



Escuela Superior de Ingeniería
Programa de Doctorado en Fabricación, Materiales e Ingeniería
Ambiental

TESIS DOCTORAL

**Diseño e implementación de una red neuromórfica
adaptativa para la generación y control de
movimiento robótico bioinspirado**

Design and implementation of an adaptive neuromorphic network
for the generation and control of bio-inspired robotic locomotion.

Autor

Pablo López Osorio

Directores

Dr. Fernando Pérez Peña y Dr. Juan Pedro Domínguez Morales

Cádiz, 2024

Conformidad de los Directores

D. Fernando Pérez Peña, Profesor Titular de Universidad del Departamento de Ingeniería en Automática, Electrónica, Arquitectura y Redes de Computadores de la Universidad de Cádiz, y Juan Pedro Domínguez Morales, Profesor Titular de Universidad del Departamento de Arquitectura y Tecnología de Computadores de la Universidad de Sevilla, siendo Directores de la Tesis titulada Diseño e implementación de una red neuromórfica adaptativa para la generación y control de movimiento robótico bioinspirado, realizada por el doctorando D. Pablo López Osorio dentro del Programa de Doctorado Fabricación, Materiales e Ingeniería Ambiental, para proceder a los trámites conducentes a la presentación y defensa de la tesis doctoral arriba indicada, informan que se autoriza la tramitación de la tesis.

Los directores de tesis

Fernando Pérez Peña

Juan Pedro Domínguez morales

En Cádiz, España, a 4 de Marzo de 2024

"That's the truth. I promise you: you'll be warm again."

Agradecimientos

En el umbral de este logro académico, me detengo con profunda gratitud para expresar de la mejor forma que pueda mis más sinceros agradecimientos a quienes han contribuido de manera significativa a la realización de esta Tesis Doctoral.

En primer lugar, quiero dar las gracias a mis directores de Tesis, Fernando y Juan Pedro, cuya orientación y apoyo fueron fundamentales en cada etapa de este viaje. Su dedicación y compromiso con mi crecimiento académico han sido invaluable, trascendiendo el simple rol de directores para convertirse incontables veces en psicólogos y amigos. Gracias.

No puedo pasar por alto el apoyo incondicional de mi familia y amigos. A mi familia, a veces refugio de la tormenta, a veces la tormenta misma, les estoy eternamente agradecido por su constante aliento y apoyo moral, por ser una fuente de constante inspiración y por creer en mí cuando ni yo mismo lo hacía. A mis amigos, por ayudarme a desconectar cuando necesitaba hacerlo, incluso cuando no era consciente de esa necesidad. Gracias.

A mis colegas y compañeros del grupo de sistemas neuromórficos Fernando Quintana, Lukas Solhbach, Jannis Kress y René Harmann, por las enriquecedoras reuniones que han servido para apoyarnos y ayudarnos mutuamente con los problemas que íbamos encontrando en nuestro camino. Gracias.

Por otro lado, quiero extender un especial agradecimiento al profesor Luis Mariscal, quien me brindó una invaluable oportunidad de trabajar codo a codo con él en el Laboratorio de Telecomunicaciones Marinas durante más de tres años. Su conocimiento y experiencia enriquecieron significativamente mi aprendizaje y desarrollo académico, siendo funda-

mental en mi formación como investigador. A través de su mentoría, he adquirido perspectivas y habilidades que de otra manera hubieran sido difíciles de obtener. Gracias.

Finalmente, además de expresar mi más sincero agradecimiento, deseo dedicar esta Tesis Doctoral a tres compañeros de viaje los cuales sin su apoyo no estaría escribiendo estas palabras hoy:

En primer lugar a Cati, mi pareja, por acompañarme ya durante más de media vida y por ser siempre un faro de luz en la tormenta. Has sufrido este camino tanto como yo, y nunca has soltado mi mano en cada paso del mismo. Gracias.

En segundo lugar a mi hermano Álvaro, que sin darse cuenta ha sido un pilar de disciplina y determinación para mí. Aunque seas mi hermano menor, me has servido como ejemplo de perseverancia ininidad de veces. Gracias.

Por último, y aunque pueda parecer raro, a mi perro Togo, por estar absolutamente todos los días de este camino a mi lado, recordándome la importancia de salir a pasear cuando la investigación me abrumaba, y esperando pacientemente cuando lograba avanzar en mis estudios durante una tarde. Me has salvado la vida de todas las formas posibles en las que un perro puede salvar a un ser humano. Gracias.

Resumen

La presente Tesis Doctoral se centra en el análisis, diseño, simulación e implementación de un modelo que imite los principios de las estructuras neuronales biológicas responsables de la locomoción: los Generadores de Patrones Centrales (CPGs). Esta Tesis se sitúa en el campo de la ingeniería neuromórfica, una disciplina que surgió a finales de los años ochenta con el objetivo de desarrollar dispositivos electrónicos basados en la neurona como unidad fundamental, capaces de aplicar modelos que aprovechen las características de los sistemas neuronales biológicos.

Aunque la ingeniería neuromórfica es un campo bien establecido, hasta la fecha no se han realizado investigaciones que integren simultáneamente sensores analógicos y neuromórficos para controlar y ajustar el comportamiento robótico en tiempo real. En esta Tesis, se presenta un sistema que emplea *redes neuronales pulsantes* (SNNs) implementadas en la plataforma neuromórfica *SpiNNaker*. Este sistema recopila información de sensores analógicos de presión (FSRs) y sensores digitales (retina neuromórfica), permitiendo así modificar el comportamiento de la estructura neuronal y, por ende, la velocidad y dirección de un robot según la estabilidad del terreno y la actividad visual detectada en su entorno.

Para lograrlo, se propone una estructura neuronal artificial basada en el modelo *Leaky Integrate-and-Fire* (LIF), inspirada en el funcionamiento básico de los CPGs. Esta estructura se implementa primero en el simulador neuromórfico *Brian 2* para evaluar su viabilidad. Posteriormente, se traslada al hardware neuromórfico de *SpiNNaker*, y se comparan los resultados entre el simulador y el hardware para verificar su similitud. Una vez confirmada su efectividad, se utiliza una FPGA para monitorizar la salida de la estructura neuronal y ajustar su comportamiento externa-

mente a *SpiNNaker*, controlando así los actuadores mediante la señal de salida de la estructura neuronal.

Finalmente, mediante una serie de experimentos, se valida que la estructura neuronal desarrollada cumple con los objetivos planteados en esta Tesis, proporcionando una plataforma capaz de adaptar su comportamiento según los parámetros externos detectados en su entorno.

Abstract

The present Doctoral Thesis focuses on the analysis, design, simulation, and implementation of a model that mimics the principles of biological neural structures responsible for locomotion: Central Pattern Generators (CPGs). This Thesis is situated in the field of neuromorphic engineering, a discipline that emerged in the late eighties with the aim of developing electronic devices based on the neuron as a fundamental unit, capable of applying models that leverage the characteristics of biological neural systems.

Although neuromorphic engineering is a well-established field, to date, there have been no research efforts integrating analog and neuromorphic sensors simultaneously to control and adjust robotic behavior in real-time. In this Thesis, a system is presented that employs *Spiking Neural Networks* (SNNs) implemented on the *SpiNNaker* neuromorphic platform. This system collects information from analog pressure sensors (FSRs) and digital sensors (neuromorphic retina), thus allowing modification of the neural structure's behavior and consequently the speed and direction of a robot based on terrain stability and visual activity detected in its environment.

To achieve this, an artificial neural structure based on the *Leaky Integrate-and-Fire* (LIF) model is proposed, inspired by the basic functioning of CPGs. This structure is first implemented in the *Brian 2* neuromorphic simulator to evaluate its feasibility. Subsequently, it is transferred to the *SpiNNaker* neuromorphic hardware, and the results between the simulator and the hardware are compared to verify their similarity. Once its effectiveness is confirmed, an FPGA is used to monitor the output of the neural structure and adjust its behavior externally to *SpiNNaker*, thus controlling the actuators through the neural structure's output signal.

Finally, through a series of experiments, it is validated that the developed

neural structure fulfills the objectives set forth in this thesis, providing a platform capable of adapting its behavior according to external parameters detected in its environment.

Índice general

I Prólogo

1	Introducción	1
1.1	Motivación y definición del problema	3
1.2	Objetivos	5
1.3	Hipótesis de investigación	5
1.4	Estructura de la tesis	6

II Estado del Arte

2	Sistemas neuroinspirados	11
2.1	Introducción	12
2.2	Ingeniería Neuromórfica	14
2.3	Hardware neuromórfico	15
2.3.1	BrainScaleS	15
2.3.2	SpiNNaker	16
2.3.3	Neurogrid/Braindrop	18
2.3.4	TrueNorth	19
2.3.5	Loihi	21
2.3.6	DYNAP-SE	22
2.3.7	ODIN	24
2.3.8	Darwin	24
3	Locomoción biológica	27
3.1	Introducción	27

3.2	Áreas de Brodmann involucradas en la locomoción biológica . . .	28
3.2.1	Corteza premotora	28
3.2.2	Corteza motora primaria	30
3.2.3	Área somatosensorial primaria	31
3.2.4	Corteza somatosensorial secundaria	32
3.3	Estructuras subcorticales	32
3.3.1	Ganglios basales	33
3.3.2	Cerebelo	34
3.3.3	Tálamo	35
3.3.4	Tronco encefálico	35
3.3.5	Regiones locomotoras mesencefálica y subtalámica . . .	36
3.3.6	Sistema piramidal y extrapiramidal	37
3.4	Médula espinal	38
3.5	Músculos	38
3.6	Información propioceptiva	40
4	Neuronas biológicas VS artificiales	41
4.1	Introducción	41
4.2	Anatomía, funcionamiento y sinapsis de las neuronas biológicas.	42
4.2.1	Anatomía y funcionamiento básico	42
4.2.2	Sinapsis neuronal biológica	46
4.3	Analogía entre neuronas biológicas y artificiales	55
5	Redes neuronales pulsantes	59
5.1	Introducción	59
5.2	Fundamentos de las SNN	60
5.3	Modelos neuronales pulsantes	62
5.3.1	Modelo HH	63
5.3.2	Modelo Izhikevich	65
5.3.3	Modelo Adaptive Exponential Integrate-and-Fire (AdEx)	66
5.3.4	Modelo Leaky Integrate and Fire (LIF)	67
5.4	Redes Central Pattern Generator	69
5.4.1	Inspiración biológica	70

Índice general

5.4.2	El CPG bioinspirado	72
5.5	Redes Winner-Take-All	73

III Materiales

6	Herramientas software en ingeniería neuromórfica	77
6.1	Introducción	77
6.2	jAER	79
6.3	PyNN	81
6.4	Brian2	81
7	El protocolo AER	83
7.1	Introducción	83
7.2	Principio de funcionamiento	84
8	Actuadores Dynamixel AX-12A	89
8.1	Introducción	89
8.2	Características	91
8.3	Consideraciones previas	92
8.4	Protocolo de comunicación	93
8.4.1	Comunicación Half-Duplex	93
9	Chip neuromórfico SpiNNaker	97
9.1	Introducción	97
9.2	El chip SpiNNaker	98
9.3	Modelo SpiNN-5	100
9.4	Modelo SpiNN-3	102
10	Sensor de visión neuromórfico	105
10.1	Introducción	105
10.2	Características técnicas	106
10.3	Principio de funcionamiento	108

IV Métodos

11 Métodos	113
11.1 Introducción	113
11.2 Fase I	114
11.2.1 Modelo neuronal asociado	115
11.2.2 Diseño de la red neuronal	116
11.3 Fase II	122
11.3.1 Optimización de la red neuronal	122
11.3.2 Optimización de los parámetros neuronales	123
11.3.3 Sistema digital	124
11.3.4 Modulación por frecuencia de pulsos (PFM)	127
11.4 Fase III	130
11.4.1 Ajuste fino de los parámetros neuronales	130
11.4.2 Diseño de la red Winner-Take-All	133
11.4.3 Implementación del sensor DVS en el sistema	135
11.4.4 Implementaciones finales en la plataforma robótica	138

V Resultados

12 Análisis de resultados	143
12.1 Introducción	143
12.2 Fase I	144
12.2.1 Análisis del CPG_{AB} en Brian 2	144
12.2.2 Análisis de la red de realimentación en Brian 2	146
12.2.3 Análisis del efecto del sensor sobre la arquitectura neuronal en Brian 2	150
12.3 Fase II	156
12.3.1 Implementación de los nuevos parámetros neuronales en Brian 2	156
12.3.2 Implementación de la red adaptativa en SpiNNaker	157
12.3.3 Comparación de los resultados obtenido en Brian 2 y SpiNNaker	160
12.3.4 Evaluación de los modelos de sCPG con datos reales del FSR	160

Índice general

12.4 Fase III	165
12.4.1 Subred adaptativa del FSR	168
12.4.2 Subred adaptativa del DVS	172

VI Epílogo

13 Discusión	179
14 Trabajo futuro	183

VII Anexos

A Neuromorphic adaptive spiking CPG towards bio-inspired locomotion	187
B A feedback sensor based on spiking neural networks for real-time robot adaption	202
C A Neuromorphic Vision and Feedback Sensor Fusion Based on Spiking Neural Networks for Real-Time Robot Adaption	214
Glosario de siglas	230
Referencias	235

Índice de tablas

4.1	Comparativa de las propiedades entre redes neuronales biológicas, ANN y SNN	57
7.1	Requisitos temporales del intercambio de cuatro fases.	86
8.1	Configuración mínima de un paquete de instrucción del protocolo Dynamixel.	94
8.2	Configuración mínima de un paquete de estado del protocolo Dynamixel.	95
11.1	Valores de los pesos sinápticos de la red de adaptación.	121
11.2	Parámetros neuronales utilizados en la Fase II.	123
11.3	Diferencia en los parámetros neuronales entre la fase II y la fase III para el CPG_{AB} y P_{Ref}	131
11.4	Diferencia en los parámetros neuronales entre la fase II y la fase III para el CPG_{EI}	132
11.5	Parámetros neuronales de la red WTA implementada en SpiNNaker.	134
11.6	Pesos sinápticos de la red neuronal adaptativa completa.	139

Índice de figuras

2.1	Representación de la plataforma BrainScaleS.	16
2.2	Imagen de la plataforma SpiNNaker.	17
2.3	Imagen de la plataforma Neurogrid.	19
2.4	Imagen de la plataforma NS16e TrueNorth.	20
2.5	Imagen de la plataforma Loihi 2.	22
2.6	Imagen de la plataforma Dynap-SE.	23
2.7	Imagen de la plataforma Darwin.	25
3.1	Áreas de Brodmann involucradas en la locomoción biológica	29
3.2	Componentes de la estructura y función muscular	39
4.1	Esquema general de la composición básica de una neurona biológica	43
4.2	Imagen al microscopio de una neurona	45
4.3	Esquema de conexión de una neurona	47
4.4	Representación de dos neuronas en los instantes previos a una sinapsis química	49
4.5	Representación de dos neuronas en el momento de la sinapsis química	49
4.6	Representación de dos neuronas en el momento de la sinapsis eléctrica	51
4.7	Representación temporal del potencial de membrana de una neurona biológica.	54
4.8	Analogía entre una neurona biológica y una neurona artificial.	56

5.1	Propiedades temporales del potencial de acción de una neurona pulsante.	61
5.2	Modelos neuronales pulsantes. Plausibilidad biológica VS Coste de implementación	62
5.3	Propiedades temporales del potencial de acción de una neurona pulsante.	69
5.4	Ejemplo de conectividad de una red WTA	74
6.1	Representación de eventos AER en jAER.	80
7.1	Transmisión de información pulsante usando el protocolo AER.	84
7.2	Intercambio de cuatro fases del protocolo AER.	86
8.1	Actuador Dynamixel AX-12A	91
8.2	Esquema del protocolo Dynamixel.	94
8.3	Tiempo entre bytes de un paquete del protocolo Dynamixel.	96
9.1	Chip neuromórfico SpiNNaker.	98
9.2	Imagen gráfica del chip neuromórfico SpiNNaker.	99
9.3	Esquema general del modelo SpiNN-5 de SpiNNaker.	101
9.4	Esquema general del modelo SpiNN-3 de SpiNNaker.	102
10.1	Chip neuromórfico Tmpdiff128.	106
10.2	Contorno de una mano captada por el DVS en el software jAER.	107
10.3	Esquemático de un píxel del DVS y su principio de operación.	109
10.4	DVS con su lente // Chip Tmpdiff128.	110
10.5	Mapeo de un píxel del sensor Tmpdiff128	110
11.1	Topología de la red neuronal adaptativa en la fase I.	115
11.2	Topología del CPG_{AB} junto con la corriente de estímulo.	117
11.3	Topología del CPG_{12}	118
11.4	Esquema de la conversión de los datos extraídos del FSR en frecuencia Poisson.	119
11.5	Esquema ilustrativo del comportamiento de la red adaptativa cuando $f_{Ref} \ll f_{CPG_{AB}}$	120

Índice de figuras

11.6	Esquema ilustrativo del comportamiento de la red adaptativa cuando $f_{Ref} \gg f_{CPG_{AB}}$	120
11.7	Esquema completo de la red adaptativa, junto con el ID de las conexiones sinápticas.	121
11.8	Valores frecuenciales medios, máximos y mínimos obtenidos para cada número de neuronas por población.	122
11.9	FPGA Nexys 4 DDR.	124
11.10	Ejemplo de modulación PWM.	126
11.11	Ejemplo de modulación PFM.	127
11.12	Esquema del módulo principal implementado en FPGA en la fase II.	129
11.13	Esquema de la red WTA.	133
11.14	Esquema de funcionamiento del sensor de visión dinámico.	135
11.15	Distribución de los píxeles del DVS para la selección de dirección.	136
11.16	Esquema del módulo principal implementado en FPGA para la comunicación DVS-SpiNNaker.	137
11.17	Topología completa de la red neuronal adaptativa, incluyendo la adaptación del FSR y del DVS.	138
11.18	Esquema del módulo principal implementado en FPGA para la comunicación de los sensores y SpiNNaker.	139
12.1	Topología del CPG_{AB} junto con la corriente de estímulo.	145
12.2	Simulación del CPG_{AB} con un valor de I_{St} de $10nA$	146
12.3	Simulación del CPG_{AB} con un valor de I_{St} de $2,2nA$	146
12.4	Conexionado del CPG_{AB} y CPG_{12} para su simulación.	147
12.5	Simulación del CPG_{12} en la que se le ha inyectado corriente constante a P_1	148
12.6	Simulación del CPG_{12} en la que se le ha inyectado corriente constante a P_2	149
12.7	Simulación del CPG_{12} en la que se le ha inyectado corriente constante a P_2	150
12.8	Adaptación del CPG frente a una entrada puntual extrema.	151

12.9 Adaptación del CPG en Brian 2 frente a una entrada puntual extrema.	153
12.10 Adaptación del CPG frente a entradas aleatorias.	154
12.11 Adaptación del CPG frente a entradas ascendentes.	155
12.12 Simulación en Brian 2 con los nuevos parámetros neuronales.	156
12.13 Simulación en Brian 2 de entradas límites con los nuevos parámetros neuronales.	157
12.14 Simulación en SpiNNaker del CPG_{AB}	158
12.15 Adaptación del CPG en SpiNNaker frente a una entrada puntual extrema.	159
12.16 Adaptación del CPG en SpiNNaker frente a una entrada creciente.	159
12.17 Adaptación del CPG en SpiNNaker frente a una entrada aleatoria.	160
12.18 Comparación de los resultados obtenidos en Brian 2 y SpiNNaker.	161
12.19 Salida del FSR en diferentes superficies	162
12.20 Setup del experimento realizado para obtener valores del FSR.	163
12.21 Salida del CPG en Brian 2 cuando el FSR recoge valores en terreno arenoso.	164
12.22 Salida del CPG en Brian 2 cuando el FSR recoge valores en madera.	164
12.23 Salida del CPG en SpiNNaker cuando el FSR recoge valores en terreno arenoso.	165
12.24 Salida del CPG en SpiNNaker cuando el FSR recoge valores en madera.	165
12.25 Frecuencia generada por P_{Ref} en arena y madera.	166
12.26 Transición de terrenos en Brian 2.	166
12.27 Transición de terrenos en SpiNNaker.	167
12.28 Comportamiento de la red ante una entrada mínima del FSR.	168
12.29 Comportamiento de la red ante una entrada intermedia del FSR.	169
12.30 Comportamiento de la red ante una entrada máxima del FSR.	170
12.31 Comportamiento de la red ante una entrada variable del FSR.	171
12.32 Comportamiento de la red WTA ante eventos generados en la región izquierda.	173

Índice de figuras

12.33	Comportamiento de la red WTA ante eventos generados en la región central.	174
12.34	Comportamiento de la red WTA ante eventos generados en la región derecha.	175
12.35	Comportamiento de la red WTA ante eventos generados aleatoriamente.	176



Prólogo

1

Introducción

“El paso más importante que puede dar alguien, no es el primero, ¿verdad? Es el siguiente. Siempre el siguiente paso.”

Brandon Sanderson
Juramentada

A lo largo de la historia, los seres humanos han buscado resolver artificialmente los problemas que la naturaleza ya había resuelto, con el propósito de aprovechar esos mecanismos para avanzar y simplificar la realización de tareas peligrosas o complejas. Gracias a esta curiosidad por lo desconocido, se ha impulsado el desarrollo de soluciones inteligentes para conseguir adaptarse a entornos hostiles para los humanos.

Este impulso ha permitido generar nuevos campos de investigación centrados en el procesamiento de información proveniente del entorno a través de diversos sensores, dando lugar a aplicaciones que confieren autonomía a dispositivos artificiales, permitiendo la automatización de procesos y el desarrollo de dispositivos más precisos y eficientes para

resolver problemas complejos. Un claro ejemplo de estos avances se observa en la industria, donde se han desarrollado robots capaces de realizar tareas peligrosas, precisas y repetitivas mediante la incorporación de inteligencia artificial, resultando especialmente útil en situaciones donde la intervención humana sería cara o poco práctica. Hoy en día, es cada vez más común encontrar dispositivos tecnológicos en nuestra vida cotidiana que cuentan con inteligencia incorporada para facilitar tareas que anteriormente requerían intervención humana.

En la historia reciente de la inteligencia artificial, ha surgido un nuevo paradigma de aprendizaje y procesamiento automático inspirado en el sistema nervioso: las redes neuronales artificiales (ANN). Estos sistemas interconectados de neuronas procesan información de manera conjunta para producir respuestas específicas, y se han vuelto ampliamente utilizados en diversas aplicaciones, como el análisis de texto, reconocimiento de voz y patrones, el control motor o la predicción y clasificación [1].

En relación con estos avances, ha emergido un campo de investigación denominado Ingeniería Neuromórfica, el cual se centra en estudiar y replicar los sistemas de procesamiento neural del cerebro humano [2, 3]. Este campo busca desarrollar sistemas capaces de imitar los patrones operativos de las neuronas, utilizando sensores y sistemas bioinspirados para lograr una alta velocidad de transferencia y procesamiento de datos, al tiempo que se profundiza en la comprensión del comportamiento cerebral.

En la literatura actual, se encuentran diversos ejemplos de sensores y sistemas bioinspirados, como sensores de visión y auditivos [4, 5, 6], sistemas de control motor [7, 8, 9], redes neuronales pulsantes (SNN) y fusiones de sensores, entre otros, que están siendo explorados en este campo de investigación.

Esta tesis se centra en el procesamiento neuroinspirado de los Generadores Centrales de Patrones (CPG) y en la fusión de sensores ana-

1.1. Motivación y definición del problema

lógicos y neuromórficos para la generación y control de locomoción robótica. Este tipo de procesamiento se realiza normalmente en dos fases bien diferenciadas: se implementa la estructura neuronal en una plataforma neuromórfica y se analizan sus salidas para generar el movimiento en base a esas salidas. En este capítulo se aborda la motivación que ha llevado a la realización de esta tesis doctoral sobre el diseño e implementación de una red neuromórfica adaptativa utilizando redes neuronales pulsantes para la generación y control de movimiento robótico bioinspirado. También se describen los objetivos e hipótesis de la investigación, el método utilizado y se presenta la estructura del documento.

