}=0,38$ $g_{vm}=3,0$ Offset=-2,70 Ampli=0,03	
$g_{mv}=0,38$ $g_{vm}=5$ Offset=-2,70 Ampli=0,03	

Tabla 5.12. Validación de la comunicación del circuito híbrido entre el modelo de Izhikevich y la neurona viva, variando la frecuencia de muestreo del modelo a valores diferentes de 10kHz. En rojo se representa el potencial de membrana del modelo y en azul el de la neurona viva

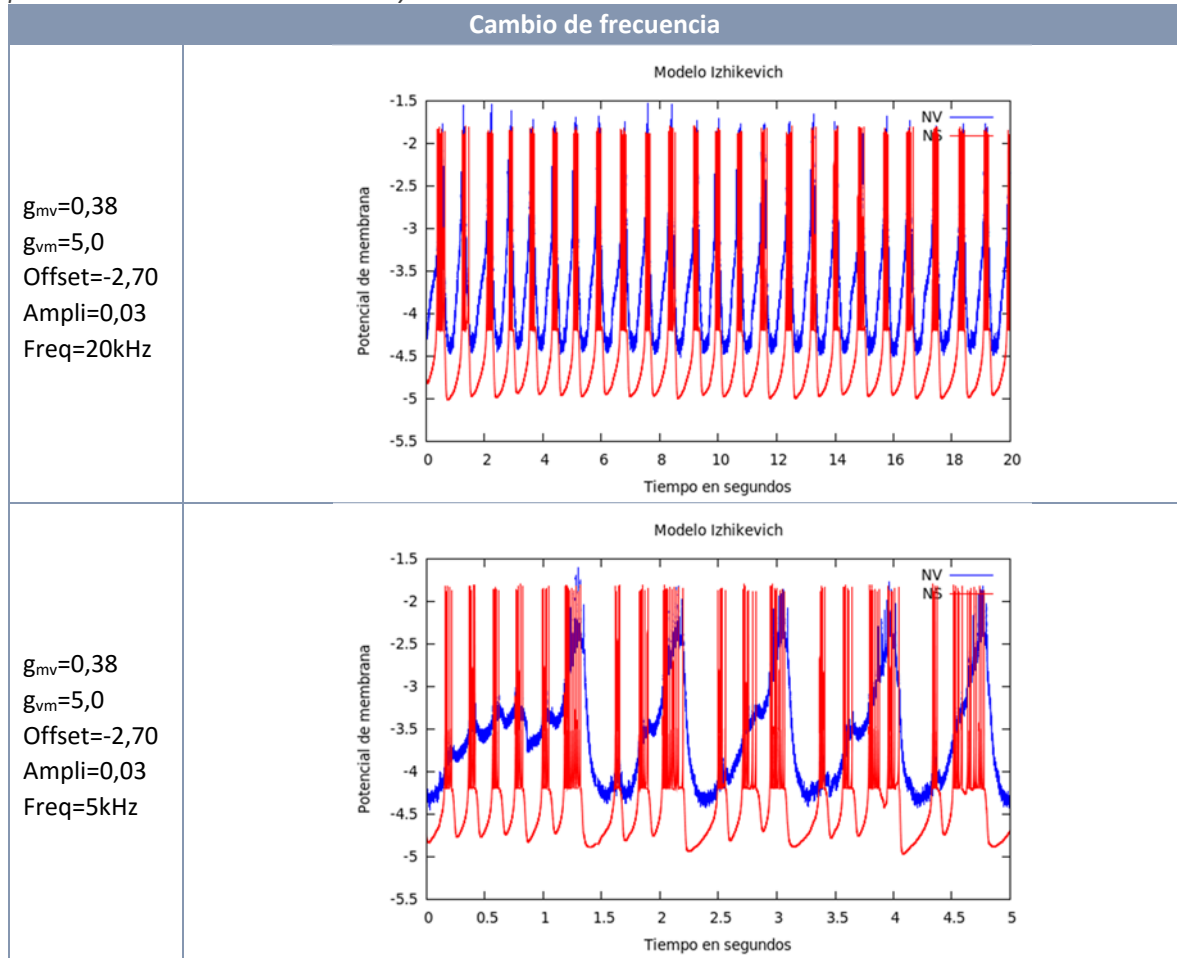
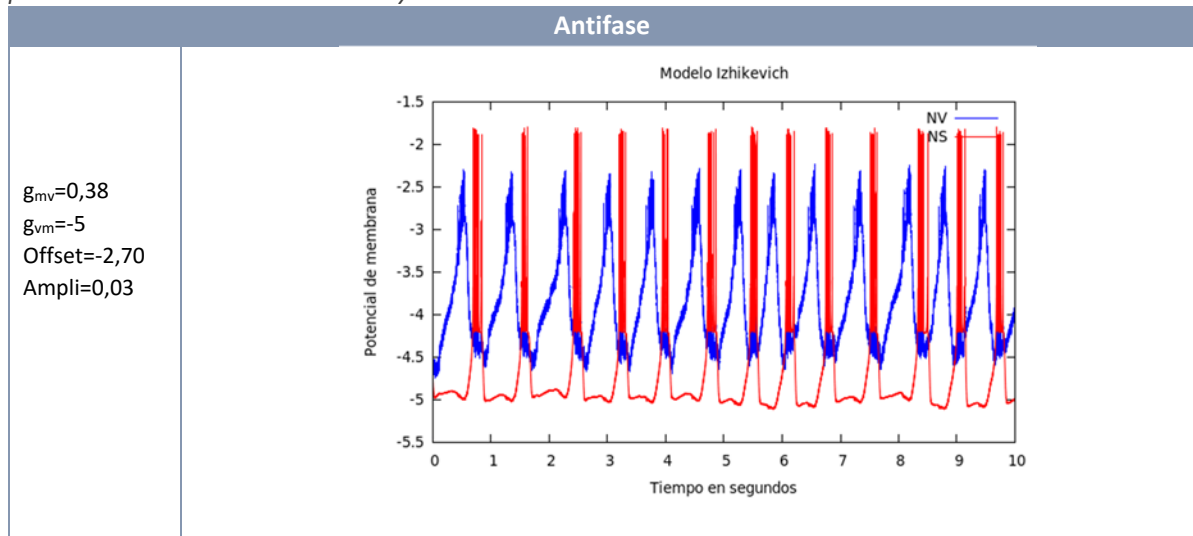


Tabla 5.13. Representación de los potenciales de membrana del circuito híbrido formado por el modelo de Izhikevich y la neurona viva, cambiando los parámetros característicos del modelo. En rojo se representa el potencial de membrana del modelo y en azul el de la neurona viva

Cambio de los parámetros propios del modelo	
$g_{mv}=0,38$ $g_{vm}=5$ Offset=-2,70 Ampli=0,03 $a=0,01$ $b=0,2$ $c=-50$ $d=2$	
$g_{mv}=0,38$ $g_{vm}=5$ Offset=-2,70 Ampli=0,03 $a=0,005$ $b=0,2$ $c=-50$ $d=2$	
$g_{mv}=0,38$ $g_{vm}=5$ Offset=-2,70 Ampli=0,03 $a=0,01$ $b=0,2$ $c=-55$ $d=3$	

Tabla 5.14. Representación en antifase de la comunicación del circuito híbrido. En rojo se representa el potencial de membrana del modelo y en azul el de la neurona viva



Validados ambos circuitos híbridos (véase apartados 5.1.4 y 5.1.5) puede comprobarse que se ha llevado a cabo la comunicación (**sinapsis eléctrica artificial**) entre ambas neuronas; es decir, la transmisión de datos se realiza en tiempo real para que sea posible la comunicación con un sistema vivo.

La variación de los valores de acoplamiento en una neurona afecta al comportamiento de la otra neurona, por lo tanto, se ha conseguido un buen nivel de sincronismo, el cual puede ser mejorado variando el valor de los ejes de representación de los potenciales de membrana, aplicando un cambio de amplitud y/o un offset al valor de la corriente, y los parámetros propios de cada modelo.

En el caso de los circuitos implementados con la neurona electrónica y la neurona software, los cambios se aplicaron a la corriente enviada a la neurona electrónica. En los circuitos híbridos (neurona software y neurona viva), los cambios se realizaron en la corriente que afecta al modelo utilizado, pues ha sido importante que la corriente enviada a la neurona viva se mantuviera constante en mínimos valores, evitando producirle daños a la muestra viva.

6. CONCLUSIONES

Este proyecto ha tenido tres objetivos concretos. El primero ha sido el estudio de modelos neuronales susceptibles de ser implementados en software para la construcción de circuitos híbridos en los que interaccionan bidireccionalmente estos modelos y neuronas vivas. El segundo ha sido llevar a cabo la interacción bidireccional entre neuronas software y la neurona electrónica del GNB como primera validación de su correcto funcionamiento. El tercer y último objetivo a la validación de su implementación en circuitos híbridos; es decir, la comunicación bidireccional a partir de una sinapsis eléctrica artificial, entre el modelo desarrollado en software y una neurona viva. Ambos casos se han validado en tiempo real o tiempo online. Para el cumplimiento de los objetivos planteados se han llevado a cabo los siguientes pasos:

1. **Diseño:** Se implementaron varios modelos de neuronas propuestos inicialmente, comparando los resultados de sus simulaciones, y eligiéndose sólo aquellos modelos que cumplían restricciones temporales estrictas y que presentaban un comportamiento en ráfaga de su potencial de membrana (Modelo de Hindmarsh-Rose, Modelo de Izhikevich y el Mapa de Rulkov) para elaborar posteriormente circuitos híbridos en CPGs.

Para el estudio de los modelos fue necesario:

- Elegir el método matemático más adecuado para resolver las EDOs que describen los modelos de neuronas, a excepción de los mapas iterados como es el caso del Mapa de Rulkov, cuya resolución es directa. La elección fue llevada a cabo entre el método de Euler y el método de Runge-Kutta de cuarto orden. Después de comparar resultados se llegó a la conclusión de que ambos métodos presentaban resultados similares. Se decidió utilizar Euler como método resolutivo de las EDOs, ya que el método de Runge Kutta puede presentar problemas en algunos modelos y, aunque el método de Euler requiere un paso de integración menor, se mostró eficaz en las pruebas realizadas.
- Elegir el paso de integración más adecuado. En este caso se optó por elegir un paso fijo para que se cumplieran las restricciones de simulación en tiempo real marcadas por el comportamiento de la neurona viva. Por lo tanto, utilizando un paso fijo elegido después de llevar a cabo varias pruebas, los modelos muestreaban 10.000 puntos correspondientes al valor de su potencial de membrana, en un tiempo de 1 segundo, importante para las pruebas posteriores en las que se conectaría el modelo software al exterior utilizando una tarjeta de adquisición de datos a una frecuencia de 10kHz. Además se impuso que los modelos generaran una ráfaga de potenciales de acción en el tiempo de 1 segundo, la referencia temporal de la preparación de neuronas vivas. En el caso del Mapa de Rulkov se generaban menos puntos de los deseados por ráfaga, por lo que se utilizó el método de interpolación lineal para completar los 10.000 puntos por ráfaga.

- Cálculo del retraso temporal para el funcionamiento de los modelos en circuitos híbridos. Los modelos de neuronas elegidos presentaban un comportamiento rápido, muestreaban los 10000 puntos muy por debajo del segundo en el que se deseaba que sucediera. Por lo tanto, fue necesario implementar un sistema de retardo para que cada valor se generase en un periodo adecuado, y así fuera posible la sincronización con la neurona viva.

2. **Desarrollo:** Elegidos los modelos de neuronas (Modelo de Hindmarsh-Rose, Modelo de Izhikevich y el Mapa de Rulkov) que iban a ser utilizados para implementar los circuitos híbridos fue necesario adaptar el código para su uso con la neurona electrónica del laboratorio del GNB, y así comprobar que el muestreo de los valores se realizaba de manera correcta y que se podía conseguir sincronismo entre ambas neuronas, para ello se llevaron a cabo los siguientes pasos:

- Se adaptó el código desarrollado para la resolución de los modelos neuronales, añadiendo las funciones necesarias para la conexión con la tarjeta de adquisición de datos del laboratorio del GNB, y para que su simulación se llevara a cabo sobre un sistema operativo en tiempo real (RTAI), el cual daría prioridad al modelo sobre las otras tareas que en ese mismo momento se estuvieran ejecutando. Se utilizaron las funciones de las librerías que ponía a disposición el uso de RTAI para llevar a cabo el control del sistema y la medición del tiempo transcurrido.

- Se realizaron varias pruebas transmitiendo únicamente el valor de potencial de membrana generado, desde la neurona software hacia la neurona electrónica y viceversa. Se comprobó que la transmisión y recepción de datos era satisfactoria y se realizaba dentro del tiempo marcado.

- Se añadió un valor de corriente al circuito, por lo que ambas neuronas se comunicaban a través de una sinapsis eléctrica creada artificialmente. Es decir, al potencial de membrana de los modelos de neuronas implementados se le añadió un valor definido por $-g_{12}(x_1(t) - x_2(t))$ para la corriente enviada desde la neurona software, y otra corriente con la misma expresión pero en sentido contrario desde la neurona electrónica.

Por lo tanto, desde la neurona electrónica se recibía un valor de potencial de membrana y a partir de éste se calculaba el valor de corriente utilizando un acoplamiento adecuado, el cual era distinto para cada modelo. El valor de acoplamiento se eligió realizando varias simulaciones y seleccionando aquel que permitía una mayor sincronización entre las neuronas.

- También fue necesario modificar la amplitud y offset de la señales generadas por la neurona software para adaptarlas a los valores del potencial de la neurona electrónica:

$$\frac{g(x_s(t) - x_E(t))}{\mathbf{Ampli}} - \mathbf{offset}$$

Variando los parámetros denotados como **Ampli** y **offset**, se ayudaba al circuito a conseguir la sincronización entre las neuronas sin alterar el modelo.

3. **Validación:** Comprobada la sincronización entre la neurona software y la neurona electrónica y, por lo tanto, la implementación de una sinapsis eléctrica artificial entre dos neuronas artificiales, se llevó a cabo la validación de los circuitos híbridos compuestos por una neurona artificial (software) y una neurona viva, comunicadas a través de una sinapsis eléctrica creada artificialmente. Para llevar a cabo la validación de estos circuitos se decidió utilizar el modelo de Hindmarsh-Rose y el modelo de Izhikevich.
- No fue necesario modificar el código, se utilizaron los mismos que se habían probado con la neurona electrónica.
 - Se realizó la preparación *in vitro* y se utilizó la neurona PD del sistema nervioso de un *Carcinus Maenas* o cangrejo común, conectando la muestra viva con electrodos al equipo del GNB para realizar su comunicación con la neurona software a través de la tarjeta de adquisición de datos, tal y como se había hecho con la neurona electrónica (preparación llevada a cabo por otros miembros del laboratorio del GNB).
 - Antes de conectar el modelo a la neurona viva fue necesario ajustar nuevamente los valores de la variable de potencial generada por el modelo, escalando dicho valor para que pudiera ser comparado con los valores generados por la neurona viva.
 - Elegido el valor de corriente más adecuado, se llevó a cabo la interacción entre ambas neuronas, consiguiéndose resultados de sincronismo muy aceptables en los dos modelos probados. Se validó así el desarrollo de circuitos híbridos con neuronas software comunicados bidireccionalmente a partir del uso de una sinapsis eléctrica artificial.

El trabajo realizado en este proyecto muestra que es posible implementar modelos neuronales en software y utilizarlos para el estudio de la dinámica de circuitos neuronales mediante la construcción de circuitos híbridos con neuronas vivas.

7. TRABAJOS FUTUROS

Para continuar con el desarrollo de este proyecto se plantean los siguientes trabajos complementarios:

- Desarrollo de nuevos modelos de sinapsis, más allá de la conectividad eléctrica, e implementación de otros tipos de modelos de neuronas, lo que permitiría la creación de una librería de modelos neuronales y de sinapsis para integrar en el sistema de pruebas RTBiomanager del laboratorio del GNB.
- Implementación de otros modelos neuronales de conductancias más realistas, principalmente de aquellos que presentan un comportamiento en ráfaga muy lento como es el caso del modelo de Thomas Nowotny et al., en el que será necesario el uso de procedimientos de aceleración de la integración, como la tabulación de las no linealidades para su ejecución en tiempo real.
- Automatización del ajuste de los parámetros del modelo y de la calibración en las comunicaciones bidireccionales.

8. REFERENCIAS

- van Boxtel, G. J. M., and Gruzelier, J. H. (2014). Neurofeedback: introduction to the special issue. *Biol. Psychol.* 95, 1–3. doi:10.1016/j.biopsycho.2013.11.011.
- Chamorro, P., Muñiz, C., Levi, R., Arroyo, D., Rodríguez, F. B., and Varona, P. (2012). Generalization of the Dynamic Clamp Concept in Neurophysiology and Behavior. *PLoS One* 7, e40887. doi:10.1371/journal.pone.0040887.
- Chen, W. R., Midtgaard, J., and Shepherd, G. M. (1997). Forward and backward propagation of dendritic impulses and their synaptic control in mitral cells. *Science* 278, 463–467.
- Cohen, A., Shappir, J., Yitzchaik, S., and Spira, M. E. (2006). Experimental and theoretical analysis of neuron-transistor hybrid electrical coupling: the relationships between the electro-anatomy of cultured Aplysia neurons and the recorded field potentials. *Biosens. Bioelectron.* 22, 656–63. doi:10.1016/j.bios.2006.02.005.
- Cole, K. (1968). Membranes, Ions and Impulses: A Chapter of Classical Biophysics. *Univ. Calif. Press. Berkeley.*
- Cole, K. S., and Curtis, H. J. (1939). ELECTRIC IMPEDANCE OF THE SQUID GIANT AXON DURING ACTIVITY. *J. Gen. Physiol.* 22, 649–670.
- Connor, J. a, Walter, D., and McKown, R. (1977). Neural repetitive firing: modifications of the Hodgkin-Huxley axon suggested by experimental results from crustacean axons. *Biophys. J.* 18, 81–102. doi:10.1016/S0006-3495(77)85598-7.
- Destexhe, A., and Bal, T. (2009). *Dynamic Clamp: From Principles to Applications.* Springer.
- Fernandez-Vargas, J., Pfaff, H. U., Rodríguez, F. B., Varona, P., and Rodriguez, F. B. (2013). Assisted closed-loop optimization of SSVEP-BCI efficiency. *Front. Neural Circuits* 7, 27. doi:10.3389/fncir.2013.00027.
- Fitzhugh, R. (1961). Impulses and Physiological States in Theoretical Models of Nerve Membrane. *Biophys. J.* 1, 445–466.
- Graham (2011). Principles of Computational Modelling in Neuroscience. *IEEE Pulse* 3, 82–82. doi:10.1109/MPUL.2012.2196841.
- Grant, G. (2007). How the 1906 Nobel Prize in Physiology or Medicine was shared between Golgi and Cajal. *Brain Res. Rev.* 55, 490–498.
- Hebb, D. O. (1949). *The Organization of Behaviour.* Wiley & Sons.
- Herreras, O. (1990). Propagating dendritic action potential mediates synaptic transmission in CA1 pyramidal cells in situ. *J. Neurophysiol.* 64, 1429–1441.