1.1 Motivación y definición del problema

El campo de estudio en el que se enmarca esta tesis doctoral es el campo de la ingeniería neuromórfica. El objetivo de este campo es diseñar y construir sistemas electrónicos que se inspiren en la estructura y el funcionamiento de los sistemas nerviosos biológicos. Estos sistemas neuromórficos buscan replicar la eficiencia y la capacidad de procesamiento de los sistemas biológicos para realizar tareas específicas, como el reconocimiento de patrones, el aprendizaje automático, la percepción sensorial y el control de robots autónomos. Dado que estos sistemas trabajan bajo el principio de funcionamiento de las neuronas biológicas, las cuales transmiten información a otros agentes (músculos, células sensoriales, glándulas, otras neuronas, etc.) a través de señales electroquímicas, es necesario el uso de circuitos electrónicos que funcionen de manera similar a los sensores biológicos que poseen los seres vivos. Debido a esta necesidad, Los investigadores en este campo han creado diversos dispositivos sensoriales que operan de manera similar a sus homólogos biológicos, como las cócleas o las retinas neuromórficas [4, 10, 6, 11]. Sin embargo, a día de hoy aun existen discrepancias entre estos sensores y los chips neuromórficos capaces de simular el sistema nervioso biológico, principalmente debido a que la creación de sistemas neuroins-

pirados es altamente compleja.

Es por ello que con esta Tesis Doctoral se pretende aportar el diseño de una arquitectura de actuación basada en *Redes Neuronales Pulsantes* (*Spiking Neural Networks*) que, gracias a la incorporación de distintos sensores analógicos y neuromórficos, permita recabar información del entorno y modificar el funcionamiento de los motores que generan el movimiento. Además, se desarrollarán pautas de adaptación en la plataforma para que, en función de los parámetros obtenidos por los sensores, se modifique el comportamiento de la red neuronal y, en consecuencia, la plataforma se adapte de manera eficiente a las condiciones del entorno en el que se encuentra. Con esta investigación se pretende, en primer lugar, aportar un mayor entendimiento de la locomoción biológica aplicada en la robótica y su implementación en una plataforma real. En segundo lugar (y no por ello menos importante), realizar el camino contrario e intentar mejorar el entendimiento de la robótica aplicada a la locomoción biológica ya que, ¿por qué no podríamos soñar con una hipotética conexión entre el sistema nervioso humano y un sistema artificial que le permitiera mejorar, recuperar o incluso ampliar la capacidad locomotora? Esta es la principal motivación para esta Tesis Doctoral: aportar una nueva vía de conocimiento sobre cómo funcionan los mecanismos del cerebro que se encargan de gestionar la locomoción biológica, así como en explorar la aplicación de este conocimiento en el diseño de una plataforma robótica que se inspire en estos principios.

Para llevar a cabo esta tarea, se pretende emular el funcionamiento de las redes neuronales biológicas que generan el movimiento en los seres vivos: los *Generadores Centrales de Patrones* (CPGs). Existen varias posibilidades de implementar un CPG: usando osciladores acoplados, usando (ANN) o SNNs. De entre las mencionadas, la implementación más cercana a la biología se basa en el uso de SNNs, las cuales se basan en modelos biológicos de neuronas plausibles y conexiones sinápticas que implementan características biológicas.

1.2 Objetivos

El objetivo principal de esta tesis doctoral es desarrollar una plataforma robótica que, mediante distintos actuadores controlados por SNNs, sea capaz de desarrollar movimiento fluido y, gracias a distintos tipos de sensores (tanto analógicos como neuromórficos), permita la adaptación del robot a las distintas condiciones del terreno y el entorno en el que se encuentra. Para ello, se plantean una serie de subobjetivos que son:

- **Objetivo 1 (OBJ1).** Estudiar y analizar los algoritmos bioinspirados que pueden ser implementados en hardware neuromórfico.
- **Objetivo 2 (OBJ2).** El diseño de las arquitecturas de red, donde se especifiquen tanto las entradas como las salidas de esta, así como su topología y dinámica de las neuronas para su uso en control motor.
- **Objetivo 3 (OBJ3).** La simulación en software de las SNNs y las técnicas de adaptación diseñadas, para realizar un estudio sobre las similitudes y diferencias con respecto al hardware.
- **Objetivo 4 (OBJ4).** El uso de las SNNs implementadas en distintas aplicaciones de control motor, ya sean mediante sensores neuromórficos o convencionales, realizando una conversión a eventos.
- **Objetivo 5 (OBJ5).** Implementar el modelo diseñado usando hardware neuromórfico en una plataforma robótica real.

1.3 Hipótesis de investigación

Las hipótesis propuestas en este trabajo con el fin de alcanzar los objetivos anteriormente propuestos, son las siguientes:

- **Hipótesis 1 (H1).** La fusión de sensores clásicos y sensores neuromórficos, junto con la implementación de un CPG bio-inspirado en

una plataforma robótica permite una adaptación de la salida de este de una forma eficiente en cuanto a consumo energético y tiempo de respuesta.

- **Hipótesis 2 (H2).** El uso de un mayor número de sensores similares en todas las extremidades o más de un sensor por cada una de ellas, no supone una mayor adaptación en el terreno, por lo que se puede ajustar el número necesario de estos para obtener la mayor adaptación posible.
- **Hipótesis 3 (H3).** Es posible obtener resultados muy similares tanto en *Spinnaker* como en *Brian2* y en las plataformas basadas en FPGAs.

1.4 Estructura de la tesis

Esta Tesis Doctoral tiene un total de seis capítulos y tres apéndices, divididos de la siguiente forma:

- **Capítulo 2: Estado del Arte**

En este capítulo se abordarán los conceptos fundamentales que sustentan esta tesis, comenzando con una revisión del campo de la ingeniería neuromórfica y los simuladores neuromórficos contemporáneos. Luego, se profundizará en un enfoque más biológico, explorando las partes del sistema nervioso implicadas en la locomoción y el funcionamiento de las redes neuronales biológicas. Después de establecer los principios básicos de las neuronas biológicas, se establecerá una analogía con sus contrapartes artificiales, concluyendo con una discusión sobre los modelos de redes neuronales pulsantes más utilizados en la actualidad y las características de cada uno de ellos.

- **Capítulo 3: Materiales**

En este capítulo se examinarán los componentes, tanto de softwa-

1.4. Estructura de la tesis

re como de hardware, empleados en esta Tesis Doctoral, ofreciendo una detallada explicación de sus principios de funcionamiento y su aplicación específica en el estudio realizado.

■ **Capítulo 4: Métodos**

Este capítulo se enfocará en la descripción y diseño del modelo neuronal vinculado a la plataforma neuromórfica. Asimismo, con el propósito de ilustrar la evolución durante el transcurso de toda la Tesis, se expondrán todas las modificaciones y diseños efectuados en tres fases bien diferenciadas. De esta manera, se brindará una visión ordenada de la metodología empleada para obtener la estructura neuronal definitiva.

■ **Capítulo 5: Resultados**

Continuando con la misma estructura del capítulo anterior, este capítulo presentará de manera sistemática los resultados obtenidos en cada una de las tres fases de esta Tesis. Se ofrecerá una visión clara de cómo los resultados fueron mejorando progresivamente en cada fase.

■ **Capítulo 6: Epílogo**

En el último capítulo se expondrán las conclusiones derivadas de la realización de esta Tesis, además de señalar las posibles direcciones de investigación que permanecen abiertas como áreas para futuros trabajos.

■ **Anexos**

Finalmente, se presentan tres anexos en los que se proporcionan las publicaciones científicas que esta Tesis Doctoral ha producido.



Estado del Arte

2

Sistemas neuroinspirados

*“Elige solo una maestra:
la naturaleza”*

Rembrandt H. Van Rijn

En este capítulo se ofrece una revisión del estado del arte de los sistemas neuroinspirados para dar contexto a las aportaciones que realiza la tesis doctoral. Se revisan los conceptos principales del campo de estudio y se propone una breve visión de la historia y los objetivos que caracterizan a la comunidad científica que hay detrás de esta. A continuación, se detallarán y explicarán las diversas interfaces hardware neuromórficas que se han empleado desde sus inicios hasta la actualidad, destacando las contribuciones significativas que estos dispositivos han aportado al campo científico. Por último, se presentará el protocolo de comunicación utilizado por los dispositivos hardware neuromórficos para establecer conexiones entre ellos.

2.1 Introducción

Desde el inicio de la vida en la Tierra, los seres vivos han ocupado diferentes áreas del planeta mediante una notable capacidad de adaptación. A lo largo del tiempo, esta habilidad para ajustarse a diversos entornos les ha permitido sobrevivir en condiciones muy variadas. Esta capacidad adaptativa es la clave de la gran diversidad de especies existentes hoy en día, cada una con estrategias únicas para asegurar su supervivencia, resultado de cambios en su código genético a través de la evolución.

En 1859, Charles Robert Darwin presentó la teoría de la evolución [12], demostrando como la Tierra alberga una inmensa diversidad de especies que han resuelto eficazmente los desafíos mediante la adaptación.

De hecho, durante siglos, la naturaleza ha sido fuente de inspiración para científicos e ingenieros, llevándolos a explorar cómo emular el comportamiento de diversas formas de vida. Como la bióloga Janine Benyus señala en un artículo para la editorial estadounidense William Morrow [13], "la naturaleza tiene más de 3 800 millones de años de experiencia creando formas de vida que se adaptan a todos los ambientes, por lo que si se busca una solución eficiente, la mejor opción es observarla e intentar replicarla" Esta perspectiva dio lugar a la biomímesis, una disciplina que busca resolver problemas utilizando principios inspirados en la naturaleza.

La biomímesis ha impulsado la creación de sistemas bioinspirados presentes en nuestra vida cotidiana, como submarinos, aviones y robots que imitan la morfología de seres vivos. Además, herramientas como los implantes cocleares [14], los marcapasos [15], la navegación mediante ondas sonoras (SONAR [16]) o el algoritmo de optimización por enjambre de partículas (PSO [17]) utilizan principios bioinspirados en su funcionamiento.

Desde la década de los 80 hemos sido testigos de una revolución en la industria y en nuestra vida diaria, gracias al avance de la electrónica, la

2.1. Introducción

informática y la robótica. Sin embargo, estos sistemas tienen limitaciones en cuanto a variedad de tareas y adaptabilidad, a diferencia de los seres vivos.

En la actualidad, la mayoría de las estructuras robóticas están controladas por sistemas informáticos basados en ordenadores. Aunque estos sistemas son altamente destacados por su velocidad y precisión, además de haber avanzado significativamente en un corto período de tiempo, presentan limitaciones notables. Estas limitaciones se relacionan con la incapacidad de los sistemas informáticos para llevar a cabo tareas que no han sido previamente programadas y para adaptarse a cambios que se produzcan durante el proceso.

A pesar del notable avance de los sistemas de computación clásicos en las últimas dos décadas, el desarrollo en robótica no ha sido tan marcado. Una de las razones principales es que los robots todavía dependen considerablemente de las limitaciones de los sistemas de computación clásicos. En este sentido, surge una pregunta fundamental que los investigadores del campo de la robótica deben plantearse: ¿son realmente los sistemas de computación clásicos la mejor opción para dirigir el comportamiento de los robots o debemos explorar soluciones más eficaces que permitan a los robots desarrollar habilidades cognitivas avanzadas?

La respuesta a esta pregunta puede no ser sencilla, pero está claro que si se desea replicar los mismos logros que la naturaleza ha alcanzado con sus creaciones, es crucial comprender cómo funciona el sistema de control biológico. Para lograr esto, la ingeniería inversa se convierte en un elemento fundamental para recrear artificialmente esta entidad biológica: el sistema nervioso y, en particular, el cerebro.

Los interrogantes planteados por la comunidad científica han dado origen al concepto de *"sistema neuro-inspirado"*. Este enfoque se utiliza, entre otros, para abordar los desafíos en el campo de la robótica al tomar como referencia el principio de funcionamiento del sistema nervioso. La investigación en este área está ganando popularidad gracias a los es-

fuerzos de los llamados ingenieros neuromórficos, quienes se dedican al estudio de cómo codificar y procesar la información de manera análoga a como lo haría el cerebro.

Ante este panorama, una alternativa prometedora es explorar cómo el cerebro lleva a cabo el procesamiento de la información y aplicar esos principios a sistemas diseñados para propósitos específicos en lugar de sistemas de propósito general. En la siguiente sección se abordará el concepto de ingeniería neuromórfica, se examinará brevemente su historia y se detallarán los principios que guían su desarrollo.

2.2 Ingeniería Neuromórfica

La ingeniería neuromórfica es un campo interdisciplinario que fusiona la ingeniería y la neurociencia, enfocándose en el diseño y desarrollo de sistemas electrónicos y computacionales que se inspiran en el funcionamiento del sistema nervioso biológico [18, 19]. Estos sistemas neuromórficos buscan imitar eficazmente ciertas características y procesos de los sistemas neuronales y cerebrales con el propósito de llevar a cabo tareas específicas.

El término "ingeniería neuromórfica" fue utilizado por primera vez por Carver Mead en la década de 1980 en el Caltech [20]. Su principal objetivo era emular el comportamiento de las neuronas presentes en el sistema nervioso mediante el uso de circuitos analógicos VLSI [21]. Gracias a esta iniciativa, en la actualidad es posible abordar los sistemas neuronales biológicos desde una perspectiva tecnológica a través de implementaciones en hardware neuromórfico. Aunque en sus primeros años los dispositivos neuromórficos eran limitados y se encontraban en unos pocos grupos de investigación o laboratorios, en los últimos 25 años han surgido diversas plataformas capaces de simular modelos neuronales con miles de neuronas, las cuales se describen a continuación.

2.3 Hardware neuromórfico

En esta sección se describirán las características más destacadas de los chips neuromórficos basados en CMOS. Aunque la investigación en este campo se remonta a más de tres décadas, la discusión en esta sección se limitará a algunas de las tecnologías más recientes de plataformas neuromórficas a gran escala que son capaces de integrar más de 1 000 neuronas.

2.3.1 BrainScaleS

BrainScaleS (Figura 2.1) es un sistema de hardware neuromórfico resultado de la colaboración de varios grupos de investigación, como la Universidad de Heidelberg (RKUH) y la Universidad Técnica de Dresden (TUD) [22, 23, 24]. La pieza central de BrainScaleS es el Núcleo de Red Analógica (ANC), que se compone de múltiples circuitos de neuronas analógicas y sus sinapsis (en la Figura 2.1 se muestra un diagrama del sistema BrainScaleS a escala completa). La fabricación del ANC se realizó mediante tecnología CMOS de 180 nm, lo que resultó en la creación de una Red Neural Analógica de Alto número de Entradas (HiCANN), la unidad funcional más pequeña posible en esta estructura, que admite un máximo de 512 neuronas pulsantes y aproximadamente 14 000 sinapsis por neurona [22]. No obstante, en una única oblea de 20 cm pueden alojarse hasta 352 chips HiCANN, lo que suponen 4×10^7 sinapsis y hasta 180 000 neuronas.

Los prototipos de la próxima generación de BrainScaleS admiten plasticidad programable a través de procesadores de propósito general usados junto con la circuitería neuromórfica [26]. Estos procesadores tienen acceso a sensores dedicados en las sinapsis que miden el intervalo de tiempo entre los disparos pre y post-sinápticos. Se pueden definir funciones arbitrarias que calculan actualizaciones de los pesos sinápticos a partir de esta información, lo que permite implementar reglas de aprendizaje altamente flexibles en BrainScaleS, incluido el aprendizaje basado

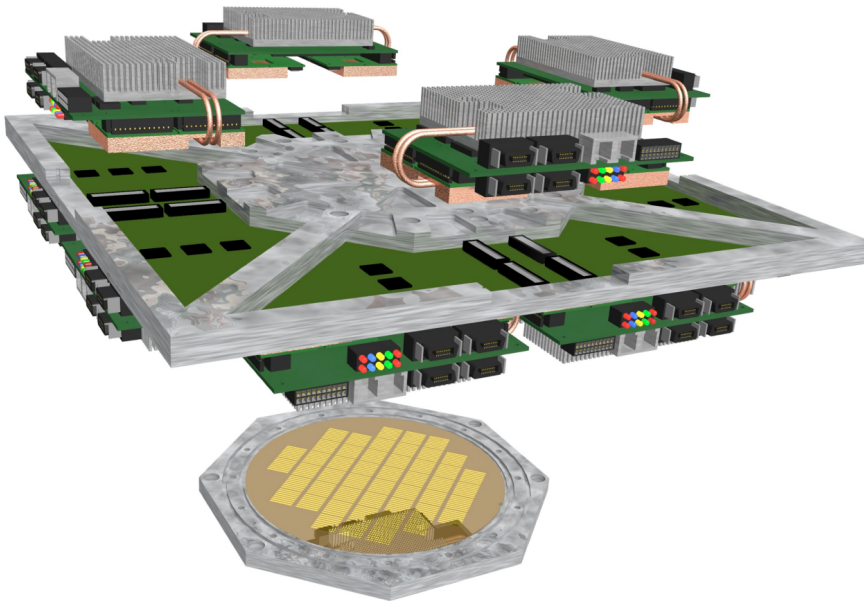


Figura 2.1: Plataforma BrainScaleS. Imagen extraída de [25].

en recompensas. Además, al igual que otras plataformas neuromórficas como SpiNNaker, BrainScaleS utiliza PyNN para permitir al usuario especificar redes pulsantes para su emulación en el hardware [27].

2.3.2 SpiNNaker

SpiNNaker [28] (Figura 2.2) es un sistema digital diseñado para simular eficientemente grandes redes de neuronas que se asemejan en complejidad al cerebro humano en tiempo real. Sus componentes clave son los núcleos ARM9 [29], que tienen acceso a una pequeña cantidad de memoria local y a algo de memoria compartida en un chip multinúcleo, sin una memoria global. La comunicación entre nodos se realiza a través de una interfaz de alto rendimiento optimizada para el enrutamiento de mensajes pequeños (no mayores de 72 bits) con alta eficiencia.

El diseño de SpiNNaker está orientado a eventos y se refleja en su paradigma de manejo de mensajes. Cuando un mensaje llega a un núcleo,

2.3. Hardware neuromórfico

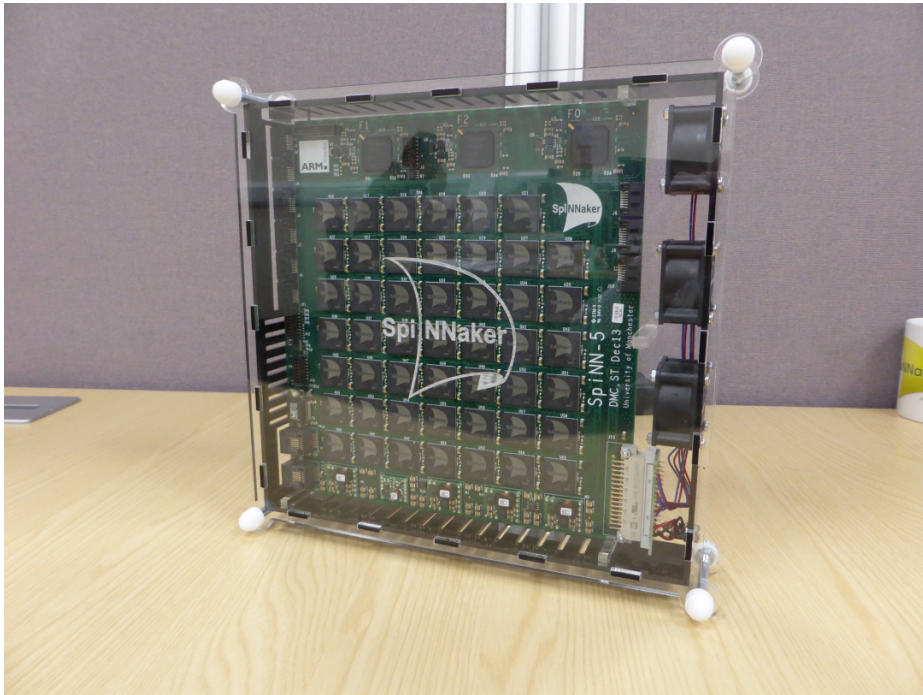


Figura 2.2: Plataforma SpiNNaker, modelo SpiNN-5.

desencadena una interrupción que pone en espera el paquete para su procesamiento por ese núcleo. El sistema está optimizado para manejar paquetes pequeños de manera eficiente, lo que mantiene las colas cortas. De esta manera, SpiNNaker implementa conceptos fundamentales del cerebro, como la computación orientada a eventos, la localización de la información, la alta conectividad de entrada y salida, y la comunicación con mensajes relativamente breves.

Si bien SpiNNaker se puede programar directamente, su *framework* ofrece varios niveles de abstracción. Facilita la implementación de redes de neuronas utilizando PyNN [27], una librería de Python que admite la portabilidad de diseños de redes entre diversos simuladores neuronales y hardware, incluido el sistema BrainScaleS. La interfaz de PyNN en SpiNNaker ofrece varios modelos de neuronas estándar, como LIF y el modelo de sistemas dinámicos de Izhikevich [30], junto con algoritmos comunes de plasticidad sináptica, como STDP [31]. Su sucesor, SpiNNaker2

[32] aumenta el número de núcleos por chip y agrega aceleradores de álgebra lineal, entre otras mejoras, y ha demostrado ser útil en el aprendizaje profundo con conectividad dispersa.

Dado que la plataforma neuromórfica que se empleará a lo largo de todo el desarrollo de esta tesis es SpiNNaker, en los próximos capítulos se proporcionarán más detalles acerca de su estructura, composición y funcionamiento.

2.3.3 Neurogrid/Braindrop

La exploración de sistemas neuromórficos ha llevado al desarrollo de plataformas como NeuroGrid y Braindrop (Figura 2.3). NeuroGrid, diseñada con el propósito principal de implementar modelos neuronales a gran escala y emular su funcionamiento en tiempo real, se caracteriza por una arquitectura que combina recursos de memoria y cómputo bien sincronizados con las señales a procesar, beneficiándose de las características físicas de los transistores CMOS en regímenes subumbrales para emular directamente diversas funciones neuronales y sinápticas [33, 34]. La configuración de NeuroGrid incluye una placa que aloja 16 chips NeuroCore CMOS estándar interconectados en una red jerárquica, con cada chip compuesto por una matriz de 256x256 neuronas de dos compartimentos. Una característica destacada es la capacidad de cada neurona en esta matriz para dirigir sus señales a múltiples destinos a través de una infraestructura digital de enrutamiento en árbol asíncrono. Esta disposición permite que NeuroGrid implemente modelos de redes corticales con hasta un millón de neuronas y miles de millones de conexiones sinápticas, estableciendo conexiones dispersas de largo alcance y una conectividad local densa [35].

En un desarrollo posterior, se ha introducido Braindrop, un núcleo neurosináptico de señal mixta que amplía las posibilidades en la ingeniería neuromórfica. Braindrop consta de 4 096 neuronas y 64 KB de memoria de pesos, diseñado para ser programado a un alto nivel de abstrac-

2.3. Hardware neuromórfico

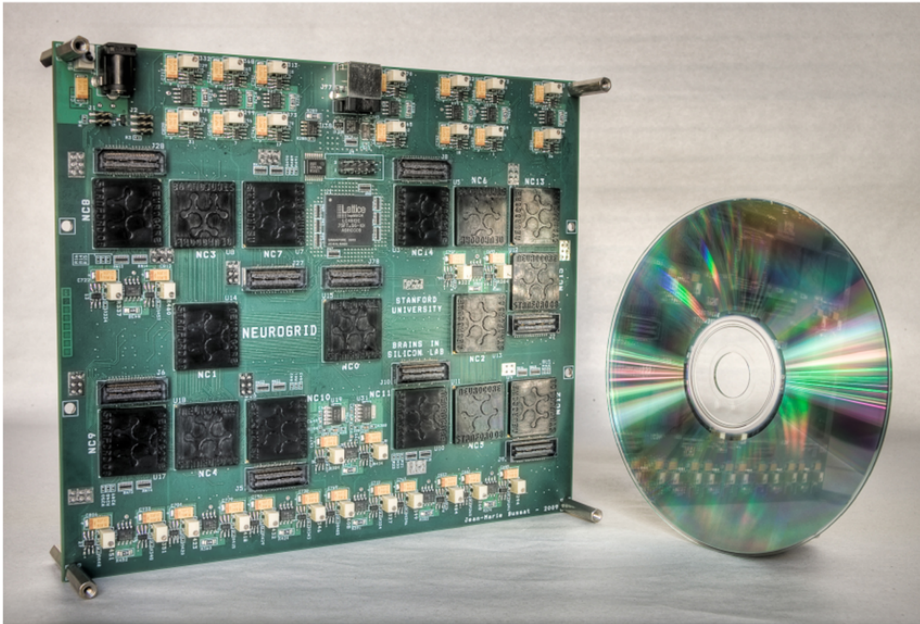


Figura 2.3: Plataforma Neurogrid.

ción, lo cual facilita su uso en comparación con sistemas neuromórficos anteriores que requerían una programación a nivel neurosináptico y un conocimiento experto del hardware. Aprovecha la variabilidad inherente de los circuitos analógicos para realizar cálculos útiles, presentando una vía prometedora en la simulación de modelos neuronales y la realización de cálculos neuromórficos eficientes [33].

2.3.4 TrueNorth

El chip neuromórfico TrueNorth (Figura 2.4) representa un avance significativo en el esfuerzo por emular la eficiencia del cerebro humano en hardware electrónico. Fue diseñado como un procesador neurosináptico en tiempo real con un consumo energético notablemente bajo de 65 mW. La arquitectura de TrueNorth se desvía de las reglas tradicionales de la computación, adoptando un diseño que no se basa en el modelo de *Von Neumann* [36]. Este diseño es reconocido por su baja demanda de energía, alto paralelismo, escalabilidad y tolerancia a defectos, lo que lo hace

distinto a las arquitecturas de computadora tradicionales que siguen el modelo Von Neumann. El chip está compuesto por 4096 núcleos neurosinápticos, albergando en total 1 millón de neuronas digitales y 256 millones de sinapsis, todas estrechamente interconectadas mediante una red de comunicación basada en eventos [37].

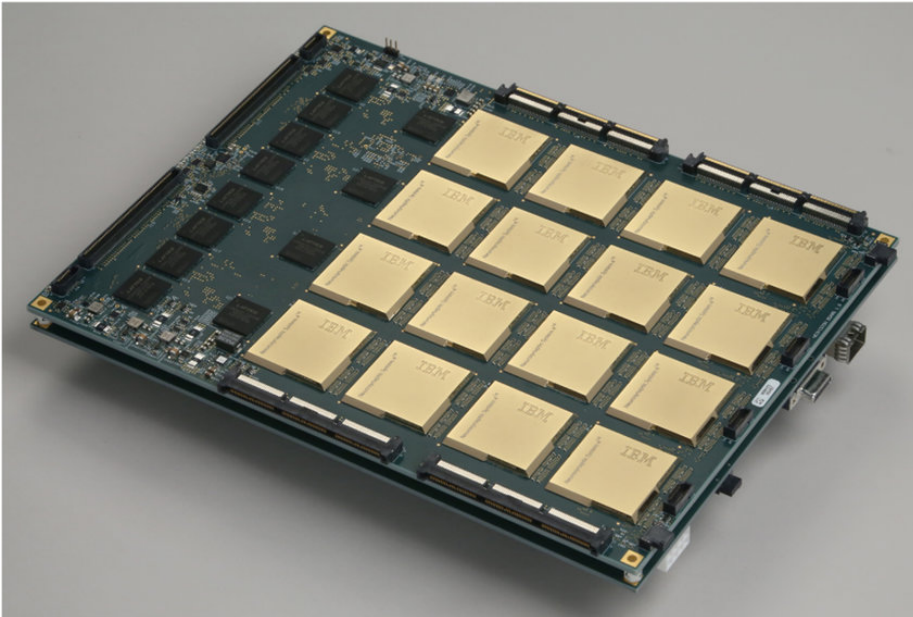


Figura 2.4: Plataforma NS16e, la cual incorpora 16 chips TrueNorth.

TrueNorth utiliza pulsos digitales para llevar a cabo la computación neuromórfica, lo que resulta en un paralelismo de ejecución y una eficiencia energética sobresaliente. A pesar de estas ventajas, se ha identificado que la resolución de cuantificación baja de los pesos sinápticos y los pulsos puede limitar la precisión de la inferencia, por ejemplo, en tareas de clasificación ejecutadas por redes neuronales en el chip [38].

El diseño de TrueNorth se inspiró notablemente en la estructura y funcionamiento del cerebro humano, lo que se refleja en su arquitectura *Non-Von-Neumann*. Esta arquitectura se caracteriza por la modularidad, paralelismo y escalabilidad, lo que contribuye a su eficiencia energética, bajo consumo de energía y volumen compacto [39].

2.3. Hardware neuromórfico

En términos de construcción física, cada chip TrueNorth incorpora $5,4 \times 10^9$ transistores. La fabricación de estos chips se realiza mediante una tecnología de proceso de semiconductores de óxido metálico complementario (CMOS) de 28 nm de baja potencia, con pares complementarios de funciones lógicas, lo que subraya la avanzada ingeniería y la innovación tecnológica involucrada en el desarrollo de TrueNorth [40].

2.3.5 Loihi

El chip Loihi de Intel (Figura 2.5) marca un hito significativo en la modelización de redes neuronales pulsantes, fabricado mediante el proceso de 14 nm de Intel y ocupando un área de 60 mm^2 . Este chip incorpora una variedad de características innovadoras en el ámbito de la computación neuromórfica, como la conectividad jerárquica, compartimientos dendríticos, retrasos sinápticos y, de manera destacada, reglas de aprendizaje sináptico programables [41].

En la búsqueda de desarrollar soluciones neuromórficas inteligentes, se ha creado un emulador para el chip neuromórfico Loihi utilizando el simulador de redes neuronales pulsantes Brian. Este emulador facilita la comprensión de los componentes esenciales del hardware de Loihi, proporcionando una base conceptual sólida que es determinante para el desarrollo de soluciones neuromórficas eficaces [42].

Con la llegada de Loihi 2, la siguiente versión del procesador de investigación neuromórfica de Intel, se ofrece soporte para una amplia variedad de modelos de neuronas impulsadas con dinámicas totalmente programables. Esta mejora facilita el procesamiento eficiente de datos en tiempo real en experimentos de simulación en hardware emulado de Loihi 2 [43].

La implementación y aplicación de Loihi en ámbitos de investigación práctica también ha sido extensamente analizada. Se han introducido y explorado el chip Loihi y sus sistemas de hardware en términos de su aplicación y utilidad en los campos de investigación actuales, señalando



Figura 2.5: Plataforma Loihi 2.

do una tendencia reciente en el hardware de computación neuromórfica [44].

Desde una perspectiva arquitectónica, el chip Loihi se basa en una mailla neuromórfica compuesta por 128 *neurocores*, que ejecutan cálculos neuronales de manera altamente paralela y eficiente en términos energéticos. La gestión de estos *neurocores* es supervisada por 3 procesadores x86 integrados, y una red asíncrona en chip (NoC) facilita la comunicación entre las neuronas. Esta estructura optimiza el consumo energético de Loihi sin sacrificar la rapidez de la comunicación entre las operaciones neuronales [45].

2.3.6 DYNAP-SE

El chip neuromórfico DYNAP-SE (Figura 2.6) es una plataforma analógica neuromórfica diseñada para emular el comportamiento de las redes neuronales biológicas. Este chip pertenece a una familia de procesadores neuromórficos, junto con otros modelos como DYNAP-SE2 y DYNAP-SEL, que han sido desarrollados para propósitos de investigación, en contraste con DYNAP-CNN que está dirigido a aplicaciones comerciales [46].

DYNAP-SE es específicamente una composición de cuatro chips neuro-

2.3. Hardware neuromórfico

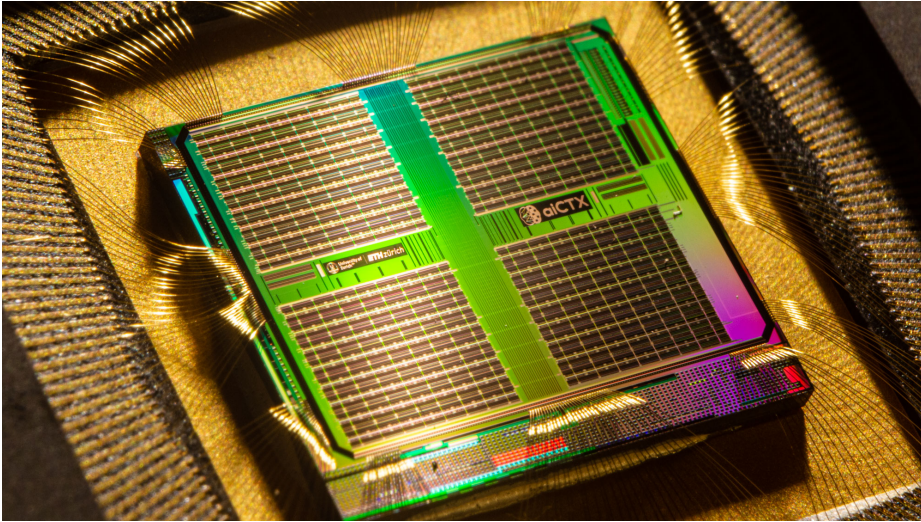


Figura 2.6: Plataforma Dynap-SE.

mórficos, cada uno con 1 000 circuitos de neuronas analógicas. Cada neurona en estos chips tiene un bloque de Memoria de Acceso por Contenido (CAM) que contiene 64 direcciones representando a las neuronas pre-sinápticas con las que está conectada [47]. Este diseño permite la implementación de redes neuronales reconfigurables y en tiempo real, proporcionando una plataforma para desarrollar y validar modelos complejos de procesamiento neural [48].

Dentro de las características del chip DYNAP-SE, se incluyen neuronas adaptativas LIF, una variedad de opciones de sinapsis, conectividad recurrente configurable, y opciones de transmisión en broadcast/multicast. Esto último permite una comunicación eficaz entre diferentes partes de la red neuromórfica emulada [49]. Por último, aunque el DYNAP-SE2 es una versión más reciente de la familia, hereda la eficiencia energética de las generaciones anteriores y añade nuevas características para aplicaciones en tiempo real con baja latencia, utilizando redes *feed-forward* generales, recurrentes y de reserva, lo que lo convierte en una herramienta adecuada para aplicaciones de extremo a extremo en tiempo real [50].

2.3.7 ODIN

ODIN es un procesador digital neuromórfico que ha sido propuesto para permitir la utilización de modelos de neuronas ligeramente más complejo [51]. Este procesador, fabricado con tecnología CMOS de 28 nm, está compuesto por 256 neuronas y 64 000 sinapsis, y destaca por cómo las restricciones de diseño sobre las sinapsis pueden ser liberadas al trasladar la mayoría de los cálculos sinápticos al nivel de la neurona [52].

En términos de diseño, el chip ODIN tiene un área de $0,086 \text{ mm}^2$ y ha sido optimizado para lograr una energía mínima de de 12.7 pJ por cada operación sináptica[53, 54]. Este diseño compacto y eficiente en energía destaca la promesa de los chips neuromórficos en la emulación eficiente de procesos neuronales.

El chip ODIN fue desarrollado por el equipo de ICTEAM en la UCLouvain, en Bélgica [54]. En el núcleo del chip, las neuronas pueden ser configuradas con el modelo LIF o el modelo Izhikevich. Los parámetros de las neuronas se almacenan en una matriz SRAM de 4 KB, mientras que las 64 000 sinapsis están implementadas como una matriz SRAM de 32 KB [45].

2.3.8 Darwin

Darwin NPU (Figura 2.7) es un chip neuromórfico desarrollado como un coprocesador de hardware, cuya arquitectura se basa en SNN. La implementación de este chip se realizó mediante lógica digital, constituyendo inicialmente un sistema monolítico. No obstante, sus creadores establecieron una interfaz de comunicación estandarizada a través del formato AER, lo que posibilita futuras expansiones hacia un sistema distribuido compuesto por múltiples chips, conectados mediante un bus AER [55].

El proyecto Darwin NPU fue concretado por un equipo de investigadores de la Universidad de Zhejiang y la Universidad Hangzhou Dianzi, ambas situadas en Hangzhou, China . Este innovador chip fue concebido para

2.3. Hardware neuromórfico

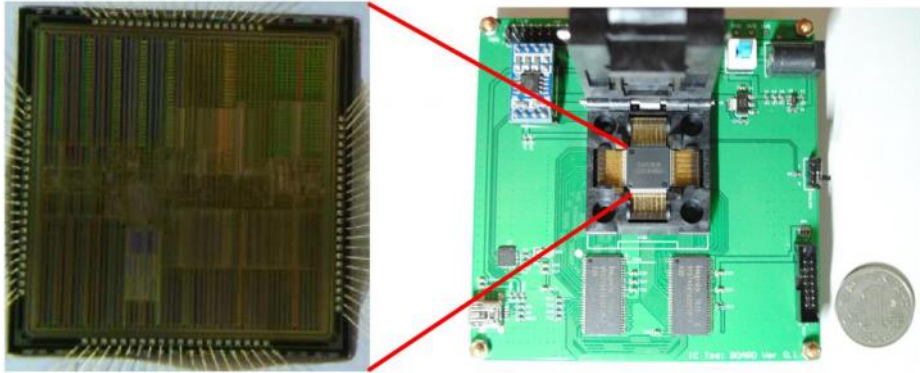


Figura 2.7: Plataforma Darwin NPU.

simular las interacciones entre neuronas y sinapsis que ocurren en el cerebro humano. Con una capacidad para soportar hasta 2 048 neuronas, 4 194 304 sinapsis y 15 posibles retrasos sinápticos, el Darwin NPU se presenta como una plataforma robusta para el estudio y desarrollo de aplicaciones en el campo de la computación neuromórfica [56].

En lo que respecta a las aplicaciones prácticas, el Darwin NPU ha demostrado ser una herramienta versátil. Se han desarrollado diversas aplicaciones que incluyen reconocimiento de gestos, reconocimiento de imágenes, reconocimiento de voz y decodificación de señales de electroencefalograma (EEG). Un logro destacado es la significativa reducción del consumo energético, en al menos dos órdenes de magnitud, al emplear el Darwin NPU, resaltando así la eficiencia energética que aporta esta tecnología neuromórfica en comparación con soluciones tradicionales [55].

3

Locomoción biológica

*“La vida es movimiento;
todo lo que vive, se mueve”*

Leonardo Da Vinci

3.1 Introducción

La locomoción biológica es el resultado de un complejo entramado de interacciones entre estructuras neuronales y musculares, todas orquestadas para permitir el movimiento coordinado y adaptativo en los seres vivos. Este capítulo ofrece una visión general de los componentes del sistema nervioso central involucrados en la locomoción.

Aunque no se abordará cada componente en su máxima profundidad, se profundizará lo necesario para captar los conceptos esenciales que se alinean con los ejes principales de esta tesis, definiendo los componentes biológicos que de una forma u otra afectan a la locomoción.

3.2 Áreas de Brodmann involucradas en la locomoción biológica

Antes de abordar la discusión sobre las áreas de *Brodmann* implicadas en la locomoción, es fundamental entender qué son estas áreas y cómo fueron clasificadas.

Las áreas de *Brodmann* son regiones específicas dentro de la corteza cerebral en el cerebro humano y de otros primates, clasificadas según su citoarquitectura, que se refiere a la estructura histológica y la organización celular [57]. Fueron delineadas por el neurólogo alemán Korbinian Brodmann, quien basó su clasificación en las variaciones observadas en la organización y densidad celular en diferentes regiones de la corteza cerebral [58].

Brodmann segmentó el cerebro en 52 áreas funcionales distintas, cada una con características arquitectónicas únicas en términos de grosor y estratificación cortical, y son ampliamente aceptadas y utilizadas en el campo de la neurofisiología [59, 60]. Además, Brodmann introdujo y elaboró un enfoque combinado evolutivo y citoarquitectónico para definir estas áreas; por ejemplo, la región motor-somatosensorial unificada en insectívoros se descompone en numerosas áreas motoras, sensoriales y de asociación en el cerebro humano [61].

3.2.1 Corteza premotora

La corteza premotora ocupa el área 6 de *Brodmann* (Figura 3.1) y desempeña un papel determinante en varias funciones motoras y cognitivas. Según lo detallado en la Enciclopedia de Neurociencia [62], la región lateral de la corteza premotora se divide en cuatro áreas específicas: F2, F4, F5 y F7. De estas, las áreas F4 y F5 se agrupan comúnmente bajo la denominación de corteza premotora ventral, mientras que F2 y F7 se categorizan como corteza premotora dorsal. Además, se describe a la corteza premotora como un conjunto complejo de áreas interrelacio-

3.2. Áreas de *Brodman* involucradas en la locomoción biológica

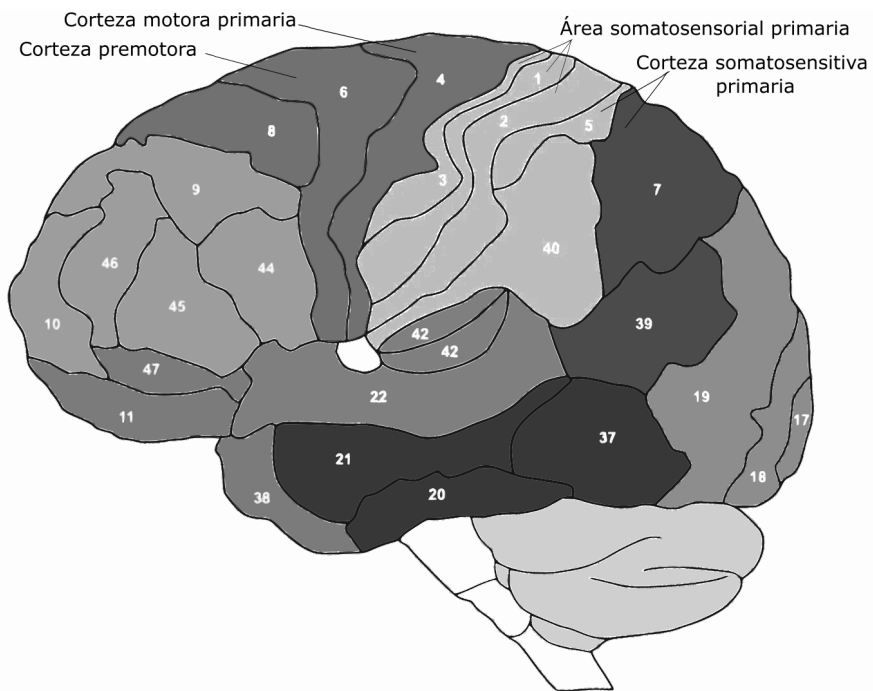


Figura 3.1: Áreas de *Brodman* involucradas en la locomoción biológica.

nadas en el lóbulo frontal, situadas rostralmente a la corteza motora primaria, las cuales contribuyen significativamente a las funciones motoras. En esta región, las neuronas motoras superiores influyen el comportamiento motor a través de conexiones recíprocas extensas con la corteza motora primaria [63].

En cuanto a la percepción de acciones, un estudio reveló que la corteza premotora humana tiene la capacidad de discriminar entre las observaciones de acciones de agarre realizadas por un modelo biológico y uno no biológico, lo que sugiere una afinación biológica que ha evolucionado y persiste en el cerebro humano [64]. Otro estudio destacó que las neuronas en la corteza premotora dorsal codifican el nivel de dificultad durante la selección de objetivos para la toma de decisiones motoras a nivel individual y de población, indicando que la corteza premotora participa en procesos cognitivos de alto nivel asociados con la toma de decisiones y el aprendizaje [65].

Además, se ha investigado la función de la corteza premotora medial en la percepción del ritmo y el tiempo. En el experimento presentado en [66], se observó que las trayectorias de la actividad neural en esta región realizan un ciclo oscilatorio, donde la amplitud de la oscilación es mayor para tempos más lentos en comparación con tempos más rápidos.

3.2.2 Corteza motora primaria

La corteza motora primaria del cerebro ocupa el área 4 de *Brodmann*, y esta desempeña un papel fundamental en la ejecución y adquisición de movimientos especializados. Esta región contiene poblaciones de neuronas que responden a la activación de músculos específicos y otras que codifican la dirección del movimiento independientemente de los músculos involucrados [67]. Los circuitos intrínsecos de la corteza motora son altamente adaptables, y se puede inducir plasticidad sináptica en respuesta a diversos protocolos de estimulación [67].

Además de su papel en la ejecución motora, recientes avances en técnicas de estimulación no invasivas han revelado que la corteza motora primaria también puede estar involucrada en procesos cognitivos superiores [68]. En particular, la corteza motora primaria recibe *inputs* principalmente de la corteza premotora y la corteza secundaria, de la corteza somatosensorial detrás de ella, y del cerebelo a través del tálamo ventral lateral. También recibe señales cutáneas y propioceptivas directamente de los núcleos ventroposteriores del tálamo [67]. Se ha sugerido que la función principal de la corteza motora primaria es controlar los movimientos voluntarios, emergiendo este papel de redes distribuidas en lugar de representaciones discretas, y que en los mamíferos adultos, estas redes son capaces de adaptarse y modificarse en función de las características particulares del individuo. [69].

3.2.3 Área somatosensorial primaria

El área somatosensorial primaria (S1) se encuentra en el giro postcentral, y es esencial para el procesamiento de la información sensorial, incluyendo el tacto, la presión, la temperatura y el dolor, la cual proviene de diversas partes del cuerpo. Esta área es crucial para la integración adecuada de las señales sensoriales y motoras, aspectos fundamentales para la ejecución de movimientos hábiles y coordinados, contribuyendo significativamente a la locomoción biológica [70].

En S1, el área 3 se destaca por ser la primera en recibir las entradas sensoriales aferentes de los receptores de la piel, así como por transmitir esta información a otras áreas somatosensoriales secundarias. Esta transmisión de información sensorial es vital para la coordinación y ejecución de movimientos, lo que resalta la importancia de S1 en la locomoción biológica [71].

Los estudios de estimulación magnética han proporcionado avances sobre la localización de S1, revelando que el área relacionada con las manos en la S1 se encuentra a unos 2 cm lateral a la corteza motora. Esta proximidad y relación espacial entre la corteza somatosensorial y la corteza motora subraya la interacción intrínseca entre la percepción sensorial y la acción motora, elementos centrales en la locomoción biológica [72].

El entendimiento de la organización funcional de la S1 ha evolucionado con el tiempo, desde la conceptualización de *Penfield*, quien a través de la electroestimulación delineó un arreglo somatotópico en el cual diferentes regiones de la S1 corresponden a diferentes partes del cuerpo [73]. Esta organización somatotópica es fundamental para la locomoción, ya que permite una representación precisa del cuerpo en el cerebro, facilitando la integración sensoriomotora necesaria para la locomoción biológica [74].

Recientemente, se ha explorado cómo las distintas capas corticales en el área 3 procesan la información sensorial y cómo este procesamiento

influye en la percepción del error de predicción temporal. Los hallazgos sugieren una modulación diferencial en las capas profundas del área 3, lo que podría reflejar una contribución distinta de estas capas al procesamiento sensoriomotor y, por ende, a la locomoción biológica [71].

3.2.4 Corteza somatosensorial secundaria

La corteza somatosensorial secundaria (SII) se ubica en el banco superior del surco lateral y comprende las áreas de Brodmann 5 y 7 [75]. A diferencia de la corteza somatosensorial primaria (S1), la SII representa niveles más altos de procesamiento de estímulos somatosensoriales, teniendo gran importancia para la interpretación de estos estímulos [76]. En un estudio muy reciente, se hipotetizó que la SII podría ser un sustrato crucial para aquellas modalidades somatosensoriales que la S1 no codifica [77].

Por otro lado, estudios recientes de imagenología en humanos han revelado que la SII está involucrada en procesos que requieren una integración de información de alto nivel, como la autoconciencia, las relaciones sociales, la representación del cuerpo entero y las extrapolaciones metafóricas [78]. Para finalizar, también es importante destacar que la SII juega un papel determinante en el procesamiento del dolor [79].

3.3 Estructuras subcorticales

En la sección anterior se discutieron las áreas de *Brodman* y su papel en la locomoción biológica. Estas regiones corticales son esenciales para coordinar diversos movimientos y funciones motoras. Ahora, se profundizará en las estructuras subcorticales, que, aunque ubicadas debajo de la corteza cerebral, interactúan estrechamente con las áreas corticales para facilitar la locomoción.

Las estructuras subcorticales como los ganglios basales, el cerebelo, el tálamo y el tronco encefálico son esenciales en la modulación y regula-

3.3. Estructuras subcorticales

ción del movimiento. A continuación se explorarán en detalle estas estructuras subcorticales y cómo contribuyen a la locomoción, proporcionando una comprensión más profunda de esta compleja red neuroanatómica.