- Herrero-Carrón, F., Rodríguez, F. B., and Varona, P. (2011). Bio-inspired design strategies for central pattern generator control in modular robotics. *Bioinspir. Biomim.* 6, 016006. doi:10.1088/1748-3182/6/1/016006.
- Hille, B. (1978). Ionic Channels in Excitable membranes. *Biophys. J.* 22, 283–294. Available at: <http://www.getcited.org/pub/102366977>.
- Hindmarsh, J. L., and Rose, R. M. (1984). A model of neuronal bursting using three coupled first order differential equations. *Proc. R. Soc. Lond. B. Biol. Sci.* 221, 87–102.
- Hodgkin, A. L., and Huxley, A. F. (1952). A quantitative description of membrane current and its applications to conduction and excitation in nerve. *J. Physiol.* 117, 500–544. Available at: <http://www.sciencedirect.com/science/article/B6WC7-4GP2506-4/2/8a68ac710fc6dd933c684747f30d96f3>.
- Hooper, R. M., Tikidji-Hamburyan, R. A., Canavier, C. C., and Prinz, A. A. (2015). Feedback control of variability in the cycle period of a central pattern generator. *J. Neurophysiol.* 114, 2741–52. doi:10.1152/jn.00365.2015.
- Hormuzdi, S. G., Filippov, M. A., Mitropoulou, G., Monyer, H., and Bruzzone, R. (2004). Electrical synapses: A dynamic signaling system that shapes the activity of neuronal networks. *Biochim. Biophys. Acta - Biomembr.* 1662, 113–137.
- Ijspeert, A. J. (2008). Central pattern generators for locomotion control in animals and robots: a review. *Neural Netw.* 21, 642–53. doi:10.1016/j.neunet.2008.03.014.
- Izhikevich, E. M. (2003). Simple model of spiking neurons. *IEEE Trans. Neural Netw.* 14, 1569–72. doi:10.1109/TNN.2003.820440.
- Kandel, E. R., Schwartz, J. H., and Jessell, T. M. (2000). *Principles of Neural Science.*, eds. E. R. Kandel, J. H. Schwartz, and T. Me. Jessell McGraw-Hill doi:10.1036/0838577016.
- Kepler, T. B., Abbott, L. F., and Marder, E. (1992). Reduction of conductance-based neuron models. *Biol. Cybern.* 66, 381–387. doi:10.1007/BF00197717.
- López-Muñoz, F., Boya, J., and Alamo, C. (2006). Neuron theory, the cornerstone of neuroscience, on the centenary of the Nobel Prize award to Santiago Ramón y Cajal. *Brain Res. Bull.* 70, 391–405. doi:10.1016/j.brainresbull.2006.07.010.
- Manor, Y., and Nadim, F. (2001). Frequency regulation demonstrated by coupling a model and a biological neuron. *Neurocomputing* 38-40, 269–278.
- Le Masson, G., Renaud-Le Masson, S., Debay, D., and Bal, T. (2002). Feedback inhibition controls spike transfer in hybrid thalamic circuits. *Nature* 417, 854–858.
- Le Masson, S., Laflaquière, A., Bal, T., and Le Masson, G. (1999). Analog circuits for modeling biological neural networks: Design and applications. *IEEE Trans. Biomed. Eng.* 46, 638–645.

- McCulloch, W. S., and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.* 5, 115–133.
- Morris, C., and Lecar, H. (1981). Voltage oscillations in the barnacle giant muscle fiber. *Biophys. J.* 35, 193–213.
- Muñiz, C., Rodríguez, F., and Varona, P. (2009). RTBiomanager: a software platform to expand the applications of real-time technology in neuroscience. *BMC Neurosci.* 10, P49. doi:10.1186/1471-2202-10-S1-P49.
- Nagumo, J., Arimoto, S., and Yoshizawa, S. (1962). An Active Pulse Transmission Line Simulating Nerve Axon. *Proc. IRE* 50.
- Nowotny, T., Levi, R., and Selverston, A. I. (2008). Probing the dynamics of identified neurons with a data-driven modeling approach. *PLoS One* 3, e2627. doi:10.1371/journal.pone.0002627.
- Nowotny, T., Szucs, A., Pinto, R. D., and Selverston, A. I. (2006). Stdpc: A modern dynamic clamp. *J. Neurosci. Methods* 158, 287–299. doi:10.1016/j.jneumeth.2006.05.034.
- Paniagua, R., Nistal, M., Sesma, P., Álvarez-Uría, M., Fraile, B., Anadón, R., et al. (2002). *Citología e histología vegetal y animal.*, ed. S. A. U. I. 84-486-0436-9. McGraw-Hill Interamericana de España.
- Pinto, R., Varona, P., Volkovskii, a., Szücs, a., Abarbanel, H., and Rabinovich, M. (2000). Synchronous behavior of two coupled electronic neurons. *Phys. Rev. E* 62, 2644–2656. doi:10.1103/PhysRevE.62.2644.
- Potter, S. M., El Hady, A., and Fetz, E. E. (2014). Closed-loop neuroscience and neuroengineering. *Front. Neural Circuits* 8, 115. doi:10.3389/fncir.2014.00115.
- Prescott, S. a, De Koninck, Y., and Sejnowski, T. J. (2008). Biophysical basis for three distinct dynamical mechanisms of action potential initiation. *PLoS Comput. Biol.* 4, e1000198. Available at: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2551735&tool=pmcentrez&rendertype=abstract>.
- Prinz, A. A., Abbott, L. F., and Marder, E. (2004). The dynamic clamp comes of age. *Trends Neurosci.* 27, 218–224.
- Rabinovich, M. I., Varona, P., Selverston, A. I., and Abarbanel, H. D. I. (2006). Dynamical principles in neuroscience. *Rev. Mod. Phys.* 78, 1213–1265. doi:10.1103/RevModPhys.78.1213.
- Ramón y Cajal, S. (1909). *Histologie du Systeme Nerveux de l'Homme et des Vertebres*. Paris: Maloine.
- Rodriguez, F. B., Varona, P., Huerta, R., Rabinovich, M. I., and Henry, D. I. (2001). Richer network dynamics of intrinsically non-regular neurons measured through mutual information. Available at: <http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:No+Title#0>

- Roth, E., Sponberg, S., and Cowan, N. J. (2014). A comparative approach to closed-loop computation. *Curr. Opin. Neurobiol.* 25, 54–62. doi:10.1016/j.conb.2013.11.005.
- Rulkov, N. F. (2002). Modeling of spiking-bursting neural behavior using two-dimensional map. *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.* 65.
- Samu, D., Marra, V., Kemenes, I., Crossley, M., Kemenes, G., Staras, K., et al. (2012). Single electrode dynamic clamp with StpC. *J. Neurosci. Methods* 211, 11–21. doi:10.1016/j.jneumeth.2012.08.003.
- Selverston, A. I., Rabinovich, M. I., Abarbanel, H. D. I., Elson, R., Szücs, A., Pinto, R. D., et al. (2000). Reliable circuits from irregular neurons: a dynamical approach to understanding central pattern generators. *J. Physiol. (Paris)*, 94 357-374. 94, 357–374.
- Sharp, A. A., O’Neil, M. B., Abbott, L. F., and Marder, E. (1993). Dynamic clamp: computer-generated conductances in real neurons. *J. Neurophysiol.* 69, 992–995.
- Sharp, A. A., Skinner, F. K., and Marder, E. (1996). Mechanisms of oscillation in dynamic clamp constructed two-cell half-center circuits. *J. Neurophysiol.* 76, 867–883.
- Simmons, G. (2007). *Ecuaciones Diferenciales Teoría y Práctica*. 1st ed. , ed. McGraw-Hill Interamericana de España S.L.
- Skinner, F. K., Turrigiano, G. G., and Marder, E. (1993). Frequency and burst duration in oscillating neurons and two-cell networks. *Biol Cybern* 69, 375–83. doi:10.1007/bf00199437.
- Stiesberg, G. R., Reyes, M. B., Varona, P., Pinto, R. D., and Huerta, R. (2007). Connection topology selection in central pattern generators by maximizing the gain of information. *Neural Comput.* 19, 974–993. doi:10.1162/neco.2007.19.4.974.
- Stuart, G., Schiller, J., and Sakmann, B. (1997). Action potential initiation and propagation in rat neocortical pyramidal neurons. *J. Physiol.* 505 (Pt 3, 617–632.
- Szücs, A., Varona, P., Volkovskii, A. R., Abarbanel, H. D. I., Rabinovich, M. I., and Selverston, A. I. (2000). Interacting biological and electronic neurons generate realistic oscillatory rhythms. *Neuroreport* 11, 563–569. doi:10.1097/00001756-200002280-00027.
- Varona, P., Torres, J. J., Abarbanel, H. D. I., Rabinovich, M. I., and Elson, R. C. (2001). Dynamics of two electrically coupled chaotic neurons: Experimental observations and model analysis. *Biol. Cybern.* 84, 91–101. doi:10.1007/s004220000198.
- Wallach, A. (2013). The response clamp: functional characterization of neural systems using closed-loop control. *Front. Neural Circuits* 7, 5. doi:10.3389/fncir.2013.00005.
- Williams, R. W., and Herrup, K. (1988). The control of neuron number. *Annu. Rev. Neurosci.* 11, 423–453.

Wilson, R. A., and Keil, F. C. (2001). *The MIT encyclopedia of the cognitive sciences*. Available at: <http://www.mitpressjournals.org/doi/pdfplus/10.1162/coli.2000.26.3.463>.