3.3.1 Ganglios basales

Los ganglios basales, compuestos por el cuerpo estriado (caudado + putamen), el palidum, el núcleo subtalámico y sustancia negra, desempeñan un papel protagonista en diversas funciones motoras, formando parte de circuitos corticosubcorticales paralelos y funcionalmente segregados que respaldan una amplia gama de funciones cerebrales sensorimotoras, cognitivas y emocionales-motivacionales [80]. Una de las funciones primordiales de los ganglios basales es la facilitación de los movimientos voluntarios y la inhibición simultánea de movimientos competidores o interferentes. Además, se involucran en el control de una variedad de comportamientos no motores, que abarcan desde emociones hasta aprendizaje procedimental y memoria de trabajo [81].

En un sentido más específico, los ganglios basales desempeñan un papel en el control motor, particularmente en la modulación de los detalles cinemáticos de las habilidades motoras aprendidas, aunque aún no se comprende completamente cómo contribuyen a especificar la cinemática de estas habilidades motoras [82]. Los mecanismos neurales en los ganglios basales actúan de manera selectiva para eliminar o mejorar la inhibición, permitiendo diferentes combinaciones de señales motoras. Además, controlan las salidas motoras ya sea directamente a través de sus proyecciones a las regiones motoras del mesencéfalo o indirectamente a través de los núcleos talámicos. Además, los ganglios basales también están implicados en el aprendizaje y la selección de los programas motores o conductuales más apropiados, siendo su organización funcional interna muy adecuada para tales mecanismos de selección, tanto en el desarrollo como en la edad adulta. Se discute la posibilidad de que la torpeza pueda atribuirse, al menos en parte, a la disfunción

de los ganglios basales, en el contexto de los roles diferenciales, complementarios o interactivos de los ganglios basales y el cerebelo en el desarrollo del control motor [80, 83].

3.3.2 Cerebelo

El cerebelo participa en la regulación del movimiento motor y el control del equilibrio. Se encarga de coordinar la marcha y mantener la postura, además de controlar el tono muscular y la actividad muscular voluntaria, aunque no puede iniciar la contracción muscular [84].

El cerebelo recibe información de la corteza motora primaria, el córtex sensorial y la médula espinal. Esta información es integrada, procesada y retransmitida a la corteza motora primaria, por lo que se sugiere que el cerebelo tiene un efecto facilitador sobre la excitabilidad de esta [85].

Además, se encuentra fundamentalmente involucrado en la temporización y en el control de la ejecución ordenada y precisa de secuencias motoras. La rápida reacción del cerebelo a las entradas está sustentada por mecanismos celulares específicos que aseguran precisión a escala de milisegundos [86]. Una de las teorías principales sobre el funcionamiento del cerebelo es la del "modelo interno", respaldada por investigadores como Kawato y Wolpert [87], donde proponen que el cerebelo utiliza modelos internos para generar predicciones sobre las consecuencias de los movimientos, permitiendo ajustes rápidos y precisos para alcanzar los objetivos motores.

El cerebelo es esencial para la planificación, ejecución, terminación y aprendizaje de los movimientos, ya que compara el rendimiento motor actual con el rendimiento motor deseado, lo cual es importante en la adaptación del movimiento y la postura. También contribuye a la cognición y el comportamiento, aunque este aspecto se extiende más allá de las funciones motoras [88].

3.3. Estructuras subcorticales

3.3.3 Tálamo

El tálamo, concretamente el tálamo motor, desempeña tareas de modulación y coordinación de las funciones motoras. Tradicionalmente, se ha considerado que el tálamo motor comprende dos conjuntos de núcleos talámicos anteriores. Uno de estos conjuntos recibe inputs del cerebelo e incluye ciertos núcleos ventrolaterales y ventroposterolaterales. El otro conjunto recibe inputs de los ganglios basales e incluye diferentes núcleos ventrolaterales y ventroanteriores [89]. Los núcleos talámicos constituyen un sistema de retransmisión diencefálica entre diferentes sistemas aferentes y el telencéfalo, vinculando las aferencias cerebelosas y palidales con las regiones motoras y premotoras corticales [90].

El tálamo motor recibe aferencias del cerebelo, del globo pálido interno y de la sustancia negra, y proyecta principalmente hacia la corteza motora, la corteza premotora y el área motora suplementaria [91]. Además, el tálamo también funciona como un monitor de las salidas motoras, ya que muchas de las vías ascendentes hacia el tálamo tienen ramas involucradas en el control del movimiento. Por otro lado, la recientemente definida y rica inervación de los núcleos talámicos "superiores" (como el pulvinar) proveniente de células piramidales en la capa cinco de la neocorteza también proviene de ramas de axones descendentes largos que están involucrados en el control del movimiento. [92].

3.3.4 Tronco encefálico

El tronco encefálico trabaja como intermediario de las funciones motoras, actuando como un conducto principal para muchas vías ascendentes y descendentes que son fundamentales para la comunicación entre diferentes partes del sistema nervioso central. Alberga la mayoría de los núcleos de los nervios craneales y contiene las neuronas motoras inferiores para los músculos de la cabeza, lo que implica que juega un papel vital en el procesamiento inicial de las señales aferentes generales que están relacionadas con las funciones motoras [93].

Uno de los aspectos destacados del tronco encefálico es su participación en el control postural. Según observaciones científicas, el control postural implica un mecanismo anticipatorio o de retroalimentación anticipada. Como parte del plan motor para mover el brazo, por ejemplo, el efecto del movimiento inminente sobre la estabilidad corporal se evalúa y se utiliza para generar un cambio en la actividad de los músculos gastrocnemios (más conocidos como gemelos). Este cambio precede y proporciona soporte postural para el movimiento del brazo [94].

Además, el tronco encefálico está involucrado en la función motora, sensorial y espacial, así como en la regulación de la temperatura, la función cardíaca, la función respiratoria y la conciencia. Esta amplia gama de funciones pone de manifiesto la importancia del tronco encefálico en la regulación y coordinación de las funciones motoras y sensoriales [95].

También es notable que los trastornos que afectan a la función motora del tronco encefálico son comúnmente encontrados en la práctica clínica. Los mecanismos del tronco encefálico regulan los movimientos oculares, y hay influencias corticales y cerebelosas en estos circuitos biológicos. Además, se involucra en los mecanismos relacionados con la deglución y el habla [96].

3.3.5 Regiones locomotoras mesencefálica y subtalámica

Las regiones mesencefálica y subtalámica locomotoras desempeñan funciones de modulación y control del comportamiento motor. La región motora mesencefálica actúa como una interfaz entre los sistemas motores de orden superior y las neuronas motoras inferiores. Se compone de un módulo excitatorio que incluye el núcleo pedunculopontino y el núcleo cuneiforme, cuya activación se ha propuesto que provoca diferentes modalidades de movimiento [97]. Esta región, situada en la frontera entre el mesencéfalo y el rombencéfalo, controla la iniciación locomotora, la velocidad locomotora, las transiciones de la marcha y hasta las detenciones locomotoras. En los vertebrados, la región motora mesencefálica

3.3. Estructuras subcorticales

controla la locomoción, y en humanos con enfermedad de *Parkinson*, los déficits en la locomoción están cada vez más asociados con la disminución de la actividad en la región motora mesencefálica [98].

Por otro lado, la región subtalámica también participa en la coordinación de la locomoción. Un estudio mostró que la estimulación eléctrica por parte del subtálamo y el mesencéfalo provoca una locomoción coordinada. Además, se ha evidenciado la existencia de conexiones directas y oligosinápticas entre las regiones locomotoras del subtálamo y el mesencéfalo, lo que indica una interacción funcional entre estas regiones para la modulación de la locomoción [99].

3.3.6 Sistema piramidal y extrapiramidal

Las funciones motoras en el cuerpo humano son coordinadas principalmente por el sistema piramidal y el sistema extrapiramidal.

El sistema piramidal es esencial para la iniciación y control de los movimientos voluntarios, proporcionando el control motor a los músculos de la cabeza, cuello y extremidades [100]. Este sistema se caracteriza por tener tractos que descienden directamente desde la corteza cerebral hacia la médula espinal y de ahí a los músculos esqueléticos, siendo un sistema de control motor directo y preciso.

Por otro lado, el sistema extrapiramidal juega un papel fundamental en la coordinación y modulación de estos movimientos voluntarios, así como en el control de los movimientos involuntarios. Está activamente involucrado en la iniciación y activación selectiva de movimientos voluntarios, y su estructura se distingue del sistema piramidal en que sus fibras pasan a través del tegmento en lugar de la pirámide medular. Además, el sistema extrapiramidal subyace en la organización de varias categorías de sinergias motoras, incluyendo las sinergias oculofaciales, oculocefálicas y faciorespiratorias [101].

3.4 Médula espinal

La médula espinal actúa como conductor de la comunicación entre el cerebro y la periferia del cuerpo. Está organizada por segmentos, con 31 pares de nervios espinales que emanan de ella, facilitando así la transmisión de señales motoras y sensoriales entre el cerebro y el resto del cuerpo [102]. En esta zona es donde se encuentran principalmente las redes CPGs involucradas en movimientos rítmicos como caminar o nadar y se ha observado que, tras una lesión en esta zona, los CPGs biológicos se dañan y resultan difíciles de reparar, lo que evidencia la importancia crítica de la médula espinal en la funcionalidad motora [103].

Aunque la médula espinal es la estación de salida del sistema motor central, aún se sabe poco sobre las relaciones entre su actividad funcional y los parámetros de movimiento voluntario en humanos. En esquemas tradicionales, la médula espinal se asigna una función subordinada durante la producción de movimiento, jugando un papel predominantemente pasivo. Sin embargo, la producción y control de funciones motoras complejas a menudo se atribuyen a estructuras cerebrales centrales como la corteza, los ganglios basales y el cerebelo, lo que implica que la médula espinal trabaja en conjunto con estas estructuras cerebrales para facilitar la función motora [104].

3.5 Músculos

Las funciones motoras de los músculos radican en una gran variedad de movimientos y tareas que realiza el cuerpo humano. Los músculos, en general, se clasifican en tres tipos principales: músculo esquelético, músculo cardíaco y músculo liso, cada uno con funciones, componentes celulares, fisiología y patología únicas [105]. El músculo esquelético, por ejemplo, es fundamental para controlar el movimiento y la postura.

La contracción muscular, que es esencial para la función motora, comienza en la unión neuromuscular, que sirve como punto de conexión entre

3.5. Músculos

el nervio y el músculo. A partir de esta unión, se desencadena un proceso conocido como acoplamiento de excitación-contracción, que es la transducción de la señal eléctrica en una acción mecánica, es decir, la contracción muscular [106]. Este proceso, los resultados que surgen de él y todos los agentes que intervienen en él para llevarlo a cabo pueden observarse en la Figura 3.2 .

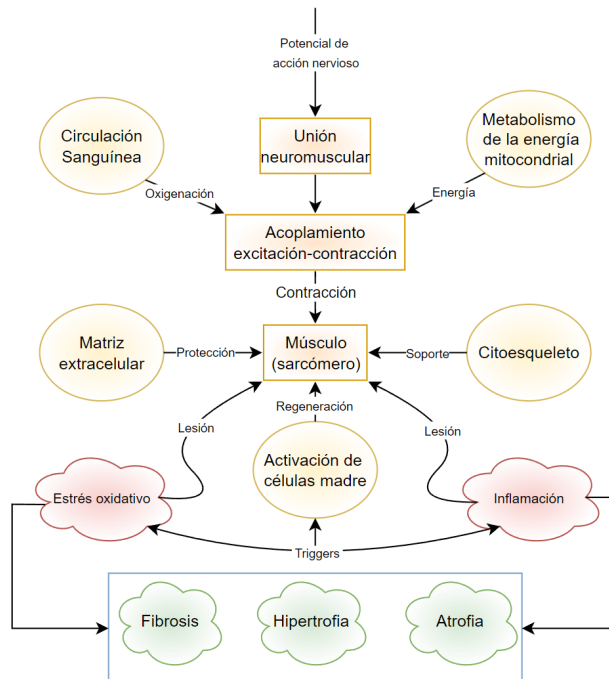


Figura 3.2: Representación esquemática de los diversos componentes funcionales necesarios para la función muscular, o que surgen como consecuencia de ella.

La biomecánica y el control neural del movimiento son áreas de estudio que exploran cómo los músculos interactúan con otros sistemas corporales para realizar tareas motoras complejas como evitar un obstáculo, escalar o saltar. Por ejemplo, se ha investigado cómo las propiedades musculares y los reflejos neurales interactúan durante el aprendizaje motor y la ejecución de tareas, lo que resalta la importancia de examinar la función muscular en vivo durante estas tareas motoras complejas [107].

Además, los músculos y los nervios pueden ser estimulados eléctrica-

mente para generar contracciones musculares y crear movimientos funcionales de las extremidades. Esta estimulación eléctrica va más allá de la simple contracción muscular, implicando la activación motora y sensorial de los músculos para crear estos movimientos funcionales [108].

3.6 Información propioceptiva

La información propioceptiva juega un papel imprescindible en las funciones motoras, actuando como un regulador que modula los *outputs* motores a través de la percepción interna de la posición corporal. Esto es esencial para generar y mantener patrones precisos de movimiento o marcha, especialmente en lo que respecta al control de la locomoción [109].

En los mamíferos, la información propioceptiva se transmite al sistema nervioso central desde el órgano del tendón de Golgi¹ y los husos musculares², siendo estos últimos la principal fuente de retroalimentación propioceptiva para la regulación sensoriomotora espinal y el servocontrol [110].

Además, en trabajos relacionados con la neuromecánica, se ha estudiado cómo las neuronas sensoriales transmiten información propioceptiva desde los husos musculares a las poblaciones de neuronas motoras α^3 , destacando la importancia de la información propioceptiva en la codificación de la tasa y el reclutamiento de neuronas motoras [111].

¹El tendón de Golgi es una estructura especializada que se localiza en el punto de unión de los tendones y los músculos, y miden la tensión que aplica el músculo.

²Los husos musculares son receptores sensoriales incrustados en las fibras musculares que detectan variaciones en la longitud del músculo, siendo modulables mediante las motoneuronas γ para que respondan adecuadamente a los cambios de la longitud del músculo.

³Las neuronas α son las neuronas motoras de mayor tamaño, y su principal función es la de activar la musculatura esquelética.

4

Neuronas biológicas VS artificiales

“El cerebro humano tiene 100 mil millones de neuronas, cada neurona conectada a otras 10 mil neuronas. Sentado sobre sus hombros está el objeto más complicado del universo.”

Michio Kaku

4.1 Introducción

En la cúspide del entendimiento humano y la innovación técnica, la ingeniería neuromórfica se presenta como un puente que une los dominios de la biología y la computación. En el camino hacia la creación de sistemas computacionales que emulan la eficiencia y adaptabilidad del cerebro humano, es imperativo entender las similitudes y diferencias fundamentales entre las neuronas biológicas y las artificiales. Este capítulo se sumerge en una exploración detallada de las neuronas biológicas y artificiales, delineando sus operaciones, características y las interacciones

en sistemas neuroinspirados.

El análisis comparativo entre las neuronas biológicas y artificiales permite no solo apreciar la complejidad inherente y la eficiencia de los sistemas biológicos, sino también destilar principios fundamentales que pueden ser emulados en hardware neuromórfico. A través de esta comparación, se busca entender cómo las arquitecturas neuromórficas pueden ser diseñadas para incorporar las características del mundo biológico con un mínimo consumo de energía, llevando así la ingeniería neuromórfica un paso más cerca de alcanzar su promesa de revolucionar la computación y, en última instancia, desentrañar los misterios del cerebro humano.

4.2 Anatomía, funcionamiento y sinapsis de las neuronas biológicas.

A través de una exploración lo suficientemente meticulosa como para comprender la estructura de las neuronas, esta sección desvelará la anatomía fundamental y la organización funcional de estas, que con su vasta red de conexiones, son los pilares del pensamiento, la memoria y la acción. Con este punto se aspira a proporcionar una comprensión profunda y enriquecedora de cómo estas estructuras microscópicas conforman la esencia de la comunicación y el funcionamiento cerebral.

4.2.1 Anatomía y funcionamiento básico

En este apartado se hará un recorrido por la estructura básica de las neuronas: desde las dendritas, que reciben las señales, pasando por el soma o cuerpo celular que integra la información, hasta el axón, que transmite los impulsos eléctricos y químicos hacia otras células mediante la sinapsis neuronal. Durante el transcurso del mismo se utilizará la Figura 4.1, donde se presenta un esquema general de la estructura básica de una neurona biológica.

4.2. Anatomía, funcionamiento y sinapsis de las neuronas biológicas.

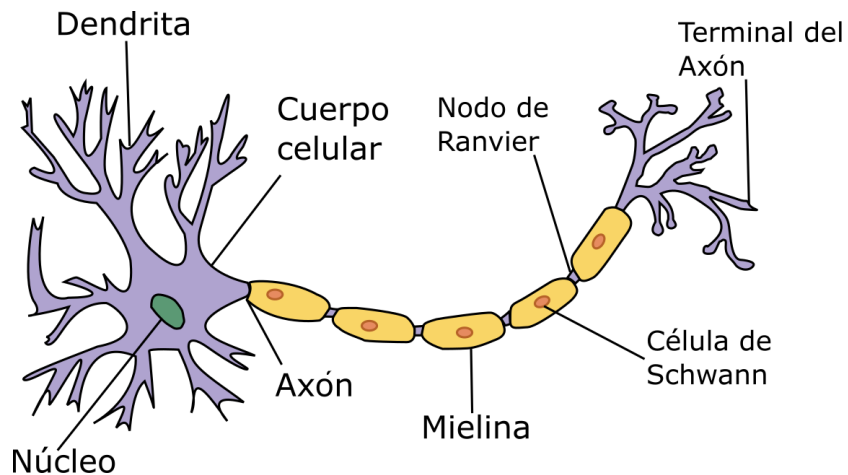


Figura 4.1: Esquema general de la composición básica de una neurona biológica [112].

4.2.1.1 Dendritas

Las dendritas (Figura 4.1) son estructuras altamente ramificadas que se especializan en la recepción y procesamiento de la entrada sináptica, conteniendo proteínas especializadas que facilitan la recepción, el procesamiento y la transferencia de estas señales al cuerpo celular de la neurona, representando así la sustancia estructural para las entradas sinápticas en las células nerviosas y proporcionando el plano para la conectividad sináptica en las redes neuronales [113]. Actúan como antenas, recogiendo información de miles de entradas presinápticas, algunas veces en decenas o incluso cientos de miles, lo que destaca la importancia de las dendritas en la mediación del flujo de información a través de las redes neuronales [114].

Las dendritas no son meros conductos pasivos para las señales eléctricas, sino que poseen capacidades de procesamiento potentes. Dentro de las dendritas, pueden existir fuertes gradientes de voltaje a lo largo del árbol dendrítico debido a la fuerte atenuación de las corrientes de entrada, lo que significa que las sinapsis individuales pueden tener efectos muy localizados [115]. Además, el trabajo teórico ha proporcionado nuevas perspectivas sobre las computaciones subyacentes de las neuro-

nas individuales y las redes utilizando implementaciones biológicamente plausibles de los procesos dendríticos. Esta comprensión avanzada de las dendritas está contribuyendo significativamente tanto a la comprensión biológica como a la computación neural artificial [116].

En la actualidad, hay un campo emergente en la ingeniería neuromórfica conocido como “computación dendrítica”, el cual busca diseñar sistemas informáticos que imiten las capacidades de procesamiento de información del cerebro utilizando conceptos como la plasticidad sináptica, la distribución masiva de las conexiones y el procesamiento paralelo [117, 118]. Aunque la computación dendrítica todavía se encuentra en sus etapas iniciales de desarrollo, se espera que tenga aplicaciones significativas tanto en la ingeniería neuromórfica como en medicina en general [119].

4.2.1.2 Soma o cuerpo celular

El soma o cuerpo celular (Figura 4.1) contiene el núcleo de la neurona, que alberga el material genético, y es el lugar donde se integran las señales eléctricas que provienen de otras neuronas a través de las estructuras dendríticas. Las dendritas son responsables de recoger las señales eléctricas entrantes hacia el soma. Una vez en el soma, estas señales pueden ser procesadas y, si alcanzan un cierto umbral, pueden generar una nueva señal eléctrica que viaja a lo largo del axón hacia otras neuronas o células efectoras [120, 121]. En la Figura 4.2 puede observarse en detalle el soma neuronal de una neurona biológica.

Quizás el rol más importante del soma es el hecho de que en él se lleva a cabo la síntesis de la mayor parte de proteínas que se encuentran en la neurona, y entre ellas las que van a formar parte o van a dar inicio a la síntesis de la mayor parte de neurotransmisores. El tamaño del soma ha sido explorado como un enfoque eficaz para distinguir diferentes poblaciones de neuronas en modelos experimentales, proporcionando una forma sensible de analizar y comprender las diferencias estructurales y

4.2. Anatomía, funcionamiento y sinapsis de las neuronas biológicas.

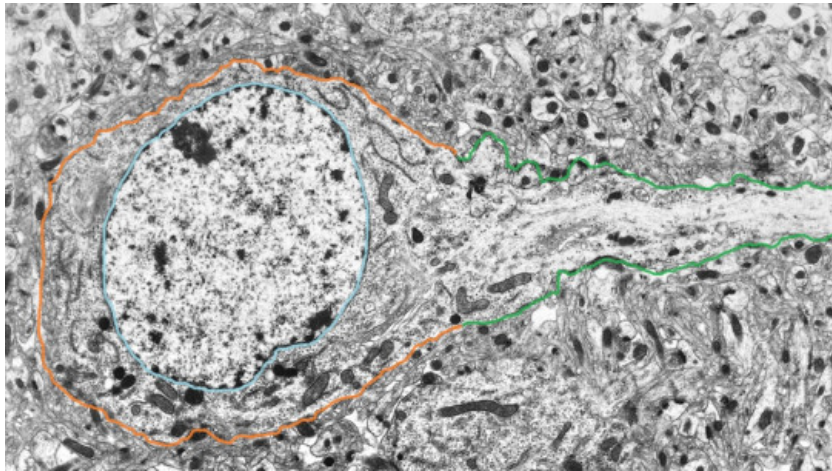


Figura 4.2: Imagen al microscopio de una neurona. Pueden observarse distintas áreas delimitadas: el soma, en naranja, el núcleo en azul y el axón en verde. Imagen extraída de [122].

funcionales entre las neuronas [123].

4.2.1.3 Axón

Los axones (Figura 4.1) son extensiones prolongadas de las neuronas que funcionan como cables biológicos para transmitir señales eléctricas (potenciales de acción) desde el cuerpo celular hacia otras neuronas o músculos. En humanos, los axones pueden variar en tamaño desde unas pocas micras hasta casi los dos metros, y tienen diámetros que fluctúan entre aproximadamente 0.3 a 15 μm , dependiendo del tipo de axón [124].

Estructuralmente, los axones exhiben una organización citoesquelética altamente periódica conocida como el *esqueleto periódico asociado a la membrana*, que es determinante para mantener la estructura y función del axón. Este citoesqueleto proporciona el soporte necesario para que el axón pueda extenderse a lo largo de distancias muy largas, resistir fuerzas mecánicas y torsiones, además de facilitar la adaptación y respuesta ante traumas o daños, como las lesiones axonales [125].

En el contexto de la mielinización, las células de *Schwann* desempeñan

un papel vital en el sistema nervioso periférico, al envolver los axones con capas de mielina [126]. La mielina es una sustancia lipídica que actúa como aislante eléctrico, permitiendo una transmisión de señales mucho más rápida y eficiente a lo largo del axón. Los nodos de *Ranvier*, que son pequeños espacios interrumpidos en la vaina de mielina, juegan un papel determinante en este proceso de conducción rápida de los potenciales de acción, ya que permiten la regeneración de la señal eléctrica en estos puntos, un fenómeno conocido como conducción saltatoria [127, 128].

A nivel funcional, los axones son responsables del transporte bidireccional a larga distancia entre el cuerpo celular y la sinapsis, representando la principal estructura de salida de las neuronas. Transportan información neuronal en forma de potenciales de acción desde el sitio de iniciación en el segmento inicial del axón hasta el terminal, en algunos casos, situado a varios centímetros de distancia, sin atenuación. Este proceso se ve enormemente optimizado por la presencia de la mielina y la estructura específica proporcionada por los nodos de *Ranvier* y las células de *Schwann* [128].

4.2.2 Sinapsis neuronal biológica

Durante el periodo que abarca finales del siglo XIX y comienzos del XX, existió una notable controversia respecto a si la transmisión sináptica era de naturaleza eléctrica o química. Había quienes sostenían que la señalización entre sinapsis se realizaba mediante el flujo directo de iones de una neurona a otra, caracterizando así una transmisión eléctrica. Por otro lado, había quienes argumentaban que esta transmisión dependía de la liberación de sustancias químicas por parte de una neurona, lo cual desencadenaba una respuesta en la neurona receptora, identificando así una transmisión química.

Con el tiempo, hemos llegado a entender que la transmisión sináptica puede manifestarse tanto de manera eléctrica como química, e incluso,

4.2. Anatomía, funcionamiento y sinapsis de las neuronas biológicas.

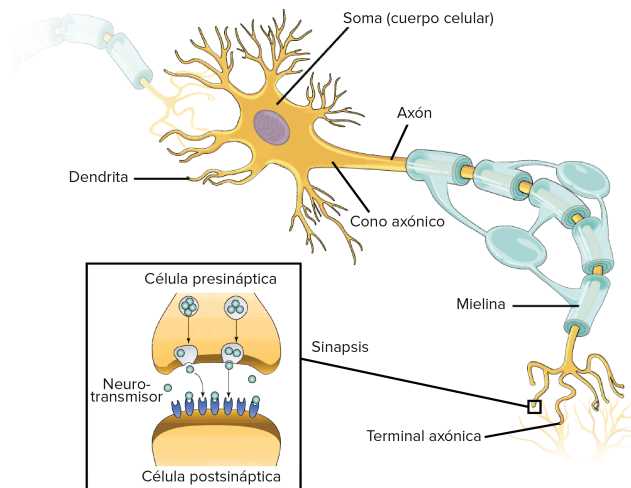


Figura 4.3: Esquema donde se presenta la conexión de una neurona con otras neuronas post y presinápticas. Imagen extraída de [129].

en algunos casos, ambas pueden coexistir en una misma sinapsis, involucrando varios tipos de canales iónicos que facilitan el flujo de iones a través de las membranas neuronales, lo que a su vez modula el potencial eléctrico de la membrana. A continuación se presentarán de forma breve los distintos canales que pueden intervenir en una sinapsis neuronal:

- **Canales de Sodio [Na⁺]:** Son esenciales para iniciar y propagar potenciales de acción. Se abren en respuesta a un cambio en el potencial de membrana, permitiendo la entrada de iones de sodio, lo que conduce a la despolarización de la membrana neuronal.
- **Canales de Potasio [K⁺]:** Ayudan a repolarizar la membrana neuronal después de la despolarización, permitiendo la salida de iones de potasio de la célula. También están involucrados en la hiperpolarización, que sigue a la repolarización.
- **Canales de Calcio [Ca²⁺]:** Están presentes en la terminación axonal de la neurona presináptica y se abren en respuesta al potencial de acción. La entrada de iones de calcio es esencial para la liberación de neurotransmisores en la sinapsis.

- **Bombas de Sodio-Potasio:** Aunque no son canales iónicos en el sentido tradicional, estas bombas activas son cruciales para mantener el equilibrio iónico y el potencial de membrana en reposo, extrayendo sodio y transportando potasio contra sus gradientes de concentración.
- **Canales Rectificadores:** Permiten el flujo de iones en una dirección preferente, ayudando a mantener el potencial de membrana en reposo y a modular la actividad eléctrica de las neuronas.

Dado que la transmisión química es más común y posee una complejidad mayor en comparación con la eléctrica, abordaremos inicialmente la transmisión química.

4.2.2.1 Sinapsis química

La sinapsis química es un proceso mediante el cual se transmiten mensajes entre neuronas a través de sustancias conocidas como neurotransmisores. Esta transmisión de información se da desde una neurona presináptica, o emisora, hacia una célula postsináptica, o receptora.

Los axones de una neurona individual pueden ramificarse en múltiples direcciones, lo que permite la formación de sinapsis con varias células postsinápticas. De forma análoga, una sola neurona puede recibir miles de entradas sinápticas provenientes de diversas neuronas presinápticas.

Dentro de la terminal axónica de la neurona emisora, se hallan numerosas vesículas sinápticas, que son esferas membranosas llenas de moléculas neurotransmisoras. Entre esta terminal axónica y la membrana de la célula postsináptica, se encuentra un pequeño espacio llamado espacio sináptico [130, 131]. En la Figura 4.4 puede observarse este proceso previo al disparo sináptico.

Cuando un impulso nervioso, o potencial de acción, llega a la terminal axónica, se activan los canales de calcio sensibles al voltaje presentes en la membrana celular, permitiendo la entrada del ion calcio [Ca^{2+}], que

4.2. Anatomía, funcionamiento y sinapsis de las neuronas biológicas.

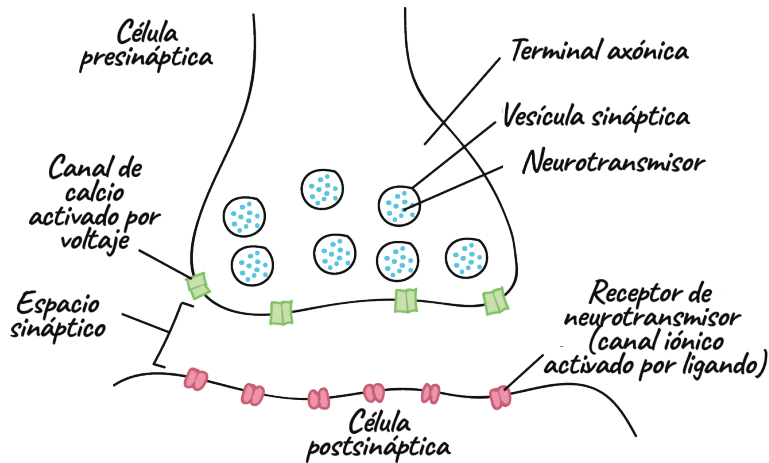


Figura 4.4: Ilustración del axón de una neurona presináptica y la dendrita de una neurona postsináptica, momentos antes de la generación de un impulso neuronal. Imagen extraída de [129].

se halla en mayor concentración fuera de la neurona. Esta incursión de $[Ca^{2+}]$ facilita la fusión de las vesículas sinápticas con la membrana de la terminal axónica, liberando así los neurotransmisores al espacio sináptico (Figura 4.5).

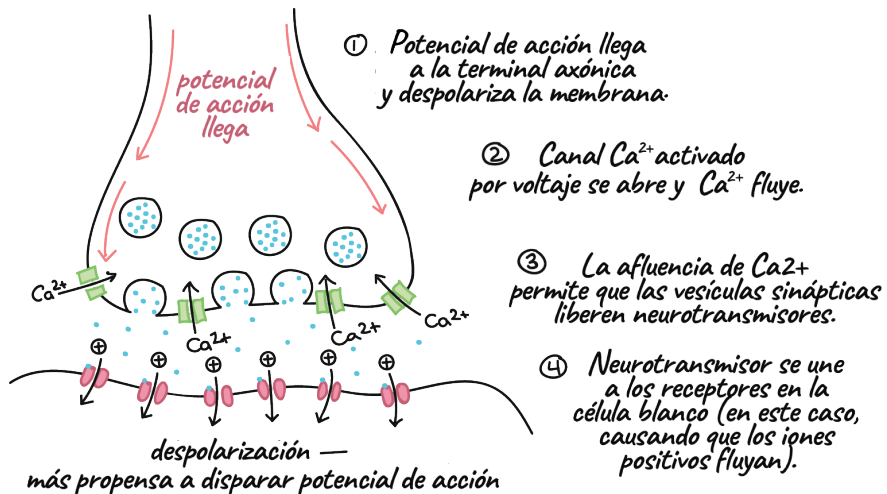


Figura 4.5: Esquema donde se representa una sinapsis química con despolarización negativa entre dos neuronas. Imagen extraída de [129].

Posteriormente, las moléculas neurotransmisoras atraviesan el espacio

sináptico y se acoplan a las proteínas receptoras situadas en la célula postsináptica. La activación de estos receptores postsinápticos desencadena la apertura o cierre de canales iónicos en la membrana celular, lo cual, dependiendo de los iones involucrados, puede resultar en una despolarización¹ o una hiperpolarización² [131].

Uno de los principales neurotransmisores inhibitorios en el sistema nervioso central es el *ácido gamma-aminobutírico* (GABA), el cual se ha simulado en numerosos trabajos de ingeniería neuromórfica en la última década, ya que juega un papel determinante en la función sináptica y la modulación de la actividad neuronal [132, 133, 134]. Su función principal es reducir la actividad neuronal al unirse a receptores específicos en las neuronas postsinápticas, lo que provoca una hiperpolarización de la membrana celular y disminuye la probabilidad de que la neurona postsináptica genere un potencial de acción. Esta acción inhibitoria del GABA es imprescindible para mantener el equilibrio y la estabilidad en la actividad neuronal, regulando la excitabilidad de las redes neuronales y previniendo la sobreexcitación que podría llevar a condiciones patológicas como la epilepsia [135]. El GABA también desempeña un papel importante en la regulación del sueño, la ansiedad y el estrés, así como en la coordinación motora y la función cognitiva [136, 137]. Además, su disfunción está implicada en una variedad de trastornos neurológicos y psiquiátricos, como la ansiedad, la depresión, el trastorno de estrés postraumático y la esquizofrenia [138, 139, 140].

¹Fenómeno que provoca que la membrana neuronal se polarice positivamente.

²Fenómeno que provoca que la membrana neuronal se polarice negativamente.

4.2. Anatomía, funcionamiento y sinapsis de las neuronas biológicas.

4.2.2.2 Sinapsis eléctrica

La sinapsis eléctrica en las neuronas permite una comunicación directa y rápida entre las células nerviosas. En la sinapsis eléctrica, las membranas de las neuronas pre y postsinápticas están unidas por una unión tipo "gap", o unión comunicante, a través de la cual fluye la corriente eléctrica de una célula a otra de manera directa (Figura 4.6) [141, 142]. En contraste con las sinapsis químicas, donde se liberan neurotransmisores que viajan a través de un espacio sináptico para interactuar con receptores en la neurona postsináptica, las sinapsis eléctricas tienen una conexión física directa entre las neuronas pre y postsinápticas, que se materializa en una estructura conocida como unión en hendidura o "gap junction"[143].

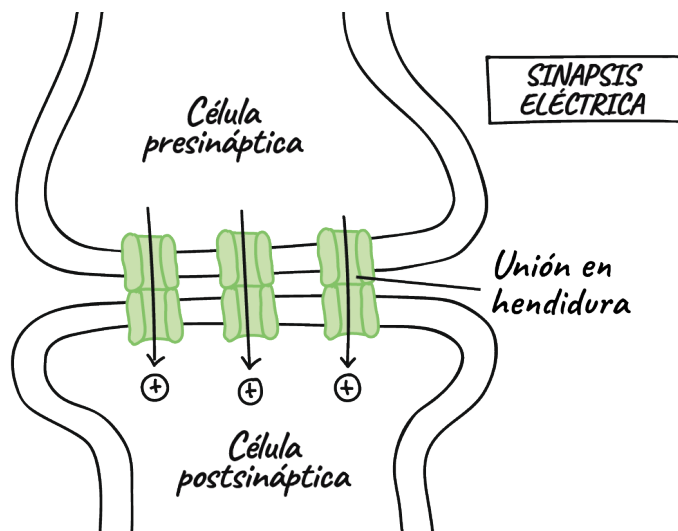


Figura 4.6: Esquema donde se representa una sinapsis eléctrica. En el espacio comprendido entre ambas neuronas pueden observarse las uniones de hendidura, por donde el flujo de la corriente fluye de la neurona presináptica a la postsináptica. Imagen extraída de [129].

El sustrato anatómico de las sinapsis eléctricas se basa en estas uniones en hendidura, que son regiones especializadas de la membrana compuestas por canales *transmembrana* que conectan directamente el citoplasma de las células adyacentes. Cada uno de estos canales inter-

celulares está formado por la conjunción de dos *hemicanales*, llamados *conexones*, que a su vez están formados por el ensamblado de seis proteínas conocidas como *conexinas* [144]. Estas estructuras permiten el flujo bidireccional de pequeñas moléculas, básicamente iones, entre las células, proporcionando una vía de baja resistencia al paso de la corriente eléctrica entre las dos células. Así, la despolarización de la membrana de una neurona puede transmitirse "directamente" a una neurona vecina a través de la sinapsis eléctrica [142].

Desde una perspectiva funcional, la comunicación interneuronal a través de las sinapsis eléctricas es notablemente más rápida que la comunicación a través de las sinapsis químicas. En las sinapsis químicas, existe un retraso sináptico mientras el potencial de acción alcanza el terminal presináptico, se libera el neurotransmisor y este interactúa con el receptor en la célula postsináptica. Sin embargo, en las sinapsis eléctricas, este retraso es prácticamente nulo, lo que permite una alta velocidad en la comunicación intercelular. Esta característica es determinante para el acoplamiento funcional simultáneo, o sincronización, de redes de neuronas que están unidas por sinapsis eléctricas[142].

Además, las sinapsis eléctricas y químicas a menudo se encuentran separadas, aunque también existen sinapsis mixtas quimicoeléctricas que se encuentran acopladas en algunos casos. La disfunción sináptica, ya sea eléctrica o química, puede tener efectos adversos en los órganos y las estructuras vitales, lo que resalta la importancia de entender las diferencias y las funciones de cada tipo de sinapsis [131].

4.2.2.3 Potenciales postsinápticos excitatorios e inhibitorios

Cuando un neurotransmisor se acopla a su receptor en una neurona receptora o postsináptica, se origina la apertura o el cierre de canales iónicos, lo que puede inducir una variación localizada en el potencial de la membrana de la célula receptora:

- Cuando el neurotransmisor produce que la neurona postsináptica sea mas propensa a generar su propio disparo neuronal, se dice que este cambio en el potencial de la membrana se llama *Potencial Excitatorio Postsináptico (PEPS)*.
- Por el contrario, si el neurotransmisor produce que la neurona postsináptica sea menos propensa a generar su propio disparo neuronal se dice que este cambio en el potencial de la membrana se llama *Potencial Inhibitorio postsináptico (PIPS)*.

Volviendo a los conceptos de hiperpolarización y despolarización mencionados en la sección 4.2.2.1, ahora es posible vincular los PEPS al fenómeno despolarizante, mientras que los PIPS son hiperpolarizantes. Estos conceptos forman el principio de funcionamiento básico de las sinapsis neuronales. Ligado a estos se presenta el concepto de suma de potenciales, ya sean excitatorios o inhibitorios, para que se produzca el disparo neuronal. Por ejemplo, supongamos que en cuatro diferentes dendritas de la misma neurona postsináptica se producen sinapsis excitatorias, como se muestra en la Figura 4.7:

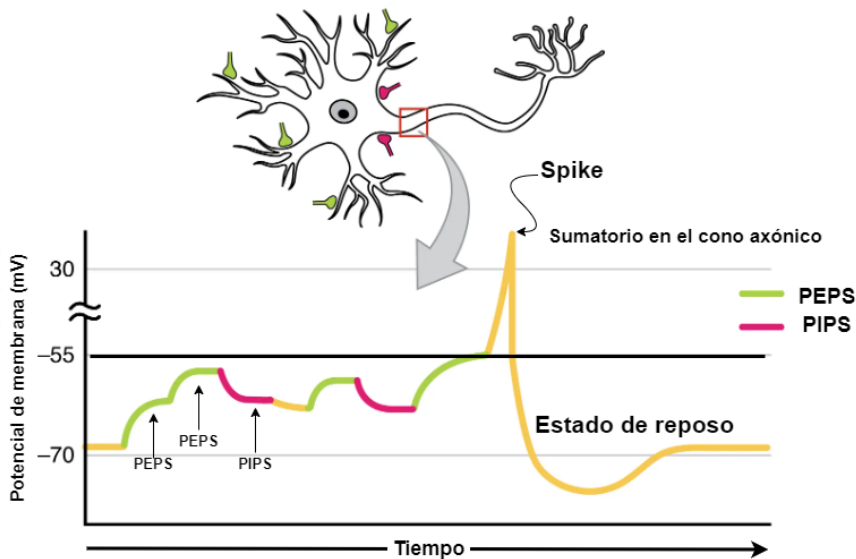


Figura 4.7: El resultado de la suma de los potenciales postsinápticos es el cambio global del potencial de membrana. En el punto A, varios PEPS diferentes se suman para producir una gran despolarización. En el punto B, una mezcla de PEPS y PIPS da lugar a un resultado final diferente para el potencial de membrana. Imagen basada en [145].

En el cono axónico, donde se origina el potencial de acción, una sola sinapsis no puede generar un PEPS suficientemente grande como para alcanzar el umbral necesario para disparar dicho potencial. No obstante, si dos PEPS ocurren simultáneamente, pueden combinarse y facilitar que el potencial de membrana alcance el umbral requerido.

Ahora bien, en el escenario donde junto a los dos PEPS se presenta un PIPS, este último podría contrarrestar la excitación generada por los PEPS, evitando así que el potencial de membrana alcance el umbral necesario para desencadenar un potencial de acción. Esta interacción entre potenciales excitatorios e inhibitorios en diferentes localizaciones de la membrana, pero que ocurren al mismo tiempo, es un ejemplo de suma espacial.

Por otro lado, es importante tener en cuenta que los potenciales postsinápticos no son fenómenos instantáneos, sino que persisten durante un

4.3. Analogía entre neuronas biológicas y artificiales

breve período antes de disiparse. En un escenario en el que una neurona presináptica disparase en dos ocasiones consecutivas rápidamente, se podrían generar dos PEPS. Si el segundo PEPS se produce antes de que el primero se haya disipado por completo, la sumatoria de ambos potenciales puede empujar el potencial de membrana hacia el umbral necesario para disparar un potencial de acción. En este contexto, la frecuencia de disparo presináptica juega un papel crucial en la modulación del potencial de membrana postsináptica, ilustrando así un caso de suma temporal [146].

4.3 Analogía entre neuronas biológicas y artificiales

Para resumir lo expuesto en las secciones anteriores, las neuronas biológicas (Figura 4.8a) están compuestas por un cuerpo celular o soma que alberga un núcleo, y por ramificaciones llamadas dendritas. Estas dendritas transfieren información a través de sinapsis, que se efectúan con neuronas adyacentes al soma. Además, cuentan con un axón que transporta los impulsos nerviosos desde el soma hasta la neurona postsináptica.

Por otro lado, la neurona artificial (Figura 4.8b) representa una versión simplificada de esta estructura biológica. Incluye solamente los elementos fundamentales: sinapsis, dendritas, soma y axón. En la neurona artificial, las sinapsis y dendritas funcionan como entradas al soma, que actúa como elemento procesador. Cada entrada posee un peso sináptico asignado, emulando la intensidad de una conexión biológica específica. El soma procesa estas entradas multiplicando cada una por su peso sináptico correspondiente y sumando los resultados. Esta suma es luego procesada por una función de transferencia, generando un resultado que se transmite a través del axón, funcionando como la vía de salida.

4. Neuronas biológicas VS artificiales

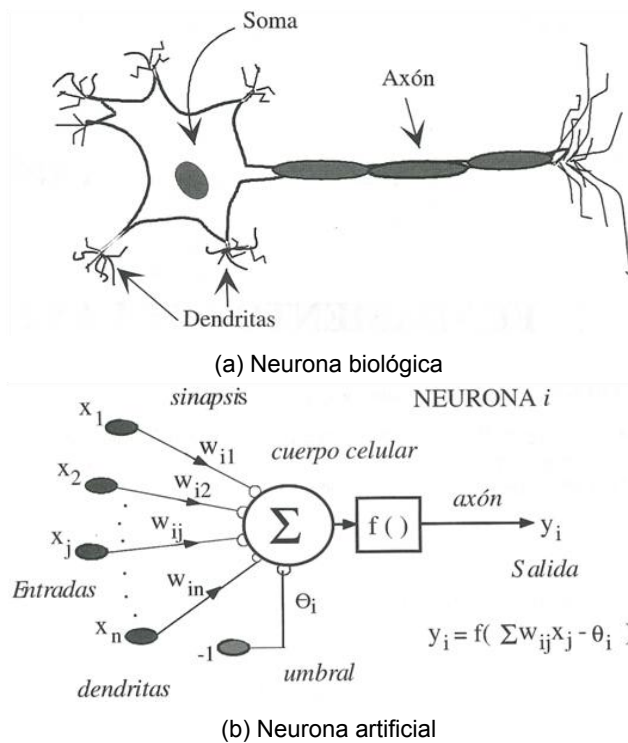


Figura 4.8: Analogía entre una neurona biológica (a) y una neuronal artificial (b), donde pueden apreciarse los elementos básicos de cada una de ellas.

En la última década, hemos sido testigos de un notable desarrollo en el campo de las ANNs, comenzando con la primera generación del perceptrón multicapa (MLP) y avanzando hacia técnicas más sofisticadas en las redes neuronales profundas de segunda generación (*Deep Neural Networks* (DNNs)). Este progreso ha sido impulsado en gran medida por la disponibilidad de grandes volúmenes de datos y por avances en dispositivos de computación de alto rendimiento, incluyendo las Unidades de Procesamiento Gráfico (GPU) de propósito general.

A pesar de estos avances, las ANNs aún no han logrado igualar la eficiencia energética y las capacidades de aprendizaje en línea de las redes neuronales biológicas. Aunque las ANNs y DNNs se inspiran históricamente en el cerebro humano, difieren significativamente en su estructura, cálculos neuronales y reglas de aprendizaje. Esta diferencia ha llevado

4.3. Analogía entre neuronas biológicas y artificiales

al desarrollo de las Redes Neuronales Pulsantes (SNNs), a menudo referidas como la tercera generación de redes neuronales. Las SNNs representan un paso adelante potencial para superar las limitaciones de las ANNs.

La implementación de SNNs en hardware neuromórfico, como TrueNorth, Loihi, SpiNNaker o NeuroGrid (analizado detalladamente en el capítulo 2), es un enfoque prometedor para abordar el consumo de energía. En las SNNs, similar a las redes neuronales biológicas, las neuronas se comunican mediante señales eléctricas discretas conocidas como pulsos o "spikes", operando en tiempo continuo. Esta similitud funcional con las redes neuronales biológicas permite a las SNNs imitar la baja demanda energética observada en la biología. Aunque las SNNs todavía no igualan el rendimiento de las DNNs en algunas tareas, están cerrando la brecha, requiriendo generalmente mucha menos energía para operar. Sin embargo, entrenar SNNs sigue siendo un desafío, principalmente debido a la complejidad de la dinámica neuronal y la naturaleza no diferenciable de las operaciones pulsantes. En la tabla 4.1 se presenta un resumen de las principales similitudes y diferencias entre neuronas biológicas, ANNs y SNNs, las cuales se comentarán en profundidad en el capítulo 5.

Propiedades	Redes neuronales biológicas	ANNs	SNNs
Representación de la información	Spikes	Escalar	Spikes
Paradigma de aprendizaje	Plasticidad sináptica	BP	Plasticidad/BP
Plataforma	Cerebro	VLSI	VLSI neuromórfico

Tabla 4.1: Comparativa de las propiedades entre redes neuronales biológicas, ANNs y SNNs. Datos extraídos de [147].

5

Redes neuronales pul- santes

“Lo que tenemos que hacer es asegurarnos de que estamos usando la IA de una manera que sea en beneficio de la humanidad, no en detrimento de ella.”

Tim Cook

5.1 Introducción

En la vanguardia de la investigación en ingeniería neuromórfica, las redes neuronales pulsantes (SNNs) emergen como una prometedora línea de estudio dentro de este área. Estas redes, que se distinguen por su capacidad para imitar más estrechamente la dinámica y la funcionalidad de las neuronas biológicas, representan un salto significativo desde el paradigma tradicional de las redes neuronales artificiales. A diferencia de sus predecesoras, las SNN se basan en la transmisión y el procesamiento de pulsos o *spikes*, que son las contrapartes artificiales a los

potenciales de acción en el cerebro humano. Esta particularidad no solo aporta una mayor biorealismo a los modelos computacionales, sino que también abre nuevas vías para el desarrollo de sistemas más eficientes en términos de consumo energético y capacidad de procesamiento. La investigación en SNN no solo aborda los retos inherentes a la imitación de los procesos neuronales complejos, sino que también explora cómo estas redes pueden ser aplicadas en una variedad de contextos, desde la robótica avanzada hasta el procesamiento sofisticado de señales.