Yarom, Y. (1991). Rhythmogenesis in a hybrid system - Interconnecting an olivary neuron to an analog network of coupled oscillators. *Neuroscience* 44, 263–275.

Zill, D. (2008). *Matemáticas avanzadas para ingeniería vol. I: ecuaciones diferenciales.* , ed. McGraw-Hill Interamericana de Mexico.

9. GLOSARIO

AP:	Action Potenciales
CH:	Chattering
COMEDI:	Control and Measurement Device Interface
CPG:	Central Pattern Generator (Generador Central de Patrones)
DAQ:	Data Acquisition (Tarjeta de Adquisición de Datos)
EI:	Eugene Izhikevich
FHN:	FitzHugh Nagumo
FP:	Field Potentials
GNB:	Grupo de Neurocomputación Biológica
HH:	Hodgkin-Huxley
HR:	Hindmarsh-Rose
I&F:	Integration and Fire (Integración y disparo)
MR:	Mapa de Rulkov
NE:	Neurona Electrónica
NS:	Neurona Software
NV:	Neurona Viva
PC	Personal Computer
RK	Runge Kutta
RTAI:	Real Time Application Interface
Stdpc:	Spike timing-dependent plasticity Clamp
STG:	Stomatogastric Ganglion
TN:	Thomas Nowotny
UAM	Universidad Autónoma de Madrid

10. ANEXO A

10.1. MODELO HODGKIN-HUXLEY

El código base desarrollado para la implementación del modelo Hodgkin-Huxley, a partir del uso del método matemático de Euler, es el siguiente:

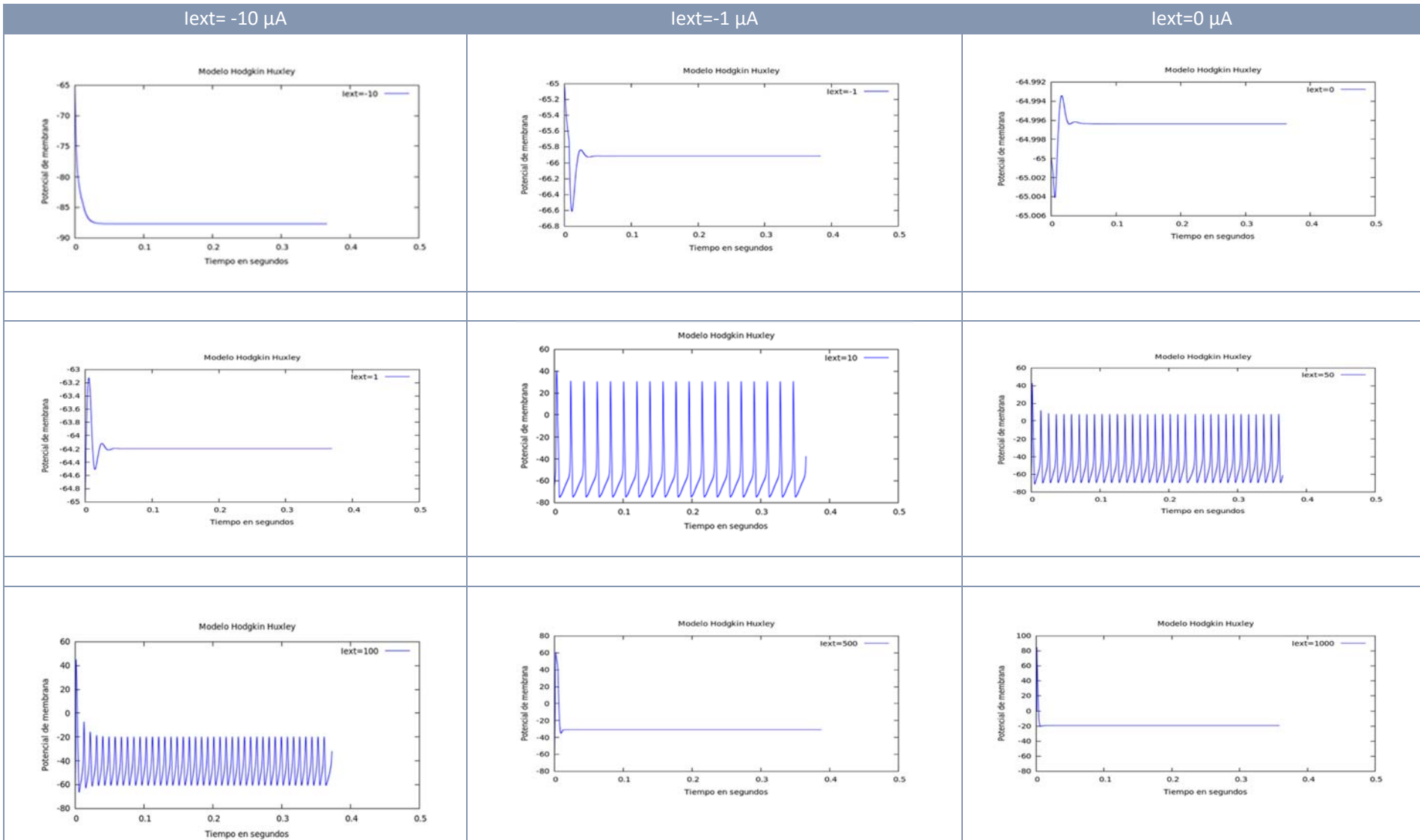
```
for(t=DT; t<=TIEMPO; t+=DT){  
  
    alpha_m= (0.1*(-x-40))/(exp((-x-40)/10)-1);  
    beta_m=4*exp((-x-65)/18);  
  
    alpha_h=0.07*exp((-x-65)/20);  
    beta_h=1/(exp((-x-35)/10)+1);  
  
    alpha_n=(0.01*(-x-55))/(exp((-x-55)/10)-1);  
    beta_n=0.125*exp((-x-65)/80);  
  
    x1= x + ((DT/Cm)*((Iext/area)-gL*(x-xL)-gNa*(m*m*m)*h*(x-xNa)-gK*(n*n*n*n)*(x-xK)));  
  
    m1=m+(DT*(alpha_m*(1-m)-beta_m*m));  
    h1= h+(DT*(alpha_h*(1-h)-beta_h*h));  
    n1=n+(DT*(alpha_n*(1-n)-beta_n*n));  
  
    x=x1;  
    m=m1;  
    h=h1;  
    n=n1;  
  
}
```

- **Valor de las variables utilizadas:**

TIEMPO	280	XK	-77 mV	area	1µcm ²
DT	0,001	xL	-54,387 mV	x	-65,0
Iext	Valor elegido (µA)	gNa	120 mS/cm ²	m	0,053
Cm	1 µF/cm ²	gK	36 mS/cm ²	h	0,6
xNa	50 mV	gL	0,3 mS/cm ²	n	0,318

- **Simulaciones para distintos valores de Iext:**

Para observar como varia el comportamiento del modelo de Hodgkin-Huxley para distintos valores dados a la corriente externa Iext se ha llevado a cabo varias simulaciones, utilizando en todas ellas para el valor de las variables TIEMPO =280 y DT (paso de integración) =0,001 y el método de Euler.



10.2. MODELO FITZHUGH NAGUMO

El código base desarrollado para la implementación del modelo Fitzhugh Nagumo, a partir del uso del método matemático de Euler, es el siguiente:

```
for(t=DT; t<=TIEMPO; t+=DT)
{
    v1=v+(DT*(v-(v*v*v)/3 - w + Iext));
    w1=w+(DT*(v+a-b*w)/tao);
    v=v1;
    w=w1;
}
```

- **Valor de las variables utilizadas**

TIEMPO	280	Iext	0,5
DT	0,001	a	0,7
v	1.606562	b	0,8
w	0.038222	tao	12,5

Para el caso $a=b=0$, se reduce al oscilador de Van der Pol, obteniéndose el siguiente comportamiento, ver figura 9.1:

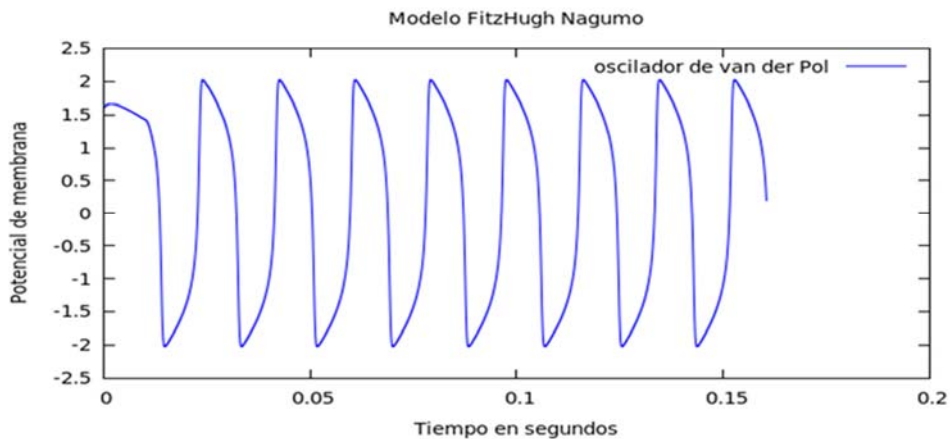
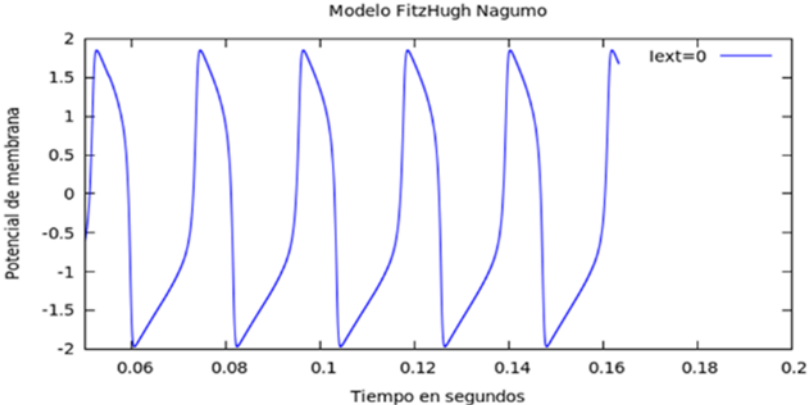
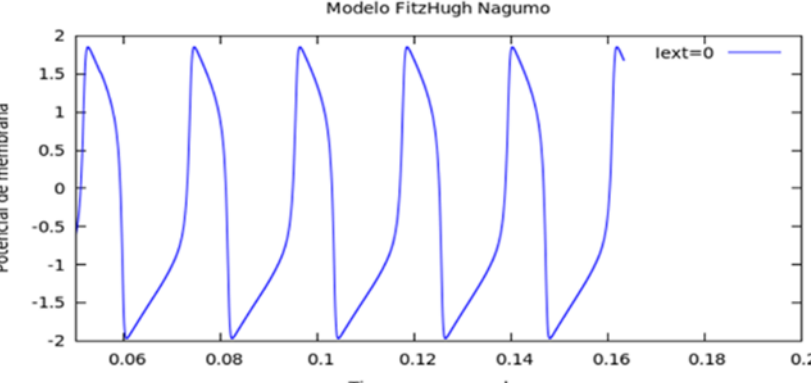
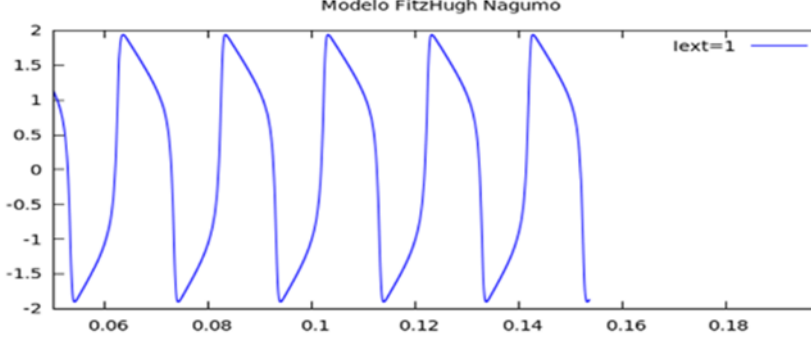
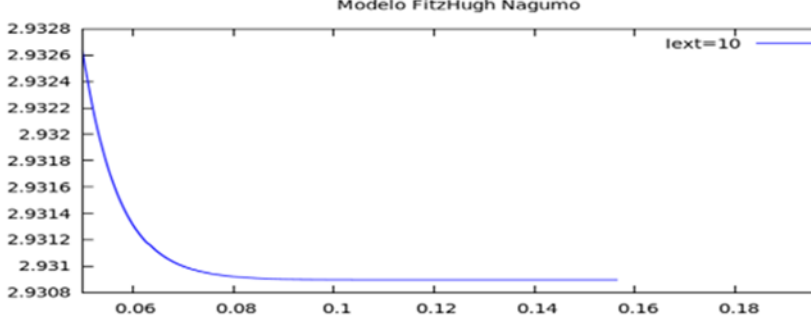


Figura 10.1 Comportamiento del oscilador de Van der Pol, caso $a=b=0$

Para distintos valores de I_{ext} , utilizando el método de Euler y las variables determinadas en la tabla anterior, se obtienen las siguientes simulaciones con $TIEMPO = 280$ y $DT = 0,001$.

Tabla 10.1. Resultados obtenidos de la simulación del modelo Fitzhugh Nagumo para distintos valores de I_{ext} . Se observa que para valores de I_{ext} igual o mayores a 10 mA el modelo pierde su comportamiento de spiking (no produce potenciales de acción)

I _{ext} (mA)	Representación
0	
0,5	
1	
10	

10.3. MODELO HINDMARSH-ROSE

El código base desarrollado para la implementación del modelo Hindmarsh-Rose, a partir del uso del método matemático de Euler, es el siguiente:

```
for(t=DT; t<=TIEMPO; t+=DT)
{
    x1= x + (DT*(y + 3*x*x - x*x*x - z + e));
    y1= y + DT*(1 - 5*x*x - y);
    z1= z + DT*(MU*(-z + S*(x + 1.6)));
    x=x1;
    y=y1;
    z=z1;
}
```

- **Valor de las variables utilizadas**

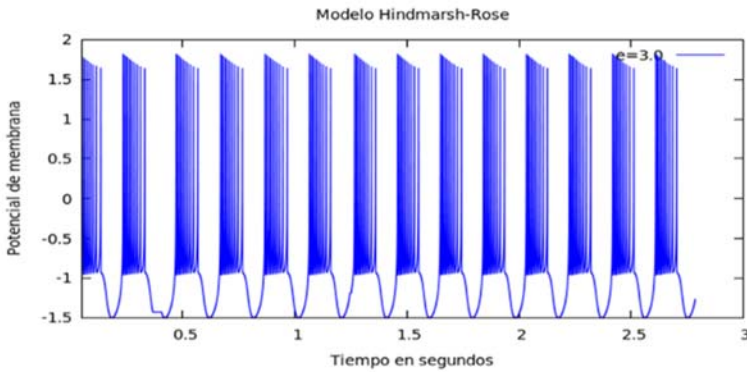
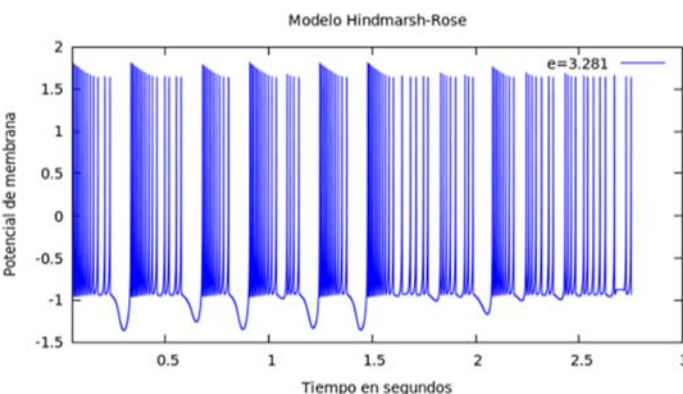
TIEMPO	280	S	4
DT	0,001	x	-1.464213
MU	0.0021	y	-9.771895
e	3.0 (Regular)	z	2.795284

El modelo HR presenta un comportamiento regular para $e=3.0$ y no regular en el caso en el que la variable $e=3,281$.

En algunas representaciones de este modelo, la variable aquí llamada e es nombrada con el nombre I de corriente externa.

Utilizando los datos de la tabla anterior, se lleva a cabo dos simulaciones con $TIEMPO=4000$, en las que se pueden observar los resultados obtenidos si se simula con un valor e regular o no regular.

Tabla 10.2. Resultados obtenidos de la simulación del modelo de Hindmarsh-Rose. La imagen superior muestra el comportamiento regular del potencial de membrana del modelo cuando $e=3,0$. La imagen inferior, con $e=3,281$ representa el comportamiento no regular o caótico del modelo.

Valor de e	Representaciones
3,0	 <p>The graph, titled 'Modelo Hindmarsh-Rose', shows the membrane potential over a 3-second period for $e=3.0$. The y-axis is labeled 'Potencial de membrana' and ranges from -1.5 to 2. The x-axis is labeled 'Tiempo en segundos' and ranges from 0 to 3. The plot displays a series of regular, periodic spikes that reach a peak of approximately 1.7 and a trough of approximately -1.4, with a consistent interval between each spike.</p>
3,281	 <p>The graph, titled 'Modelo Hindmarsh-Rose', shows the membrane potential over a 3-second period for $e=3.281$. The axes are the same as in the top graph. The plot displays a series of irregular, chaotic oscillations. The spikes are no longer periodic, with varying intervals and amplitudes, indicating a transition to chaotic behavior.</p>

10.4. MAPA DE RULKOV

Este modelo no necesita el uso de métodos matemáticos para su resolución, por lo que no depende del paso de integración elegido. El código C desarrollado para su implementación es el siguiente:

```
for(t=DT; t<=TIEMPO; t+=DT)
{
    x=x1;
    y=y1;
    if (x<=0)
    {
        x1=ALPHA/(1-x)+y;
        y1=y-MU*(x+1-SIGMA);
    }
    if (x>0 && x<(ALPHA+y))
    {
        x1=ALPHA+y;
        y1=y-MU*(x+1-SIGMA);
    }
    if (x>=ALPHA+y)
    {
        x1=-1.;
        y1=y-MU*(x+1-SIGMA);
    }
}
```

- Valor de las variables utilizadas

TIEMPO	280	MU	0,001	x1	-1.958753
DT	1	SIGMA	-0,1	y1	-3.983966
ALPHA	6				

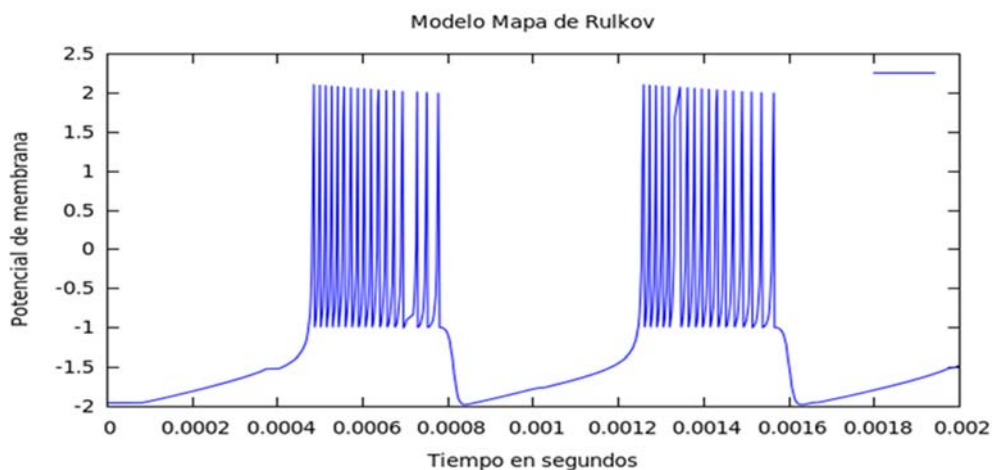


Figura 10.2. Representación de dos ráfagas del modelo del Mapa de Rulkov

10.5. MODELO IZHKEVICH

El código base desarrollado para la implementación del modelo IZHKEVICH, a partir del uso del método matemático de Euler, es el siguiente:

```
for(t=DT; t<=TIEMPO; t+=DT)
{
    if(v>=30)
    {
        v=c;
        u=u+d;
    }

    v1= v+ DT*(0.04*v*v + 5*v + 140 - u + I);
    u1= u+DT*a*(b*v - u);
    v=v1;
    u=u1;
}
```

- **Valor de las variables utilizadas**

TIEMPO	280	u	0.346447	c	-50
DT	0,001	a	0,02	d	2
v	-68.324165	b	0,2	I	10

- **Valores de los parámetros adimensionales a,b,c,d**

Dependiendo de los valores elegidos para los parámetros adimensionales a, b, c y d, el modelo de Izhikevich permite representar diferentes tipos de neuronas, clasificadas por su patrón de disparo. Así, se puede representar el comportamiento de neuronas corticales excitatorias y neuronas corticales inhibitorias. Además, este modelo permite reproducir neuronas tálamo-corticales, que proveen una gran parte de la entrada a las neuronas de la corteza.

A continuación se puede observar la simulación para los distintos comportamientos neuronales que el modelo permite representar, utilizando TIEMPO=280, DT=0,001 e I=10.

Tabla 10.3. Representaciones de los resultados obtenidos de las simulaciones del modelo de Izhikevich para los casos RS, IB y CH correspondientes a neuronas corticales excitatorias

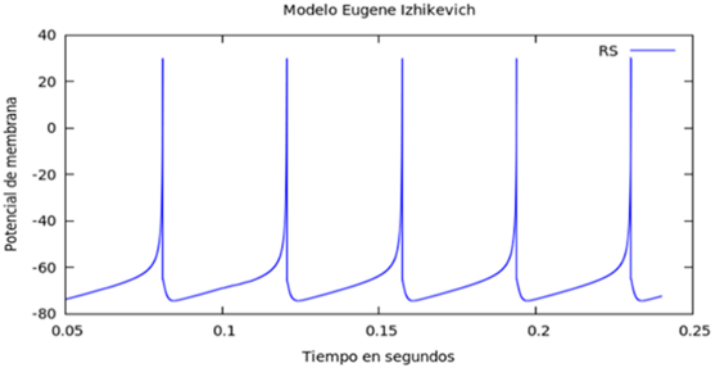
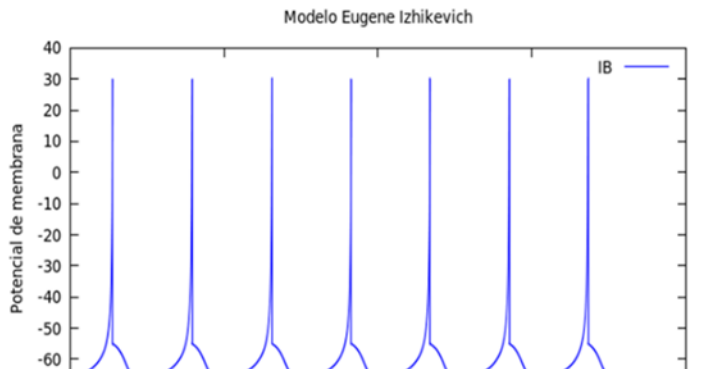
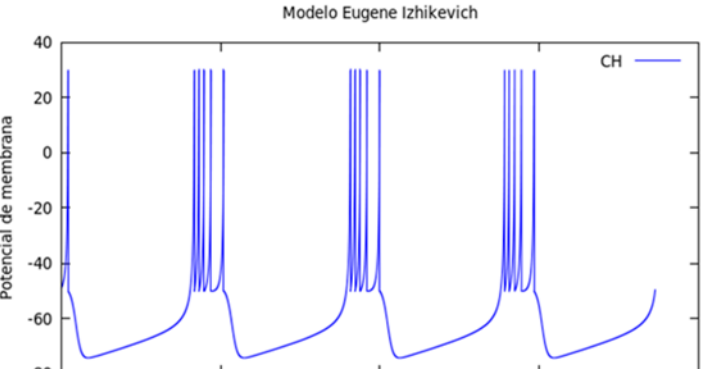
Tipo de neurona	Valor de parámetros	Representación
RS (Regular Spiking)	$a=0,02$ $b=0,2$ $c=-65$ $d=8$	 <p>Modelo Eugene Izhikevich</p> <p>RS</p> <p>Potencial de membrana</p> <p>Tiempo en segundos</p>
IB (Intrinsically Bursting)	$a=0,02$ $b=0,2$ $c=-55$ $d=4$	 <p>Modelo Eugene Izhikevich</p> <p>IB</p> <p>Potencial de membrana</p> <p>Tiempo en segundos</p>
CH (Chattering)	$a=0,02$ $b=0,2$ $c=-50$ $d=2$	 <p>Modelo Eugene Izhikevich</p> <p>CH</p> <p>Potencial de membrana</p> <p>Tiempo en segundos</p>

Tabla 10.4. Representaciones de los resultados obtenidos de las simulaciones del modelo de Izhikevich para los casos FS y LTS a interneuronas corticales inhibitorias

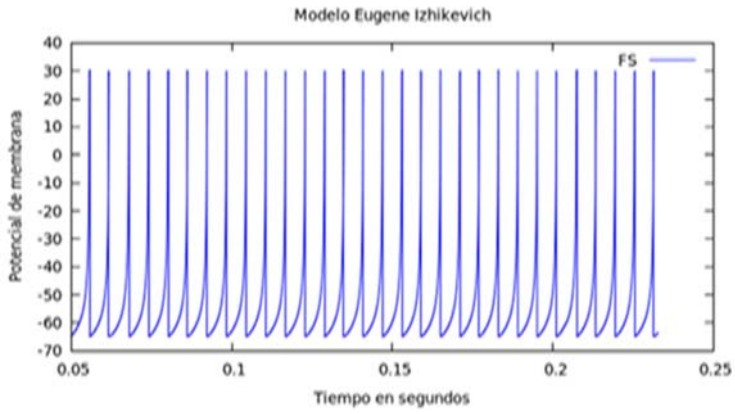
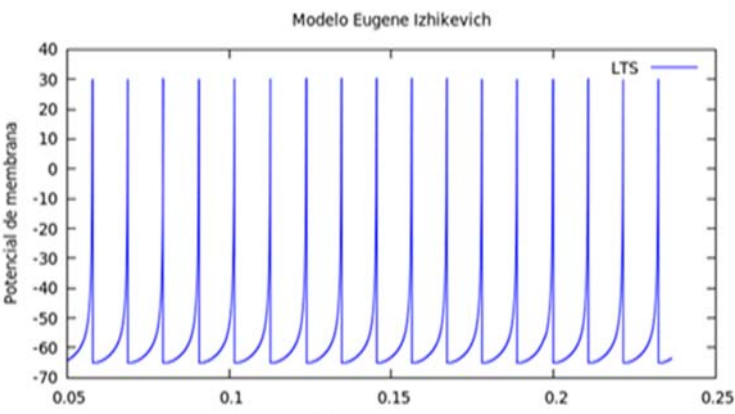
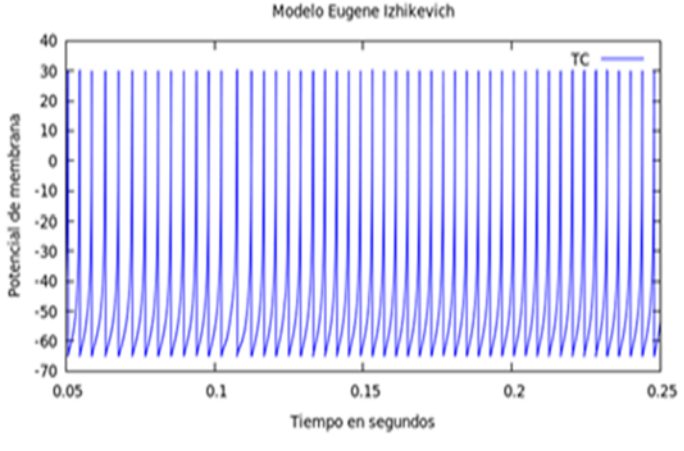
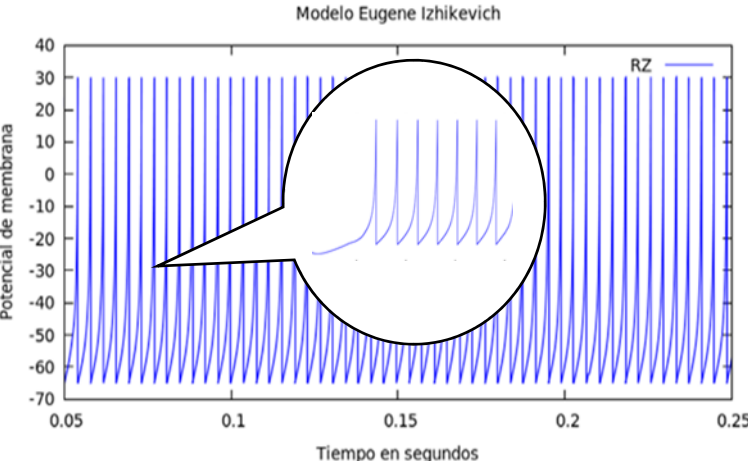
Tipo de neurona	Valor de parámetros	Representación
FS (Fast Spiking)	$a=0,1$ $b=0,2$ $c= -65$ $d=2$	 <p>The graph shows the membrane potential of a Fast Spiking (FS) neuron over a 0.25-second period. The y-axis represents the membrane potential in millivolts, ranging from -70 to 40. The x-axis represents time in seconds, ranging from 0.05 to 0.25. The plot displays a series of sharp, narrow action potentials that rise from a resting potential of approximately -65 mV to a peak of about 30 mV. The spikes are very closely spaced, indicating a high firing rate. The plot is titled 'Modelo Eugene Izhikevich' and includes a legend for 'FS'.</p>
LTS (Low-Threshold Spiking)	$a=0,02$ $b=0,25$ $c= -65$ $d=2$	 <p>The graph shows the membrane potential of a Low-Threshold Spiking (LTS) neuron over a 0.25-second period. The y-axis represents the membrane potential in millivolts, ranging from -70 to 40. The x-axis represents time in seconds, ranging from 0.05 to 0.25. The plot displays a series of action potentials that rise from a resting potential of approximately -65 mV to a peak of about 30 mV. The spikes are wider and more spaced out compared to the FS neuron, indicating a lower firing rate. The plot is titled 'Modelo Eugene Izhikevich' and includes a legend for 'LTS'.</p>