Dentro de este amplio panorama, las redes *Winner Take All* (WTA) y los *Central Pattern Generators* (CPG) se destacan como componentes clave en el diseño y la funcionalidad de las SNN. Las redes WTA, conocidas por su capacidad para realizar selecciones competitivas y robustas en respuesta a estímulos complejos, juegan un papel crucial en la facilitación de la toma de decisiones y la selección de características en las SNN. Por otro lado, los CPGs, inspirados en los circuitos neuronales que generan patrones rítmicos de actividad en los organismos biológicos, son fundamentales en aplicaciones que requieren la generación de secuencias temporales coordinadas, como en la locomoción robótica. La integración y el estudio profundo de estas dos modalidades no solo enriquecen nuestra comprensión de las SNN, sino que también potencian su aplicabilidad en una variedad de desafíos técnicos y científicos. Este capítulo se dedica a explorar las complejidades, aplicaciones y el potencial de las SNN, con un enfoque particular en las redes WTAs y los CPGs, subrayando su relevancia e impacto en el campo de aplicación de esta tesis doctoral.

5.2 Fundamentos de las SNN

Las neuronas exhiben una complejidad fisiológica notable. En 1952, *Hodgkin & Huxley* realizaron un análisis pionero del comportamiento electrónico de una neurona aislada, en el que centraron su estudio en los canales de sodio y potasio, lo que les valió el Premio Nobel de Medi-

5.2. Fundamentos de las SNN

cina en 1963 [148]. Su investigación demostró cómo las neuronas representan, comunican y procesan información a través de pulsos eléctricos, conocidos como potenciales de acción o *spikes*. El modelo de Hodgkin & Huxley describía el comportamiento neuronal mediante ecuaciones diferenciales no lineales, donde un *spike* se caracteriza por un cambio en el voltaje a través de las membranas dendríticas. Las neuronas conectadas generan nuevos *spikes* en función de los recibidos, codificando la información en estos impulsos [149].

La Figura 5.1 ilustra las propiedades temporales del potencial de membrana de una neurona: inicialmente, los canales de la membrana neuronal están polarizados, manteniéndose por debajo de un umbral específico y, por lo tanto, inactivos. Al recibir estímulos externos (PEPS), si el potencial de la membrana alcanza este umbral, ocurre una rápida despolarización, generando un *spike* que transmite el impulso nervioso. Tras la despolarización, la neurona se polariza nuevamente, entrando en un estado de hiperpolarización durante un período refractario en el que no puede transmitir nuevos *spikes* salvo que la estimulación sea muy importante.

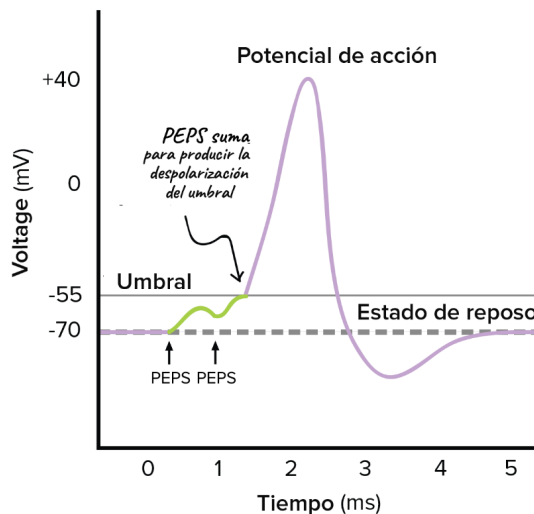


Figura 5.1: Propiedades temporales del potencial de acción de una neurona pulsante.

La eficiencia de la representación mediante *spikes* se manifiesta en varios aspectos: su simplicidad reduce la cantidad de canales de comunicación necesarios, al tiempo que proporciona información continua y minimiza la transmisión de datos redundantes. Esta forma de codificación envía *spikes* solo cuando son necesarios, evitando así la saturación innecesaria de los canales. Además, esta representación destaca por su robustez frente al ruido, ya que la información clave se encuentra en el intervalo entre los *spikes*, centrándose en la presencia o ausencia de estos. Esta capacidad de minimizar la saturación de los canales de comunicación y de centrarse en información clave entre *spikes* contribuye a optimizar el ancho de banda disponible.

5.3 Modelos neuronales pulsantes

Para simular el comportamiento de las redes neuronales biológicas, se han propuesto un amplio número de modelos neuronales pulsantes que muestran un gran equilibrio entre la precisión biológica y la factibilidad computacional (Figura 5.2), y cuya elección depende de los requisitos de implementación impuestos por el usuario. En esta sección se describirán algunos de los modelos neuronales pulsantes más conocidos, cuya información ha sido extraída de [147] y [150].

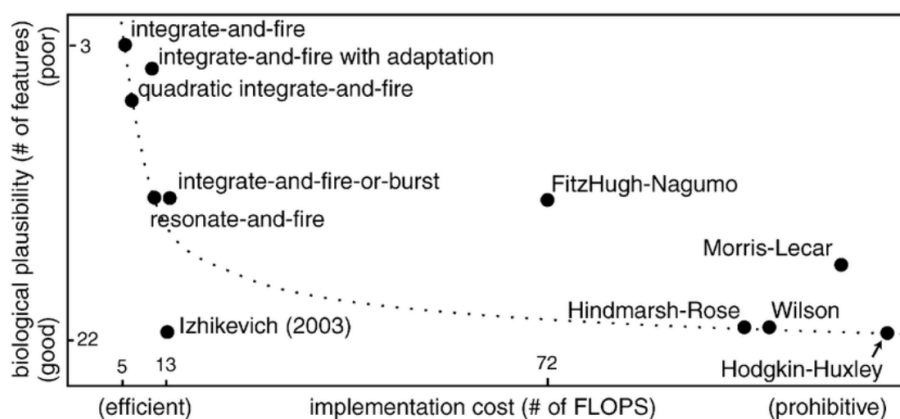


Figura 5.2: Modelos neuronales pulsantes. Plausibilidad biológica VS Coste de implementación. Gráfica extraída de [150].

5.3. Modelos neuronales pulsantes

5.3.1 Modelo HH

Hodgkin & Huxley llevaron a cabo un experimento en el axón gigante de un calamar y concluyeron que dos tipos de canales iónicos, el canal de K^+ y el canal Na^+ , están involucrados en la generación del potencial de acción (Véase sección 4.2). Este modelo se puede expresar añadiendo dos términos que se encargan del comportamiento de esos dos canales iónicos a la ecuación 5.1. Aunque el cambio en la permeabilidad del canal iónico se debe en realidad al cambio estructural de la proteína, puede describirse fenomenológicamente por la analogía de abrir y cerrar las puertas, considerando además la conductancia, que representa la facilidad con la que los iones pueden fluir a través del canal en función de su estado de apertura.

$$C_m \frac{dv_m(t)}{dt} = I_{ion}(t) + I_{syn}(t) \quad (5.1)$$

$$I_{ion}(t) = G_K n^4 (v_m - E_K) + G_{Na} m^3 h (v_m - E_{Na}) + G_L (v_m - E_L) \quad (5.2)$$

donde C_m es la capacitancia de la membrana (pF), v_m es el potencial de membrana (mV), I_{syn} es la corriente de entrada sináptica (pA), G_K representa la conductancia del ion K^+ , E_K representa el potencial de inversión del ion K^+ , G_{Na} representa la conductancia del ion Na^+ , E_{Na} representa el potencial de inversión del ion Na^+ , G_L representa la conductancia de fuga, y E_L representa el potencial de reposo, que actualmente se considera como el potencial de inversión del ion Cl^- . n , m y h son cantidades adimensionales entre cero y uno que están asociadas con la activación del canal de potasio, la activación del canal de sodio y la no activación del canal de sodio, respectivamente. Las tres puertas, n , m y h , se describen mediante la ecuación 5.3, donde g representa las variables de compuerta n , m y h , y la tasa de transición (donde $\alpha_g(v)$ es la tasa de transición de estados no permisivos a permisivos, mientras que $\beta_g(v)$ es la tasa de transición de estados permisivos a no permisivos) para cada compuerta. $\alpha_g(v)$ y $\beta_g(v)$ se definen según las ecuaciones 5.4.

$$\frac{dg}{dt} = \alpha_g(v_m)(1 - g) - \beta_g(v_m)g \quad (5.3)$$

$$\begin{aligned} \alpha_m(v_m) &= \frac{0,1(25 - v_m)}{\exp\left(\frac{25-v_m}{10}\right) - 1} \\ \beta_m(v_m) &= 4 \exp\left(-\frac{v_m}{18}\right) \\ \alpha_h(v_m) &= 0,07 \exp\left(-\frac{v_m}{20}\right) \\ \beta_h(v_m) &= \frac{1}{\exp\left(\frac{30-v_m}{10}\right) + 1} \\ \alpha_n(v_m) &= \frac{0,01(10 - v_m)}{\exp\left(\frac{10-v_m}{10}\right) - 1} \\ \beta_n(v_m) &= 0,125 \exp\left(-\frac{v_m}{80}\right) \end{aligned} \quad (5.4)$$

Al resolver estas ecuaciones, el modelo *Hodgkin & Huxley* puede simular el comportamiento del potencial de membrana durante la generación de impulsos neuronales sin introducir los procedimientos de generación de pulsos presentados en el modelo LIF (subsección 5.3.4).

Aunque el modelo de *Hodgkin & Huxley* es biológicamente preciso, tal y como se puede intuir al observar las ecuaciones que lo definen, demanda grandes recursos computacionales y es inviable en simulaciones a gran escala.

5.3.2 Modelo Izhikevich

Izhikevich propuso un modelo que combina la plausibilidad biológica de la dinámica del modelo HH y la eficiencia computacional de las neuronas LIF [30]. El modelo se representa por un sistema bidimensional de ecuaciones diferenciales ordinarias, quedando representado por las siguientes expresiones:

$$C_m \frac{dv_m}{dt} = k(v_m - E_L)(v_m - v_t) - u + I_{syn}(t) \quad (5.5)$$

$$\frac{du(t)}{dt} = a(b(v_m - E_m) - u) \quad (5.6)$$

Con las siguientes condiciones de reinicio después del *spike*:

$$\text{si } v_m \geq v_\theta \quad \left\{ \begin{array}{l} v_m \leftarrow c \\ u \leftarrow u + d \end{array} \right. \quad (5.7)$$

donde u representa la activación de las corrientes iónicas de K^+ y la no activación de las corrientes iónicas de Na^+ (pA), C_m es la capacitancia de la membrana (pF), v_m es el potencial de membrana (mV), I_{syn} es la corriente de entrada sináptica (pA), k es una constante que controla la excitabilidad de la neurona, E_L es el potencial de reposo de la membrana, a controla la velocidad de recuperación de u , b controla la sensibilidad de la variable u a la entrada de v_m y v_t es el potencial umbral instantáneo (mV). El modelo de Izhikevich puede exhibir los patrones de disparo de todos los tipos conocidos de neuronas corticales con la elección de parámetros basada en [151].

5.3.3 Modelo Adaptive Exponential Integrate-and-Fire (AdEx)

La neurona de Izhikevich mencionada anteriormente puede considerarse como un modelo cuadrático adaptativo de integración y disparo, mientras que el modelo AdEx [152] se caracteriza por una dependencia exponencial del voltaje. Gracias a la incorporación de una variable de evolución lenta que modela el doble umbral, este modelo logra mejorar el comportamiento exponencial del *spike*, permitiendo así capturar de manera efectiva la dinámica exponencial del potencial de membrana tras la activación neuronal, reflejando de manera precisa fenómenos observados en la actividad neuronal real.

$$C_m \frac{dv_m}{dt} = -G_L(v_m - E_L) + G_L \Delta T \exp\left(\frac{v_m - v_T}{\Delta T}\right) - w + I_{syn} \quad (5.8)$$

$$\tau_w \frac{dw}{dt} = a(v_m - E_L) - w \quad (5.9)$$

Con las siguientes condiciones de reinicio después del *spike*:

$$\text{si } v_m \geq v_\theta \begin{cases} v_m \leftarrow v_{\text{reset}} \\ w \leftarrow w + b \end{cases} \quad (5.10)$$

donde w es el peso sináptico que tiene en cuenta la adaptación, v_T es la corriente de rebase, a controla la influencia de $v_m - E_L$ en la adaptación y ΔT modela la nitidez de la función de activación de los canales de Na^+ . El modelo LIF se puede obtener del modelo AdEx tomando el límite $\Delta T \rightarrow 0$ y eliminando la corriente de adaptación w . El modelo AdEx comparte la capacidad de reproducir patrones de disparo a un bajo costo computacional, al igual que el modelo de Izhikevich.

5.3.4 Modelo Leaky Integrate and Fire (LIF)

El modelo conocido como IF se basa en la integración de la corriente de entrada a lo largo del tiempo. En este modelo, el potencial de membrana aumenta hasta alcanzar un umbral, sin considerar el comportamiento de los canales iónicos biológicos. Por otro lado, el modelo LIF añade un componente adicional al modelo IF: refleja la difusión de iones a través de la membrana celular e introduce un término de "fuga." decaimiento del potencial de membrana, representando situaciones en las que no se alcanza un *spike* en la célula. La dinámica de este modelo se describe mediante la siguiente ecuación:

$$C_m \frac{dv_m}{dt} = -G_L(v_m - E_L) + I_{syn}(t) \quad (5.11)$$

si $v_m \geq v_{th}$ $\{v_m \leftarrow v_{peak}$, entonces $v_m \leftarrow v_{reset}$

En este modelo, v_{th} representa el voltaje umbral, v_{peak} es el *spike*, y v_{reset} es el potencial de membrana en el cual se establecerá inmediatamente después de generar el *spike*. Cuando el potencial de la neurona alcanza el umbral v_{th} , se produce un disparo o pico de acción en la neurona. Tras este evento, el voltaje se restablece a cero durante un periodo refractario τ_{ref} , el cual limita la frecuencia con la que una neurona puede disparar.

Cuando la corriente de entrada sináptica es constante ($I_{syn}(t) = I$), y $v_{reset} = 0$, podemos proceder a calcular el potencial de membrana bajo las siguientes condiciones:

$$v_m(t) = R_m I (1 - \exp(-\frac{t}{\tau_m})) \quad (5.12)$$

Donde R_m representa la resistencia de la membrana ($M\Omega$) y $\tau_m = R_m C_m$ representa la constante de tiempo de la membrana. Dado que la neurona dispara cuando el potencial de membrana (v_m) alcanza el umbral (v_{th}), el primer *spike* $t(1)$ se puede encontrar estableciendo $v_m(t) = v_{th}$:

$$t^{(1)} = \tau_m \ln \left(\frac{R_m I}{R_m I - v_\theta} \right) \quad (5.13)$$

Por lo que la frecuencia de disparo en estado estacionario puede definirse como:

$$f = \left(\tau_{\text{ref}} + \tau_m \ln \left(\frac{R_m I}{R_m I - v_\theta} \right) \right)^{-1} \quad (5.14)$$

Debido a su simplicidad y bajo coste computacional, el modelo LIF es uno de los modelos mas utilizados para la simulación de SNNs, y será el modelo a utilizar en esta tesis doctoral.

5.4 Redes *Central Pattern Generator*

En el estudio de los seres vivos, el CPG es un componente básico del aparato locomotor. Este sistema consiste en circuitos neuronales biológicos que pueden generar salidas rítmicas sin la necesidad de entradas rítmicas. Estos circuitos, compuestos por pequeños grupos de neuronas relativamente autónomos, son responsables de crear patrones neuronales rítmicos esenciales como el caminar, la respiración o la masticación [153, 154]. En la Figura 5.3 se observa el comportamiento descrito: una pequeña red neuronal con seis neuronas que provocan a su salida un patrón oscilante.

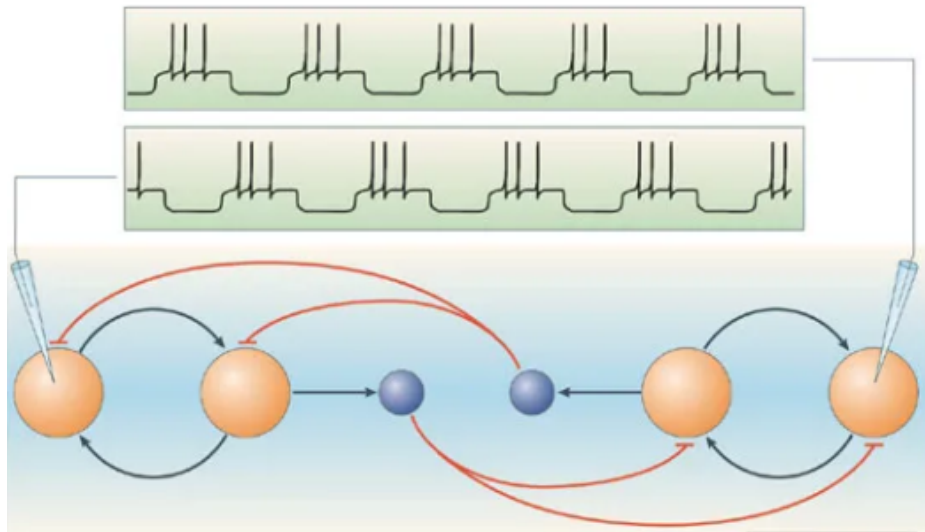


Figura 5.3: Analogía gráfica de un CPG. Arriba: patrón rítmico de salida del CPG. Abajo: estructura de la red neuronal, en negro conexiones excitatorias y en rojo inhibitorias. Imagen extraída de [155].

Antes de profundizar en una explicación más técnica, resulta útil comprender el principio de funcionamiento de los CPGs mediante una analogía simple, como la de un reloj de péndulo:

- El péndulo oscila hacia adelante y hacia atrás de manera rítmica y constante, similar a como un CPG produce patrones rítmicos de señales. Al igual que un reloj de péndulo no necesita ser empujado constantemente para seguir moviéndose, un CPG genera patrones de actividad de forma automática y rítmica, sin necesidad de un impulso externo constante.
- Así como es posible ajustar la velocidad del péndulo en un reloj, el ritmo y la intensidad de los patrones generados por un CPG pueden ser modificados por diferentes señales, lo que permite adaptarse a diferentes situaciones (como aumentar o disminuir el patrón de marcha).
- Si se imaginan varios relojes de péndulo trabajando juntos pero con ritmos ligeramente diferentes, es posible asimilar como varios CPGs en el sistema nervioso pueden coordinarse entre sí para producir movimientos complejos y coordinados, como correr o nadar.
- Aunque un CPG puede funcionar de forma independiente, este puede ser influenciado por factores externos, como el medio ambiente o las señales sensoriales, de manera similar a cómo el viento o un empujón pueden afectar en el movimiento de un péndulo.

5.4.1 Inspiración biológica

Los CPGs son clave para generar patrones de movimiento en los animales. Este mecanismo ha despertado un gran interés en su aplicación en campos como la robótica y la ingeniería neuromórfica. En robótica, los CPGs se emulan a través de osciladores acoplados, ANNs o SNNs. En particular, Las SNNs son consideradas las implementaciones más similares a los sistemas biológicos debido a su habilidad para imitar los patrones de disparo neuronal.

5.4. Redes *Central Pattern Generator*

Aunque la retroalimentación sensorial no es necesaria para generar patrones rítmicos, sí que juega un papel importante en la configuración de éstos. Gracias a la retroalimentación sensorial que poseemos los seres vivos, es posible mantener coordinados el movimiento corporal con la señal emitida por los CPGs. Hay numerosos experimentos que demuestran la importancia de esta retroalimentación sensorial en la actividad del CPG, como por ejemplo el realizado por A.D McClellan y W. Jang en 1993 [156], en el que demuestran como el movimiento de la cola de la lamprea modifica la actividad del CPG, la cual está bloqueada en frecuencia con los movimientos mecánicos, lo que sugiere una relación íntima entre la actividad neuronal que coordina los movimientos y la actividad mecánica. En un estudio mas reciente, donde una cinta mecánica induce el movimiento a un gato al cual se le había seccionado el tallo cerebral, se demuestra que se producen transiciones entre el trote y el galope del animal al introducir una variación en la velocidad de la cinta [157].

En cuanto a la localización de los CPGs, se han realizado diversos estudios moleculares, genéticos y de imagen, como el realizado por el neurocientífico *Ole Kiehn* en 2003 a una rata neonata [158], en el cual se arrojan evidencias que demuestran que las redes responsables de la locomoción están distribuidas en las regiones torácicas y lumbares inferiores de la médula espinal.

Finalmente, aunque los detalles anatómicos de los CPGs se conocen específicamente en sólo unos pocos casos, se ha demostrado que dependen de redes neuronales relativamente pequeñas y autónomas (en lugar de todo el sistema nervioso) para generar patrones rítmicos. Tal y como demuestran *Julia T. Choi* y *Amy J. Bastian* en su artículo sobre las redes neuronales independientes encargadas de la acción de caminar en el ser humano [159], hay diferentes estructuras que controlan el movimiento hacia adelante y hacia atrás, así como diferentes circuitos que controlan cada pierna. Estas estructuras generalmente no están interconectadas, aunque sí que puede existir retroalimentación entre ellas.

5.4.2 El CPG bioinspirado

Realizando una analogía eléctrica, los CPGs se suelen modelar como sistemas distribuidos de osciladores no lineales en cuanto a la relación frecuencia de entrada-salida. Estos osciladores no lineales están compuestos normalmente por una o varias neuronas artificiales, las cuales impulsan a las demás neuronas, denominadas motoneuronas pulsantes, cuyas señales suelen llevarse a un actuador [157].

Las primeras nociones de un circuito autónomo capaz de imitar estructuras neuronales y de generar oscilaciones mantenidas para posibilitar la locomoción aparecieron a principios del siglo XIX. La idea primitiva era que un sistema de neuronas totalmente autónomas podía generar un patrón rítmico, capaz de inducir el movimiento en los músculos de manera similar a la forma observada en la locomoción natural [160].

Los CPGs artificiales son autónomos en el sentido de que pueden funcionar sin una entrada por parte de sensores o de una unidad de procesamiento de mayores prestaciones, principalmente mediante redes neuronales recurrentes [161], aprendizaje no supervisado [162] o por el propio diseño del algoritmo de la red. Por otro lado, también son capaces de alterar un patrón rítmico sin la necesidad de involucrar entidades de mayor complejidad del sistema. Sin embargo, de forma análoga al CPG biológico mencionado anteriormente, los CPGs artificiales se pueden implementar junto con sensores para generar una realimentación que será procesada por unidades de procesamiento más complejas, ofreciendo mayor estabilidad y una mayor adaptación a los cambios del entorno.

La implementación de CPGs en hardware neuromórfico como SpiNNaker y Loihi ha demostrado ser particularmente prometedora [163, 164, 165]. Estos sistemas no solo replican la eficiencia energética y la velocidad de procesamiento del cerebro humano, sino que también abren nuevas vías para investigaciones en neurociencia, como el estudio de bucles sensoriomotores [166]. Un ejemplo notable es el control de un modelo simulado de lamprea, demostrando el potencial de los algorit-

mos de pulsos en el hardware neuromórfico [164].

El uso de SNN para la generación de locomoción dio lugar a los sCPG. Estas estructuras neuronales generan un patrón de salida rítmico, el cual puede utilizarse para generar movimiento en las plataformas robóticas. Además, una de las ventajas de este sistema es que permite generar patrones muy estables sin la necesidad de incorporar ningún tipo de información sensorial o cerebral que regule el funcionamiento o la frecuencia de oscilación de este [18].

5.5 Redes *Winner-Take-All*

Las redes *Winner-Take-All* (WTA) son estructuras neuronales caracterizadas por su habilidad para seleccionar una única respuesta dominante de entre múltiples entradas competitivas. Estas redes, ubicadas en diversas áreas del neocórtex, permiten que múltiples neuronas o grupos neuronales compitan por la activación. Sin embargo, solo la población con la señal más fuerte o que cumple con un criterio específico se activa, inhibiendo a las demás [167, 168].