Tabla 10.5. Representaciones de los resultados obtenidos de las simulaciones del modelo de Izhikevich para Neuronas tálamo-corticales

Tipo de neurona	Valor de parámetros	Representación
TC (Thalamo-Cortical)	$a=0,02$ $b=0,25$ $c=-65$ $d=0,05$	
RZ (Resonator)	$a=0,1$ $b=0,25$ $c=-65$ $d=0,05$	

11. ANEXO B

11.1. FUNCIONES EXTERNAS

En el desarrollo del código para cada modelo se han utilizado las siguientes funciones **gettimeofday** y **nanosleep**, correspondientes a las mediciones del tiempo y esperas.

Para describir estas funciones se ha recurrido a la información literal, obtenida de la sección correspondiente del Manual del Programador de Linux (MAN).

11.1.1. FUNCIÓN GETTIMEOFDAY()

Se utiliza la función **gettimeofday()** para calcular el tiempo que tardan los modelos en generar cada punto. (SVr4, BSD 4.3)

NOMBRE

gettimeofday, settimeofday - pone u obtiene la hora.

SINOPSIS

```
#include<time.h>
#include <unistd.h>
int gettimeofday(struct timeval *tv, struct timezone *tz);
```

DESCRIPCIÓN

Gettimeofday puede obtener la hora como una zona horaria.

- **tv** es una estructura **timeval**, tal como se especifica en /usr/include/sys/time.h:

```
struct timeval
{
    long tv_sec;    /* segundos */
    long tv_usec;  /* microsegundos */
};
```

- **tz** es una estructura **timezone** :

```
struct timezone
{
    int tz_minuteswest; /* minutos al O de Greenwich */
    int tz_dsttime;    /* tipo de corrección horaria invierno/verano */
};
```

VALOR DEVUELTO

`gettimeofday` devuelve 0 en caso de éxito ó -1 si ocurre un fallo (en cuyo caso *errno* toma un valor apropiado).

ERRORES

- EINVAL
La zona horaria (o algo más) es inválida.
- EFAULT
Uno de *tv* o *tz* apuntaba afuera de su espacio de direcciones accesible.

Información tomada de:

<http://es.tldp.org/Paginas-manual/man-pages-es-1.28/man2/gettimeofday.2.html>

11.1.2. FUNCIÓN NANOSLEEP()

Para realizar las esperas o retrasos necesarios entre cada momento de comunicación con la tarjeta de adquisición se ha utilizado la función **nanosleep()** POSIX.1b (antes, POSIX.4)

NOMBRE

Nanosleep-Esta función realiza una pausa dela ejecución en un determinado tiempo.

SINOPSIS

```
#include <time.h>
int nanosleep(const struct timespec *req, struct timespec *rem);
```

DESCRIPCIÓN

Retarda la ejecución del programa durante al menos el tiempo especificado en **req*. La función puede regresar antes si se ha mandado una señal al proceso. En este caso, devuelve -1, pone *errno* a **EINTR**, y escribe el tiempo restante en la estructura apuntada por *rem* a menos que *rem* sea **NULL**. El valor de **rem* puede emplearse para llamar a **nanosleep** de nuevo y completar la pausa especificada.

ESTRUCTURA

La estructura *timespec* se emplea para especificar intervalos de tiempo con precisión de nanosegundo. Se especifica en `<time.h>` y tiene la forma

```
struct timespec
{
    time_t tv_sec;    /* segundos */
    long tv_nsec;    /* nanosegundos */
};
```

El valor del campo de nanosegundos debe estar en el rango de 0 a 999 999 999.

Comparado con `sleep(3)` y `usleep(3)`, `nanosleep` tiene la ventaja de no afectar a ninguna señal, está normalizado por POSIX, proporciona una resolución del temporizador mayor, y permite que un 'sleep' que ha sido interrumpido por una señal continúe más fácilmente.

ERRORES

En caso de un error o excepción, la llamada al sistema **nanosleep** devuelve -1 en vez de 0 y pone en *errno* uno de los valores siguientes:

- **EINTR**

La pausa ha sido interrumpida por una señal no bloqueante que ha sido mandada al proceso. El tiempo restante de sueño ha sido escrito en **rem* de modo que el proceso pueda llamar fácilmente de nuevo a `nanosleep` para continuar así con la pausa.

- EINVAL

El valor en el campo *tv_nsec* no estaba en el rango de 0 a 999 999 999 ó *tv_sec* era un número negativo.

FALLOS

La implementación actual de **nanosleep** está basada en el mecanismo normal del temporizador del núcleo, que tiene una resolución de $1/HZ$ s (i.e., 10 ms en Linux/i386 y 1 ms en Linux/Alpha). Por lo tanto, **nanosleep** hace una pausa siempre de al menos el tiempo especificado, empero puede tardar hasta 10 ms más hasta que el proceso sea de nuevo ejecutable. Por la misma razón, el valor devuelto en **rem* en el caso de una señal enviada, se redondea normalmente al siguiente múltiplo más grande de $1/HZ$ s.

Como algunas aplicaciones requieren pausas mucho más precisas (p. ej., para controlar algún hardware que requiere respuestas en tiempo real), **nanosleep** también es capaz de pausas cortas de alta precisión. Si el proceso se planifica bajo una política de tiempo real como *SCHED_FIFO* o *SCHED_RR*, entonces se harán pausas de hasta 2 ms como las esperas ocupadas con precisión de microsegundo.

Información tomada de:

<http://es.tldp.org/Paginas-manual/man-pages-es-1.28/man2/nanosleep.2.html>

Para llevar a cabo las paradas del sistema en los modelos implementados en este proyecto se ha diseñado la función:

```
void espera(double tEspera)
{
    struct timespec tmp;
    tmp.tv_sec = 0;
    tmp.tv_nsec = tEspera*1e6;
    nanosleep(&tmp, NULL);
}
```

Esta función se encarga de hacer las llamadas a `nanosleep()`, pasándole el tiempo que se desea detener el sistema en nanosegundos.

12. ANEXO D

12.1. PRESUPUESTO

Ejecución Material	
Compra de ordenador personal (Software incluido).....	700,00 €
Tarjeta de adquisicion de datos.....	2.156,00 €
Amplificador.....	1.912,00 €
Muestras vivas.....	100,00 €
Total de ejecucion material.....	4.868,00 €
Gastos generales	
16 % sobre Ejecución Material.....	778,88 €
Beneficio Industrial	
6 % sobre Ejecución Material.....	292,08 €
Honorarios Proyecto	
1320 horas a 8€ / hora.....	10.560,00 €
Material fungible	
Gastos de impresión.....	70,00 €
Encuadernación.....	150,00 €
Subtotal del presupuesto	
Subtotal Presupuesto.....	16.718,96 €
I.V.A. aplicable	
21% Subtotal Presupuesto.....	3.510,98 €
TOTAL PRESUPUESTO	
Total Presupuesto.....	20.229,94 €

Madrid, Febrero de 2016
Fdo.: Jenniffer Cubillos Martínez
Ingeniero de Telecomunicación

12.2. PLIEGO DE CONDICIONES

Este documento contiene las condiciones legales que guiarán la realización, en este proyecto, **Modelos de neuronas artificiales en software para su uso en preparaciones de electrofisiología**. En lo que sigue, se supondrá que el proyecto ha sido encargado por una empresa cliente a una empresa consultora con la finalidad de realizar dicho sistema. Dicha empresa ha debido desarrollar una línea de investigación con objeto de elaborar el proyecto. Esta línea de investigación, junto con el posterior desarrollo de los programas está amparada por las condiciones particulares del siguiente pliego.

Supuesto que la utilización industrial de los métodos recogidos en el presente proyecto ha sido decidida por parte de la empresa cliente o de otras, la obra a realizar se regulará por las siguientes:

12.3. CONDICIONES GENERALES

1. La modalidad de contratación será el concurso. La adjudicación se hará, por tanto, a la proposición más favorable sin atender exclusivamente al valor económico, dependiendo de las mayores garantías ofrecidas. La empresa que somete el proyecto a concurso se reserva el derecho a declararlo desierto.
2. El montaje y mecanización completa de los equipos que intervengan será realizado totalmente por la empresa licitadora.
3. En la oferta, se hará constar el precio total por el que se compromete a realizar la obra y el tanto por ciento de baja que supone este precio en relación con un importe límite si este se hubiera fijado.
4. La obra se realizará bajo la dirección técnica de un Ingeniero Superior de Telecomunicación, auxiliado por el número de Ingenieros Técnicos y Programadores que se estime preciso para el desarrollo de la misma.
5. Aparte del Ingeniero Director, el contratista tendrá derecho a contratar al resto del personal, pudiendo ceder esta prerrogativa a favor del Ingeniero Director, quien no estará obligado a aceptarla.
6. El contratista tiene derecho a sacar copias a su costa de los planos, pliego de condiciones y presupuestos. El Ingeniero autor del proyecto autorizará con su firma las copias solicitadas por el contratista después de confrontarlas.
7. Se abonará al contratista la obra que realmente ejecute con sujeción al proyecto que sirvió de base para la contratación, a las modificaciones autorizadas por la superioridad o a las órdenes que con arreglo a sus facultades le hayan comunicado por escrito al Ingeniero Director de obras siempre que dicha obra se haya ajustado a los preceptos de los pliegos de condiciones, con arreglo a los cuales, se harán las modificaciones y la valoración de las diversas unidades sin que el importe total pueda exceder de los presupuestos aprobados. Por

consiguiente, el número de unidades que se consignan en el proyecto o en el presupuesto, no podrá servirle de fundamento para entablar reclamaciones de ninguna clase, salvo en los casos de rescisión.

8. Tanto en las certificaciones de obras como en la liquidación final, se abonarán los trabajos realizados por el contratista a los precios de ejecución material que figuran en el presupuesto para cada unidad de la obra.
9. Si excepcionalmente se hubiera ejecutado algún trabajo que no se ajustase a las condiciones de la contrata pero que sin embargo es admisible a juicio del Ingeniero Director de obras, se dará conocimiento a la Dirección, proponiendo a la vez la rebaja de precios que el Ingeniero estime justa y si la Dirección resolviera aceptar la obra, quedará el contratista obligado a conformarse con la rebaja acordada.
10. Cuando se juzgue necesario emplear materiales o ejecutar obras que no figuren en el presupuesto de la contrata, se evaluará su importe a los precios asignados a otras obras o materiales análogos si los hubiere y cuando no, se discutirán entre el Ingeniero Director y el contratista, sometiéndolos a la aprobación de la Dirección. Los nuevos precios convenidos por uno u otro procedimiento, se sujetarán siempre al establecido en el punto anterior.
11. Cuando el contratista, con autorización del Ingeniero Director de obras, emplee materiales de calidad más elevada o de mayores dimensiones de lo estipulado en el proyecto, o sustituya una clase de fabricación por otra que tenga asignado mayor precio o ejecute con mayores dimensiones cualquier otra parte de las obras, o en general, introduzca en ellas cualquier modificación que sea beneficiosa a juicio del Ingeniero Director de obras, no tendrá derecho sin embargo, sino a lo que le correspondería si hubiera realizado la obra con estricta sujeción a lo proyectado y contratado.
12. Las cantidades calculadas para obras accesorias, aunque figuren por partida alzada en el presupuesto final (general), no serán abonadas sino a los precios de la contrata, según las condiciones de la misma y los proyectos particulares que para ellas se formen, o en su defecto, por lo que resulte de su medición final.
13. El contratista queda obligado a abonar al Ingeniero autor del proyecto y director de obras así como a los Ingenieros Técnicos, el importe de sus respectivos honorarios facultativos por formación del proyecto, dirección técnica y administración en su caso, con arreglo a las tarifas y honorarios vigentes.
14. Concluida la ejecución de la obra, será reconocida por el Ingeniero Director que a tal efecto designe la empresa.
15. La garantía definitiva será del 4% del presupuesto y la provisional del 2%.
16. La forma de pago será por certificaciones mensuales de la obra ejecutada, de acuerdo con los precios del presupuesto, deducida la baja si la hubiera.

17. La fecha de comienzo de las obras será a partir de los 15 días naturales del replanteo oficial de las mismas y la definitiva, al año de haber ejecutado la provisional, procediéndose si no existe reclamación alguna, a la reclamación de la fianza.
18. Si el contratista al efectuar el replanteo, observase algún error en el proyecto, deberá comunicarlo en el plazo de quince días al Ingeniero Director de obras, pues transcurrido ese plazo será responsable de la exactitud del proyecto.
19. El contratista está obligado a designar una persona responsable que se entenderá con el Ingeniero Director de obras, o con el delegado que éste designe, para todo relacionado con ella. Al ser el Ingeniero Director de obras el que interpreta el proyecto, el contratista deberá consultarle cualquier duda que surja en su realización.
20. Durante la realización de la obra, se girarán visitas de inspección por personal facultativo de la empresa cliente, para hacer las comprobaciones que se crean oportunas. Es obligación del contratista, la conservación de la obra ya ejecutada hasta la recepción de la misma, por lo que el deterioro parcial o total de ella, aunque sea por agentes atmosféricos u otras causas, deberá ser reparado o reconstruido por su cuenta.
21. El contratista, deberá realizar la obra en el plazo mencionado a partir de la fecha del contrato, incurriendo en multa, por retraso de la ejecución siempre que éste no sea debido a causas de fuerza mayor. A la terminación de la obra, se hará una recepción provisional previo reconocimiento y examen por la dirección técnica, el depositario de efectos, el interventor y el jefe de servicio o un representante, estampando su conformidad el contratista.
22. Hecha la recepción provisional, se certificará al contratista el resto de la obra, reservándose la administración el importe de los gastos de conservación de la misma hasta su recepción definitiva y la fianza durante el tiempo señalado como plazo de garantía. La recepción definitiva se hará en las mismas condiciones que la provisional, extendiéndose el acta correspondiente. El Director Técnico propondrá a la Junta Económica la devolución de la fianza al contratista de acuerdo con las condiciones económicas legales establecidas.
23. Las tarifas para la determinación de honorarios, reguladas por orden de la Presidencia del Gobierno el 19 de Octubre de 1961, se aplicarán sobre el denominado en la actualidad "Presupuesto de Ejecución de Contrata" y anteriormente llamado "Presupuesto de Ejecución Material" que hoy designa otro concepto.

12.4. CONDICIONES PARTICULARES

La empresa consultora, que ha desarrollado el presente proyecto, lo entregará a la empresa cliente bajo las condiciones generales ya formuladas, debiendo añadirse las siguientes condiciones particulares:

1. La propiedad intelectual de los procesos descritos y analizados en el presente trabajo, pertenece por entero a la empresa consultora representada por el Ingeniero Director del Proyecto.
2. La empresa consultora se reserva el derecho a la utilización total o parcial de los resultados de la investigación realizada para desarrollar el siguiente proyecto, bien para su publicación o bien para su uso en trabajos o proyectos posteriores, para la misma empresa cliente o para otra.
3. Cualquier tipo de reproducción aparte de las reseñadas en las condiciones generales, bien sea para uso particular de la empresa cliente, o para cualquier otra aplicación, contará con autorización expresa y por escrito del Ingeniero Director del Proyecto, que actuará en representación de la empresa consultora.
4. En la autorización se ha de hacer constar la aplicación a que se destinan sus reproducciones así como su cantidad.
5. En todas las reproducciones se indicará su procedencia, explicitando el nombre del proyecto, nombre del Ingeniero Director y de la empresa consultora.
6. Si el proyecto pasa la etapa de desarrollo, cualquier modificación que se realice sobre él, deberá ser notificada al Ingeniero Director del Proyecto y a criterio de éste, la empresa consultora decidirá aceptar o no la modificación propuesta.
7. Si la modificación se acepta, la empresa consultora se hará responsable al mismo nivel que el proyecto inicial del que resulta el añadirla.
8. Si la modificación no es aceptada, por el contrario, la empresa consultora declinará toda responsabilidad que se derive de la aplicación o influencia de la misma.
9. Si la empresa cliente decide desarrollar industrialmente uno o varios productos en los que resulte parcial o totalmente aplicable el estudio de este proyecto, deberá comunicarlo a la empresa consultora.
10. La empresa consultora no se responsabiliza de los efectos laterales que se puedan producir en el momento en que se utilice la herramienta objeto del presente proyecto para la realización de otras aplicaciones.
11. La empresa consultora tendrá prioridad respecto a otras en la elaboración de los proyectos auxiliares que fuese necesario desarrollar para dicha aplicación industrial, siempre que no

haga explícita renuncia a este hecho. En este caso, deberá autorizar expresamente los proyectos presentados por otros.

12. El Ingeniero Director del presente proyecto, será el responsable de la dirección de la aplicación industrial siempre que la empresa consultora lo estime oportuno. En caso contrario, la persona designada deberá contar con la autorización del mismo, quien delegará en él las responsabilidades que ostente.