La Figura 5.4 ilustra dos ejemplos de conectividad en una red WTA. La Figura 5.4a muestra una red biológica, donde la inhibición es mediada por interneuronas inhibitorias globales. Estas últimas son estimuladas por neuronas excitatorias y esta, a su vez, inhibe a todas las neuronas excitatorias. En contraste, la Figura 5.4b presenta un modelo con conectividad inhibitoria completa de eficacia V_I y autoexcitación de eficacia V_{self} que estabiliza la selección del ganador.

En ingeniería neuromórfica, las redes WTA son fundamentales debido a su capacidad para emular dinámicas neuronales cerebrales, siendo determinantes en procesos como la toma de decisiones y el procesamiento de la memoria. Investigaciones destacadas en este campo incluyen el trabajo de *Wolfgang Maass* sobre el poder computacional de las redes WTA [170], el estudio de *Oster, Douglas y Liu* sobre computación pul-

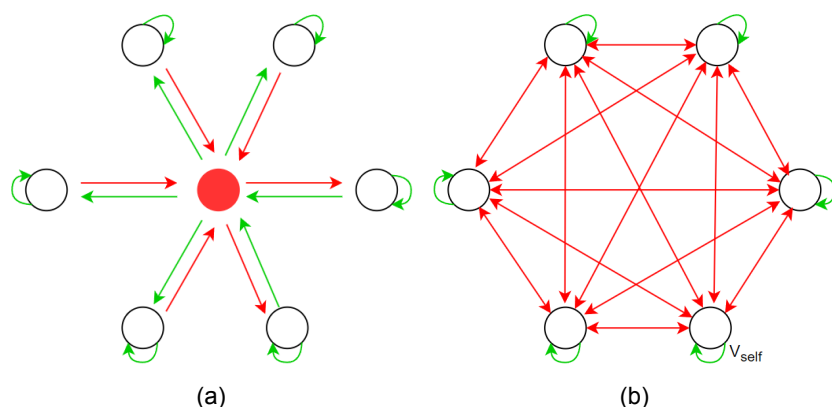


Figura 5.4: Ejemplo de conectividad de una red WTA: (a) en redes biológicas, la inhibición está mediada por poblaciones de interneuronas inhibitorias globales (círculo rojo relleno). Para realizar una operación del tipo "Winner-Take-All", estas son impulsadas por neuronas excitatorias (círculos no rellenos) y, a cambio, inhiben a todas las neuronas excitatorias (flechas verdes: conexiones excitatorias; flechas rojas: inhibitorias). (b) Modelo de red en el que la interneurona inhibidora global es reemplazada por una conectividad inhibitoria completa de eficacia V_I . La autoexcitación de la eficacia sináptica V_{self} estabiliza la selección de la neurona ganadora. Imagen basada de [169].

sante en WTA [171], y la investigación de *Kappel, Nessler y Maass* en el uso del aprendizaje STDP en WTA para modelar aprendizaje online de modelos ocultos de *Markov* [51].

Un avance significativo en aplicaciones prácticas es el uso de redes de nanocables memristivas, los cuales forman caminos de conectividad WTA esenciales para establecer estados de memoria en sistemas complejos [172]. Además, se han desarrollado circuitos neuromórficos basados en WTA para implementar mecanismos de memoria de trabajo y toma de decisiones [173].

Finalmente, el paradigma WTA se ha utilizado como un componente clave en la implementación de funciones cognitivas en sistemas electrónicos neuromórficos. Un ejemplo notable es su uso en la clasificación en tiempo real de patrones de movimiento observados por una retina neuromórfica, demostrando la versatilidad y aplicabilidad práctica de las redes WTA en tareas de procesamiento y toma de decisiones [174].



Materiales

6

Herramientas software en ingeniería neuromórfica

“La invención, y esto debe admitirse humildemente, no consiste en crear de la nada, sino del caos.”

Mary Shelley

6.1 Introducción

En la era actual, marcada por un avance sin precedentes en el desarrollo de hardware neuromórfico, incluyendo una variedad de sensores y sistemas especializados para implementar y ejecutar redes neuronales pulsantes en tiempo real, ha surgido una necesidad imperiosa de contar con un conjunto robusto y eficiente de herramientas de software. En este capítulo se abordan las herramientas software utilizadas en esta tesis doctoral, dedicadas al procesamiento de información pulsante. Estas herramientas se han vuelto indispensables por varias razones:

- La tarea de depurar sensores neuromórficos representa un desafío significativo. Esto se debe a que la información que estos sensores emiten se encuentra codificada no solo en el intervalo de tiempo entre dos pulsos consecutivos, conocido como ISI (*Inter-Spike Interval*), sino que también puede codificarse en la frecuencia de emisión de los pulsos o en el instante en el que se emite el primer *spike*, entre otros. Debido a esta particularidad, es esencial disponer de herramientas especializadas que permitan analizar adecuadamente los datos generados por la salida de estos sensores.
- Los usuarios interesados en el análisis de datos de sensores neuromórficos podrían centrarse en la información específica de los pulsos emitidos por estos sensores. Para extraer datos útiles, se requiere el uso de algoritmos especializados que permitan realizar clasificaciones o estudios detallados. Estas herramientas deben ser aptas para procesar los datos tanto en tiempo real como para almacenarlos, permitiendo así un análisis posterior.
- Para entrenar una red neuronal utilizando la información de salida de un sensor neuromórfico, es crucial contar con una aplicación de software específica. Esta aplicación debe ser capaz de generar un conjunto de datos procesables a partir de la información capturada por el sensor. El software debe enfocarse en extraer los componentes más relevantes de esta información y almacenarlos en un formato estandarizado.
- Antes de implementar una SNN en un sistema de hardware neuromórfico, comúnmente se diseña, entrena y prueba en un simulador para verificar que el modelo específico a implementar se comporte como se espera para un conjunto generalizado de muestras de entrada. Varias bibliotecas como PyNN [27] y Nengo [175] nos permiten construir redes complejas de SNNs, y simuladores como NEST [176], Brian2 [177] y NEURON [178] pueden simular esos modelos antes de desplegar la red en una plataforma de hardware.

6.2 jAER

jAER es un software de código abierto desarrollado para ordenadores, especializado en la visualización y manipulación de datos basados en eventos, ya sean en tiempo real o pregrabados [179]. Esta herramienta, construida en *Java*, facilita el desarrollo de algoritmos y aplicaciones enfocadas en el procesamiento de eventos en tiempo real. El núcleo de *jAER* es una aplicación denominada "*jAERViewer*", diseñada para interactuar con cualquier dispositivo que utilice la tecnología AER a través de una interfaz USB. *jAERViewer* permite a los usuarios realizar una variedad de operaciones con estos dispositivos, como visualizar los eventos generados por el dispositivo en tiempo real, grabar estos eventos, reproducir secuencias AER previamente grabadas, y procesar los eventos con la ayuda de diferentes filtros. Existen otras herramientas software que facilitan la interconexión entre dispositivos neuromórficos y que permiten visualizar la salida de estos dispositivos, como *NAVIS* [180] para la visualización de la salida de un NAS o *AERTools* [181], para la visualización de eventos AER emitidos por dispositivos neuromórficos.

Existe una amplia gama de dispositivos compatibles con *jAER*, abarcando desde sensores dinámicos de visión y audio hasta tarjetas AER de monitorización y secuenciación, así como controladores de servomotores. Estos dispositivos generan eventos de manera asíncrona, asignándoles una marca temporal con una precisión de 1 *s*. Posteriormente, estos eventos se transmiten al ordenador a través de USB en paquetes, que incluyen un número variable de eventos. Al recibirlos, *jAER* aplica una serie de filtros seleccionados por el usuario, permitiendo que el software visualice y añada anotaciones a los eventos procesados por el filtro final.

En este trabajo se ha utilizado *jAER* para recopilar eventos AER de un sensor de visión dinámico (DVS) para su análisis en tiempo real. La Figura 6.1 ilustra una captura de pantalla de la interfaz de *jAER*, mostrando la recepción de pulsos en tiempo real tanto de un NAS como de un DVS.

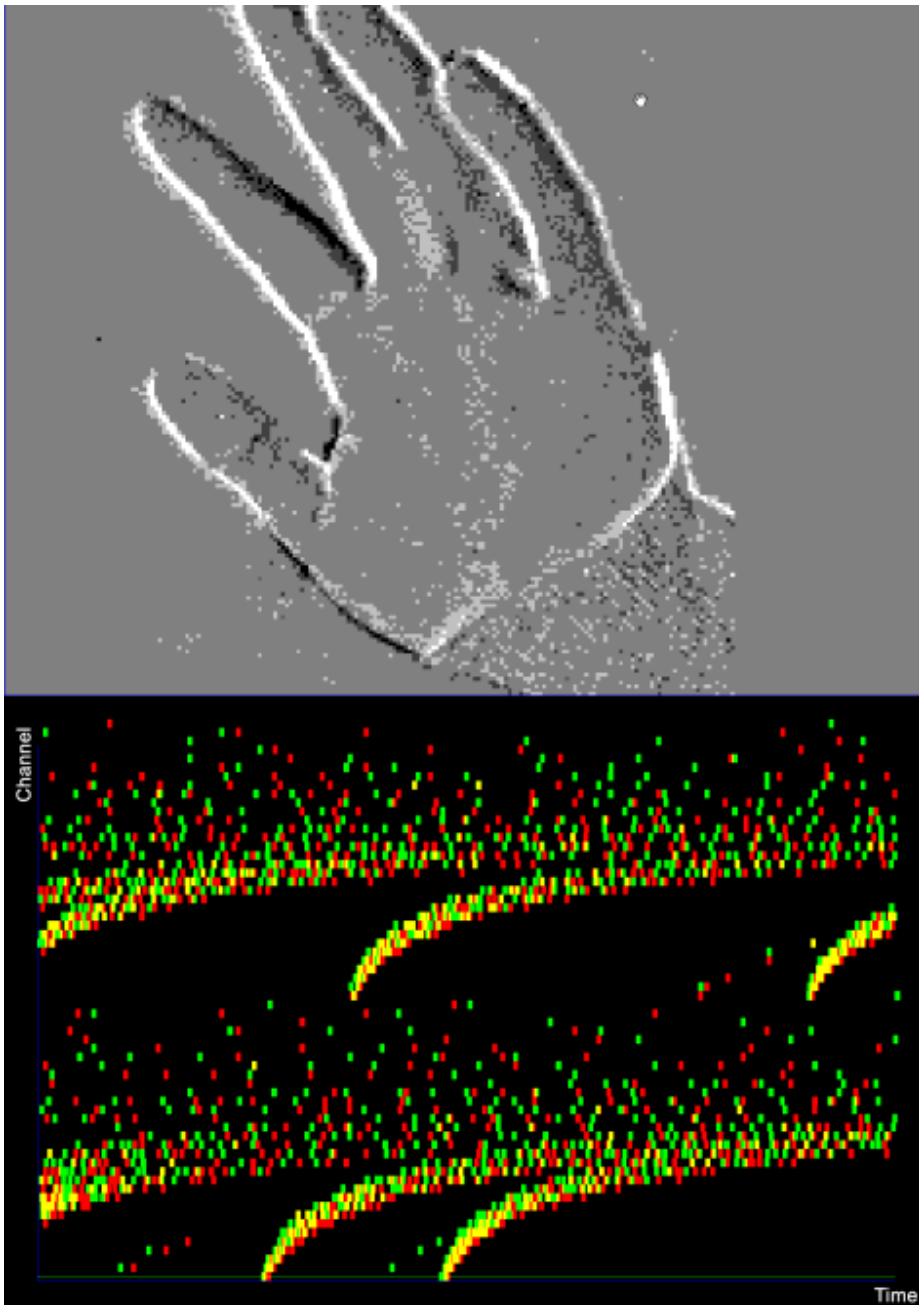


Figura 6.1: Representación de la salida del DVS de una mano en movimiento en *jAER* (arriba) y representación de la salida de un NAS de un altavoz hablando en *jAER* (abajo).

6.3 PyNN

PyNN es un lenguaje de programación basado en *Python*, diseñado específicamente para la creación de modelos de redes neuronales [27]. Permite a los usuarios desarrollar modelos de red neuronal escribiendo código con un conjunto de herramientas y funciones propias, y ejecutar estos modelos en diversos simuladores y sistemas de hardware neuromórfico sin necesidad de modificar el código. Actualmente, *PyNN* es compatible con simuladores como *NEURON* [178, 182], *NEST* [176] y *Brian2* [177], así como con sistemas de hardware neuromórfico como *SpiNNaker* [28] y *BrainScaleS* [22].

La API de *PyNN* ofrece un enfoque de alto nivel y abstracto para modelar redes neuronales. Los usuarios pueden fácilmente definir modelos de red neuronal, estableciendo poblaciones de neuronas y sus conexiones usando modelos de células predefinidos y algoritmos de conectividad comunes, sin la necesidad de desarrollarlos desde cero. Además, *PyNN* brinda la opción de acceder a características más detalladas, como los aspectos específicos de neuronas y sinapsis individuales, cuando se requiera. La plataforma incluye una biblioteca estándar de modelos de células, sinapsis y plasticidad sináptica, junto con los mencionados algoritmos de conectividad, permitiendo a su vez que los usuarios personalicen y codifiquen su propia conectividad en *Python*.

6.4 Brian2

Brian2 [177] es un simulador neuronal de código abierto diseñado para la simulación de redes neuronales biológicamente detalladas. Este software proporciona una plataforma flexible para llevar a cabo simulaciones de neuronas y redes neuronales de diversas complejidades. *Brian2* se destaca por su sintaxis intuitiva, lo que permite a los usuarios definir modelos con ecuaciones diferenciales que describen directamente los procesos biológicos de las neuronas.

Una de las características clave de *Brian2* es su capacidad para realizar simulaciones tanto en tiempo discreto como en tiempo continuo, lo que lo hace adecuado para una amplia gama de aplicaciones científicas. Además, al estar basado en *Python* admite la integración con otras herramientas y lenguajes de programación, lo que facilita su uso en proyectos de neurociencia computacional más amplios.

Por otro lado, *Brian2* también se utiliza como herramienta educativa, proporcionando a los estudiantes una forma accesible de explorar conceptos complejos en neurociencia y modelado neuronal.

El software se basa en la extensibilidad y la personalización, permitiendo a los usuarios desarrollar y compartir sus propios modelos y componentes de simulación. Esto ha llevado a una creciente comunidad de usuarios y desarrolladores que contribuyen al enriquecimiento y la mejora continua de la plataforma. *Brian2* es capaz de ofrecer resultados rápidos y robustos, lo cual es fundamental antes de proceder a la implementación de soluciones en plataformas de hardware específicas.

Dado que tanto SpiNNaker como *Brian2* soportan *PyNN*, se utilizará *Brian2* para modelar, adaptar y comprobar la SNN y, después de esto, implementar el modelo en SpiNNaker.

7

El protocolo AER

*“El trabajo del hombre de genio ,
aunque este equivocado o mal
dirigido, muy pocas veces deja
de convertirse en un verdadero
beneficio para la humanidad.”*

Víctor Frankenstein

7.1 Introducción

La mayoría de los sistemas neuromórficos se componen de al menos un sensor neuromórfico y varias capas de redes neuronales pulsantes, diseñadas para procesar la información suministrada por estos sensores [183, 184]. Estos sistemas intentan replicar la compleja interconexión de las neuronas biológicas. No obstante, debido a las limitaciones físicas en términos de conectividad, y considerando la elevada densidad de conexiones en el cerebro (donde cada neurona de las $10^5/mm^3$ puede conectar con otras 10.000, alcanzando una densidad de hasta $4km/mm^3$ [185]), resulta inviable replicar esta conectividad en sistemas VLSI. A pesar de esto, las neuronas biológicas tienen una tasa de disparo que oscila entre 1 y 10 Hz, que puede aumentar a kHz o MHz cuando se

agrupan cientos de ellas [186, 187]. Esto indica que los circuitos electrónicos modernos superan en velocidad a las neuronas biológicas. Para superar el desafío de la conectividad, y aprovechando la diferencia temporal y la alta capacidad de ancho de banda de los sistemas VLSI, se ha propuesto un mecanismo que multiplexa en tiempo la información de un grupo de neuronas en un solo canal de comunicación.

7.2 Principio de funcionamiento

En el protocolo AER cada neurona posee una dirección única, una técnica conocida como Representación de Eventos por Dirección (AER), introducida por primera vez en 1991 [188, 189, 190, 191]. Originalmente, el protocolo AER se diseñó para la comunicación pulsante entre neuronas en implementaciones VLSI [192], pero también facilita la transmisión y recepción de grandes volúmenes de información codificada por tasa de eventos a través de canales de menor ancho de banda, como los buses digitales asíncronos. Sin embargo, este enfoque requiere el empleo de un mayor número de estos buses.

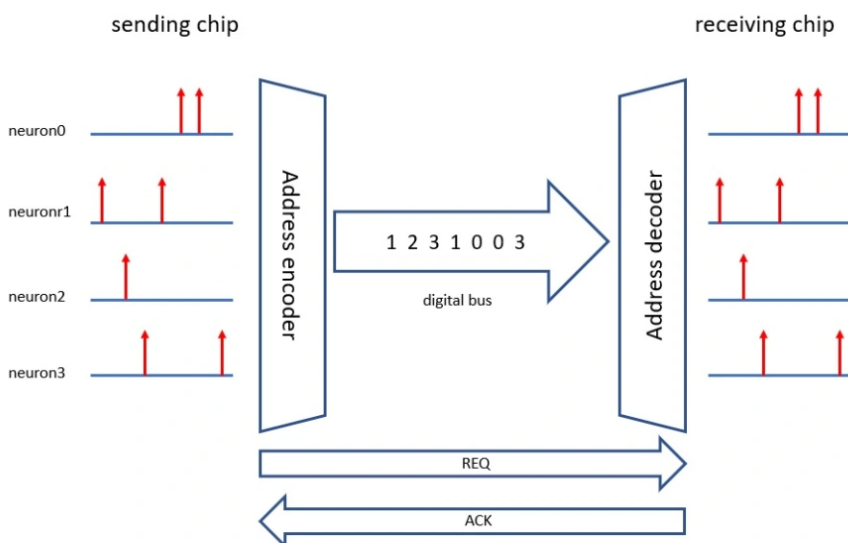


Figura 7.1: Transmisión de información pulsante usando el protocolo AER.

7.2. Principio de funcionamiento

La función principal de los circuitos AER consiste en proporcionar una representación de eventos asíncronos en tiempo real para la comunicación entre las neuronas. Como ilustra la Figura 7.1, este protocolo emplea un bus de alta velocidad y multiplexado compartido (bus AER) para la transmisión de los impulsos emitidos por las neuronas en un chip. Cada neurona se identifica por una dirección única (0, 1, 2 y 3, respectivamente). Al emitir un impulso una neurona, un circuito de arbitraje coloca la dirección de dicha neurona en el bus AER, generando así un evento. En consecuencia, los impulsos asíncronos son codificados y multiplexados por el circuito AER en el bus, respetando el orden en que se generaron. El instante en que se genera la dirección de la neurona corresponde al momento de su emisión, más un breve retraso ocasionado por el proceso de codificación, los cuales pueden llevar incluso a la pérdida de eventos, en los casos en los que la capacidad de procesamiento en la comunicación es limitada. Para abordar esta pérdida, se pueden implementar algoritmos de control, llamados comúnmente *arbitradores*, los cuales controlan la congestión computacional.

Posteriormente, la dirección de la neurona que generó el evento AER se decodifica en el chip receptor tan pronto como llega, utilizando un decodificador asíncrono, enviando el impulso inicial a la neurona correspondiente. De esta manera, las neuronas del transmisor y del receptor están conectadas a través del bus AER utilizando el protocolo AER. Si el retraso entre *spikes* adyacentes en la entrada es suficientemente largo, el decodificador AER enviará los *spikes* a las neuronas correspondientes con un retraso que corresponde al tiempo que el *spike* tarda en llegar al decodificador más un pequeño retraso causado por el circuito de decodificación.

Este protocolo presenta un intercambio de cuatro fases entre el transmisor y el receptor, el cual garantiza la sincronización entre ambos chips:

1. El transmisor inicia el proceso con una solicitud \rightarrow REQ
2. El receptor responde a la solicitud con un reconocimiento \rightarrow ACK
3. Para finalizar la transmisión, el transmisor retira la solicitud \rightarrow REQ
4. El receptor realiza lo mismo con la señal de reconocimiento \rightarrow ACK

Este intercambio de cuatro fases, propuesto en el proyecto de investigación europeo CAVIAR [193] puede apreciarse gráficamente en la Figura 7.2, junto con sus requisitos temporales en la Tabla 7.1.

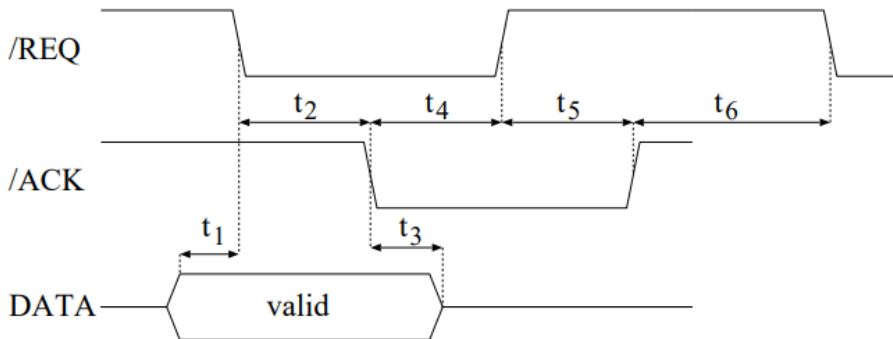


Figura 7.2: Intercambio de cuatro fases [193].

	min	max	promedio
t1	0s	∞	
t2	0s	∞	$\leq 700ns$
t3	0s	∞	
t4	0s	100ns	
t5	0s	100ns	
t6	0s	∞	

Tabla 7.1: Requisitos temporales del intercambio de cuatro fases.

7.2. Principio de funcionamiento

En resumen, este protocolo presenta las siguientes características:

- A diferencia de los sistemas de comunicación tradicionales que transmiten datos de forma continua o en intervalos fijos, el protocolo AER es impulsado por eventos. Esto significa que la información solo se transmite cuando se produce un *spike*.
- En AER, cada neurona en la red tiene una dirección única. Cuando una neurona se activa, en lugar de enviar un valor de señal completo, simplemente se envía su dirección. Esta eficiencia es particularmente útil en redes neuronales grandes, donde solo una pequeña fracción de neuronas puede estar activa en un momento dado.
- El envío de eventos es asíncrono, lo que significa que no está sincronizado con un reloj central. Esto permite que el sistema opere de manera más similar al cerebro, donde diferentes neuronas pueden disparar en momentos diferentes e independientes. Además, múltiples eventos pueden ser procesados en paralelo, aumentando la eficiencia y la velocidad del sistema.
- El protocolo AER es comúnmente utilizado en áreas como la visión artificial, donde se requiere procesamiento rápido y eficiente de la información sensorial. Por ejemplo, en una cámara neuromórfica, cada píxel actuaría como una neurona, enviando eventos solo cuando detecta un cambio en la luz, reduciendo así la cantidad de datos que necesitan ser procesados.
- Una de las principales ventajas del protocolo AER es su eficiencia energética. Dado que solo se transmiten datos cuando es necesario, se reduce significativamente el consumo de energía en comparación con los sistemas de comunicación tradicionales que constantemente envían información, incluso si no hay cambios en los datos.

En esta tesis, el protocolo AER constituye la base para la transmisión

y el procesamiento de la información visual y la comunicación entre los distintos dispositivos neuromórficos. El protocolo AER utilizado en este trabajo fue propuesto en el proyecto de investigación europeo CAVIAR [193].

8

Actuadores Dynamixel AX-12A

*“La inteligencia es la habilidad
de adaptarse a los cambios.”*

Stephen Hawking

8.1 Introducción

La elección de actuadores es un elemento clave en el diseño de locomoción bioinspirada, donde los servomotores emergen como análogos a los actuadores biológicos. La elección adecuada de estos componentes es importante por varias razones:

Primero, la locomoción biológica se distingue por su fluidez y capacidad de adaptación, lo que implica que los servomotores utilizados deben proporcionar un control de movimiento preciso y una respuesta rápida, siendo capaces de imitar los complejos patrones de locomoción presentes en organismos vivos, los cuales involucran cambios en la velocidad y la fuerza.

Además, la correcta selección de servomotores proporciona flexibilidad

en el diseño del robot, lo cual es fundamental en aplicaciones que requieren adaptación a diversos entornos o ejecución de múltiples tareas. Con los servomotores adecuados, un robot puede alcanzar una versatilidad superior, imitando más efectivamente la capacidad de los seres biológicos para adaptarse y operar en diferentes contextos.

Para la implementación de movimiento en la plataforma robótica se han seleccionado los servomotores Dynamixels AX-12A [194], los cuales presentan una versatilidad y precisión muy robustas, lo que los convierte en unos actuadores ideales para la generación de movimiento.

Se han realizado algunos avances utilizando este tipo de actuadores donde demuestran cómo la sincronización de motores en un robot móvil de cuatro extremidades mejora al usar actuadores interconectados en una red de comunicación [195]. Al comparar servomotores de control remoto con los de la serie Dynamixel, estos últimos mostraron un torque superior en la ejecución de movimientos. Además, la resolución del movimiento angular en los Dynamixel es significativamente mejor y la red de comunicación resulta ser mas fiable. Por otro lado, en [196] utilizaron la precisa resolución del movimiento angular de estos motores para mejorar la resolución espacial de un brazo robótico.

Este capítulo se centra, teniendo en cuenta todo lo expuesto anteriormente, en los desafíos e inconvenientes que han presentado estos actuadores, en el protocolo de comunicación utilizado y en los desarrollos que se han realizado para su implementación en la plataforma robótica.

8.2 Características

El motor Dynamixel AX-12A (Figura 8.1) es un actuador diseñado para proyectos avanzados de robótica. Combina varias funciones en un solo módulo, incluyendo un motor de corriente continua, un reductor de engranajes, un microcontrolador y un driver. Este dispositivo es conocido por su capacidad de programación y formación de redes, así como por su control preciso y repetitibilidad [194].



Figura 8.1: Actuador Dynamixel AX-12A. Imagen extraída de [194].

En términos de especificaciones técnicas, el AX-12A tiene un ancho de banda que varía de 7.843 bps a 1 Mbps. Su peso es de aproximadamente 52.9 gramos, con dimensiones de 32 mm×50 mm×40 mm. El motor tiene una resolución de 0.29°, la relación de transmisión es de 254:1 y ofrece un rango de giro que va de 0° a 300°, con la posibilidad de realizar giros continuos.

El par de bloqueo del AX-12A es de 1.5 N·m a 12 V y 1.5 A. En cuanto a la velocidad sin carga, alcanza los 59 rpm en modo rueda a 12V y 54 rpm en modo articulación a 12.0V. Este motor puede operar en un rango de temperaturas de -5°C a +70°C, y requiere un voltaje de entrada de 9.0 a 12.0 V, siendo 11.1 V el recomendado.

8.3 Consideraciones previas

Los motores Dynamixel presentan una comunicación *Half-duplex*, lo cual es una ventaja en cuanto al número de hilos de conexión necesarios¹. Si bien en un primer momento se barajó el uso de un microcontrolador adicional para el control de los actuadores, esto suponía algunos inconvenientes:

- Actualmente, los microcontroladores más comunes, tales como aquellos basados en ARM o ATmega, requieren un buffer triestado externo para facilitar la comunicación *Half-duplex*. Considerando que la plataforma robótica está equipada con 18 motores, se necesitarían al menos 6 chips integrados para lograr un control completo.²
- En segundo lugar, ya que un microcontrolador adicional se encargaría del control de los motores, sería imprescindible implementar un protocolo de comunicación entre este y la FPGA. Esto resultaría en un aumento del tiempo necesario para la comunicación entre ambas plataformas, aunque no se considera crítico para el funcionamiento de la aplicación desarrollada.
- Además, esta topología requeriría el uso de 18 pines en el microcontrolador, dado que no se podría aprovechar la conexión en serie característica de estos servomotores. Esto resultaría en un aumento de los requisitos mínimos necesarios para el microcontrolador seleccionado.
- La necesidad de integrar más componentes electrónicos resultaría en un aumento significativo, aunque no crítico, del consumo de energía, lo que a su vez disminuiría la eficiencia general del sistema.

¹En este protocolo, tanto el receptor (Rx) como el transmisor (Tx) utilizan el mismo bus de comunicación.

²Basándose en la configuración de un IC triestado típico, como el SN74LS241 [197].

8.4. Protocolo de comunicación

Debido a estos inconvenientes y considerando que la arquitectura de procesamiento propuesta ya incluye una FPGA para la comunicación entre los sensores y SpiNNaker, se propone el uso de esta FPGA para hacer de controlador maestro, eliminando los problemas mencionados anteriormente. Para la consecución de este objetivo, se han implementado distintos módulos en *Verilog* que permiten realizar el control de estos servomotores desde la FPGA, sin la necesidad de utilizar un microcontrolador adicional, y utilizando un único bus de comunicación para los 18 actuadores.

8.4 Protocolo de comunicación

Los actuadores Dynamixel AX-12A utilizan un protocolo de comunicación específico, mediante el cual el controlador principal y los servomotores intercambian datos enviando y recibiendo paquetes. Existen dos tipos de paquetes: de instrucciones, enviado por el controlador principal a los servomotores, y de estado, que los servomotores devuelven al controlador principal.

Cuando en el mismo bus hay mas de un actuador conectado, el controlador principal asigna números de identificación (*IDs*) a cada uno de ellos. Gracias a esto, un motor recibirá y ejecutará la instrucción únicamente si el ID que contiene el paquete coincide con el que tiene asignado. Para un diseño correcto de los módulos de Verilog, se han tenido que en cuenta las siguientes consideraciones:

8.4.1 Comunicación Half-Duplex

El protocolo utiliza comunicación serial asíncrona (UART), transmitiendo secuencialmente a través de un solo bus de comunicación, sin necesidad de una señal de reloj para sincronizar el emisor y el receptor. Dado que es un protocolo que no permite utilizar simultáneamente la transmisión y la recepción al mismo tiempo, el controlador principal (FPGA) debe diseñarse para que establezca la dirección de comunicación.

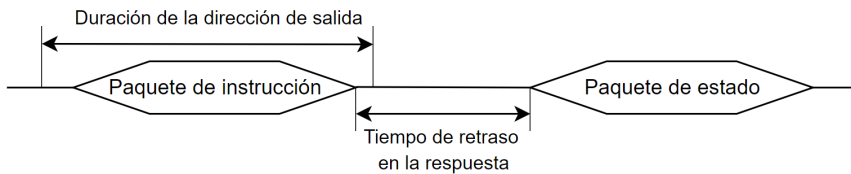


Figura 8.2: Esquema del protocolo Dynamixel.

En la Figura 8.2 se refleja un esquema básico de comunicación entre el controlador y el actuador. El tiempo que espera un actuador en enviar el paquete de estado es de $500\mu s$ ($0,5ms$), por lo que es el tiempo que tiene el controlador para cambiar el estado de transmisión a recepción.

8.4.1.1 Paquete de instrucción

El paquete de instrucción es el que envía la FPGA a los actuadores. La longitud de este paquete varía en función de algunos parámetros como el número de actuadores objetivo o las acciones a realizar, pero la estructura mínima es la siguiente:

Cabecera 1	Cabecera 2	ID	Tamaño	Instrucción	Param 1	...	Param N	Checksum
0xFF	0xFF	Packet ID	Tamaño	Instrucción	Param 1	...	Param N	CHKSUM

Tabla 8.1: Configuración mínima de un paquete de instrucción del protocolo Dynamixel.

- Encabezado: Compuesto por un byte 0xFF seguido de otro byte 0xFF.
- ID: Un único byte que indica el ID del motor que debe recibir el paquete de instrucción y procesarlo.
 - Rango: $0 \sim 253$ ($0x00 \sim 0xFD$), permitiendo la conexión de hasta 254 motores con un único bus de comunicación.
 - Broadcast ID: 254 ($0xFE$), Lo que indica que todos los motores conectados deben recibir y procesar el paquete de instrucción.

8.4. Protocolo de comunicación

- Longitud: Un byte que indica la longitud del paquete, incluyendo el área de parámetros y el byte de chequeo de error.
- Instrucción: Un byte, indicando el tipo de acción que se espera del servomotor (leer, escribir, ping...).
- Parámetros: Datos adicionales para la ejecución de la instrucción.
- Checksum: Byte que comprueba que todos los bytes anteriores se han recibido correctamente. Se calcula de la siguiente manera:
Checksum = $\sim (ID + Longitud + Instrucción + Parametro\ 1 + \dots + Parametro\ N)$

8.4.1.2 Paquete de estado

Una vez que se ha recibido el paquete de instrucción, el actuador involucrado envía a la FPGA un paquete de estado, el cual varía en función de si ha habido un error o si la operación se ha realizado correctamente. En la tabla 8.2 se presenta la configuración mínima de un paquete de estado.

Cabecera 1	Cabecera 2	ID	Tamaño	Error	Param 1	...	Param N	Checksum
0xFF	0xFF	ID	Tamaño	Error	Param 1	...	Param N	CHKSUM

Tabla 8.2: Configuración mínima de un paquete de estado del protocolo Dynamixel.

El funcionamiento de cada byte es similar al del paquete de instrucción, con la única diferencia de que el byte de error proporciona un identificador del error o errores generados.

8.4.1.3 Tiempo entre bytes

El tiempo entre bytes es el retraso máximo que puede producirse entre bytes de un mismo paquete (Figura 8.3). Si el tiempo de retardo es superior a $100ms$, el actuador lo reconoce como un problema en la comunicación y espera de nuevo la siguiente cabecera ($0xFF\ 0xFF$) de un paquete.

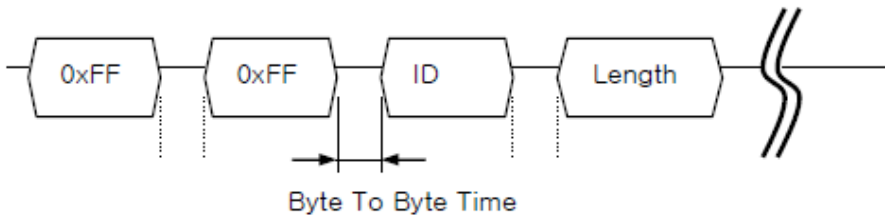


Figura 8.3: Tiempo entre bytes de un paquete.

9

Chip neuromórfico SpiN-Naker

“Lo que sería realmente aterrador es no intentar usar la tecnología para hacer el mundo mejor.”

Sheryl Sandberg

9.1 Introducción

Tal y como se detalló en el capítulo 2, SpiNNaker es un sistema digital diseñado para simular eficientemente grandes redes neuronales que se asemejen en complejidad a un sistema nervioso biológico en tiempo real. En este capítulo se detallarán las características técnicas del chip neuromórfico SpiNNaker, junto con los dos modelos que se han utilizado para la consecución de esta tesis doctoral: los modelos SpiNN-5 y SpiNN-3.

9.2 El chip SpiNNaker

El elemento principal de las máquinas SpiNNaker es el chip SpiNNaker. El chip SpiNNaker es un sistema *multicore* en chip que consta de 18 núcleos de procesador ARM-968. Dentro del chip, estos núcleos están rodeados por una infraestructura de comunicaciones asíncronas, que hacen de la máquina SpiNNaker un sistema globalmente asíncrono y localmente síncrono (GALS). Cada chip también tiene una SDRAM de 128 MBytes, la cual está físicamente montada encima del chip SpiNNaker, tal y como puede apreciarse en la Figura 9.1.

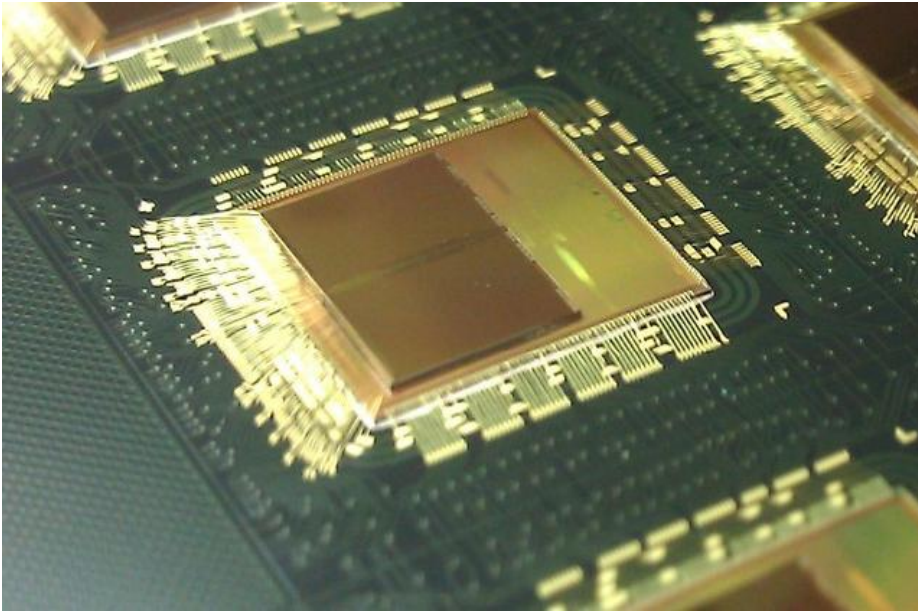


Figura 9.1: Chip neuromórfico SpiNNaker. En la imagen puede apreciarse el chip SpiNNaker, junto con el módulo SDRAM montado encima [198].

La microarquitectura del chip SpiNNaker ha sido diseñada bajo la premisa de que los procesadores son 'gratuitos': el coste real de la computación es la energía. Siguiendo este principio, se han utilizado procesadores integrados ARM-968 y SDRAM DDR móviles, sacrificando en ambos casos parte del rendimiento, pero logrando una eficiencia energética enormemente mejorada.

9.2. El chip SpiNNaker

En la Figura 9.2 se muestra un diagrama del chip SpiNNaker, con los 18 subsistemas de procesamiento ubicados en la periferia, mientras la NoC y los componentes compartidos se ubican en el centro. Normalmente, uno de estos núcleos se utiliza como núcleo monitor para realizar tareas de gestión del sistema, mientras que 16 de ellos se utilizan para respaldar la aplicación y se reserva el último para la tolerancia a fallos y mejora del rendimiento de fabricación.

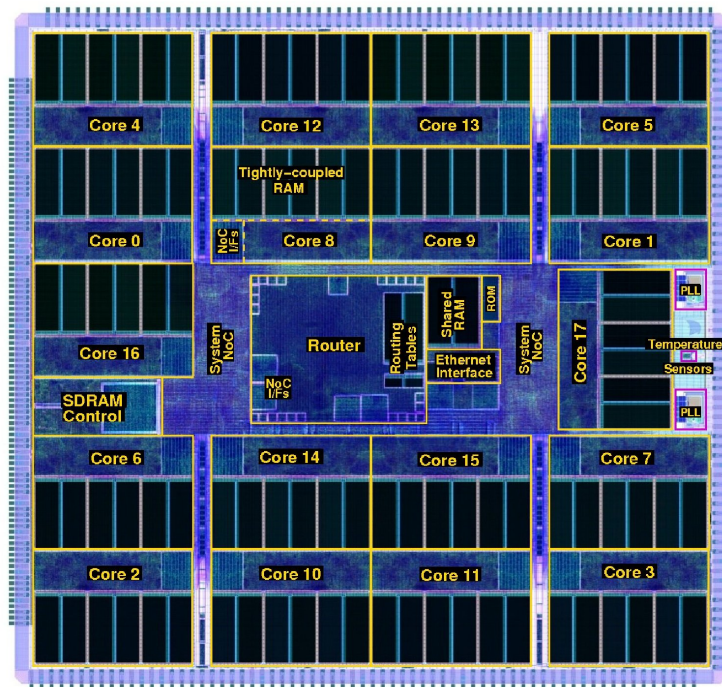


Figura 9.2: Imagen gráfica del chip neuromórfico SpiNNaker [198].

La comunicación entre procesadores se basa en una infraestructura de multidifusión eficiente inspirada en la neurobiología. Utiliza una red de conmutación de paquetes para emular la muy alta conectividad de los sistemas biológicos. Los paquetes son enrutados en origen, es decir, sólo transportan información sobre el emisor y la infraestructura de red es responsable de entregarlos a sus destinos.

El núcleo de la infraestructura de comunicaciones es un enrutador de multidifusión personalizado, diseñado para replicar paquetes de datos

según sea necesario. Esta capacidad es fundamental para llevar a cabo la función de multidifusión, que implica el envío de un mismo paquete de datos a múltiples destinos distintos.

En conclusión, los chips SpiNNaker, con sus seis enlaces bidireccionales interconectados, representan un avance significativo en la construcción de redes con topologías variadas. Aunque su sistema de comunicación exige un uso más elevado de cables, esta particularidad se compensa con una notable eficiencia energética, superior a la de los enlaces síncronos de ancho de banda comparable. Este equilibrio entre complejidad estructural y rendimiento energético subraya el potencial de los chips SpiNNaker en el panorama tecnológico actual.

9.3 Modelo SpiNN-5

El modelo SpiNN-5 consta de una placa equipada con 48 chips SpiNNaker, sumando un total de 864 núcleos de procesador ARM. Estos núcleos se distribuyen de la siguiente manera: 768 están designados para aplicaciones de respaldo, 48 funcionan como procesadores de monitor y los restantes 48 son núcleos de repuesto. En el esquema general de la placa, mostrado en la (Figura 9.3), se pueden observar estas configuraciones.

El modelo SpiNN-5 incorpora dos sensores térmicos, etiquetados como TN y TS. Su función esencial es desactivar la placa en situaciones de sobrecalentamiento. Además, la placa está bajo la supervisión de un BMP (*Board Management Processor*), que juega un papel crucial en la administración de aspectos fundamentales como la gestión de energía, el control del reloj y el reinicio de los chips SpiNNaker y de las FPGAs.

Respecto a las conexiones de red, la placa dispone de dos interfaces Ethernet de 100Mbps, ubicadas en los conectores J14 y J16. La interfaz en J14 se conecta directamente al BMP, mientras que la interfaz en J16 se enlaza con el chip SpiNNaker situado en la esquina inferior izquierda del modelo.

menos adecuado para su implementación en plataformas robóticas móviles, donde la eficiencia energética y el tamaño compacto son aspectos determinantes.

9.4 Modelo SpiNN-3

El modelo SpiNN-3 (Figura 9.4), incorpora cuatro chips SpiNNaker, cada uno con las mismas especificaciones que se han descrito previamente.

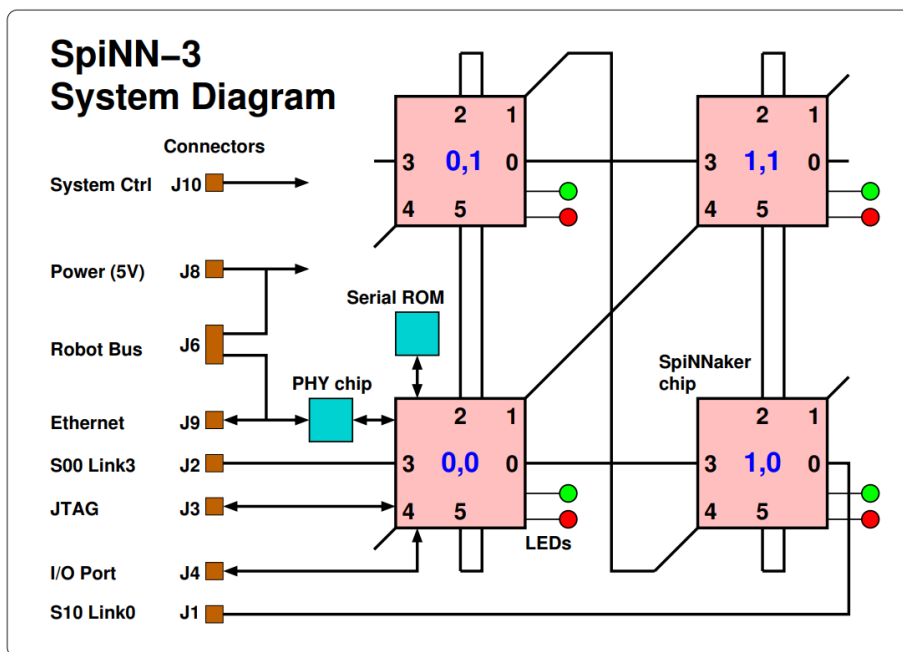


Figura 9.4: Esquema general del modelo SpiNN-3 [198].

Similar al modelo SpiNN-5, el chip designado como (0,0) actúa como el chip maestro. Este chip maestro está dotado de un módulo de ROM en serie que almacena ciertos parámetros de red, como la dirección MAC, y puede emplearse para arrancar el sistema con software especializado. Aunque generalmente los chips se activan desde una ROM interna, el chip maestro se distingue por tener una interfaz Ethernet, suministrada por un chip PHY, que es compatible con velocidades de 10 o 100 Mbps, y se conecta mediante un conector RJ45 estándar.

9.4. Modelo SpiNN-3

El chip maestro también se caracteriza por tener un puerto Ethernet y una conexión JTAG accesible a través de un conector. Cada chip posee dos LEDs, uno rojo y otro verde, controlables por un puerto de E/S, y el chip maestro añade 8 bits adicionales de su puerto de E/S a un conector. La placa se puede reiniciar usando un pulsador, y existe otro conector para permitir el reinicio de forma remota. Además, posee dos enlaces de comunicación disponibles en conectores externos, facilitando la conexión entre placas o con periféricos¹.

Dadas sus compactas dimensiones de apenas 90x100mm, el modelo SpiNN-3 se perfila como la opción ideal para su uso en esta tesis doctoral: su tamaño reducido permite una fácil integración en la plataforma robótica, facilitando así su adaptación. Además, su eficiencia energética es notable, con un consumo de tan solo 4W, lo que lo hace aún más atractivo, ya que la gestión eficiente de la energía es determinante.

¹Específicamente, el enlace 3 del chip (0,0) y el enlace 0 del chip (1,0) están accesibles.

10

Sensor de visión neuromórfico

“La IA no te sustituirá. Lo hará una persona que use IA.”

Santiago Valdarrama

10.1 Introducción

El avance en la ingeniería neuromórfica ha dado lugar a los sensores de visión neuromórficos (DVS), los cuales marcan una revolución en cómo los sistemas computacionales perciben y procesan la información visual. A diferencia de los sensores de visión convencionales, que capturan el mundo como una secuencia de fotogramas compuestos por píxeles, los sensores de visión neuromórficos emiten pulsos en respuesta al contraste temporal cuando perciben algún cambio. Inspirados en los procesos biológicos del sistema visual, estos sensores imitan la dinámica y eficiencia del ojo humano, ya que buscan replicar el comportamiento de ciertas células ganglionares de la retina, las cuales envían al cerebro señales que se convierten en las imágenes que vemos [4].

Es por ello que se ha integrado el sensor de visión dinámico 'Tmptdiff128' en la plataforma robótica para proporcionarle capacidades visuales [4]. A continuación, se detallarán las características específicas de este sensor.

10.2 Características técnicas

El sensor 'Tmptdiff128', mostrado en la Figura 10.1, es un chip sensor CMOS de visión dinámica (DVS) que responde a cambios en la intensidad relativa de la luz. Se destaca por su baja latencia y alta resolución temporal, funcionando eficazmente bajo cualquier nivel de iluminación.

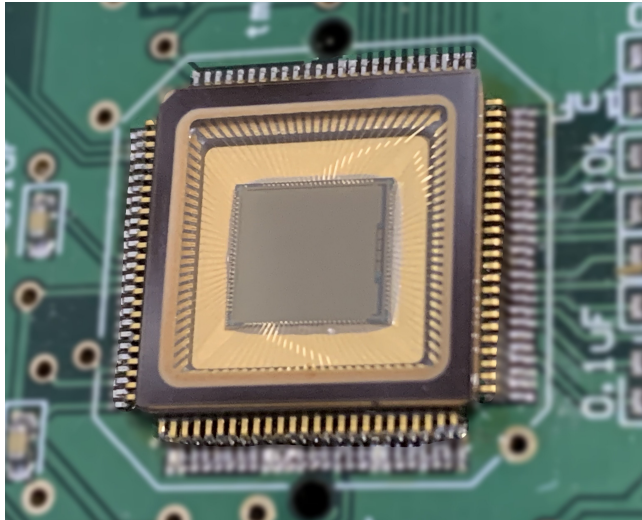


Figura 10.1: Chip neuromórfico 'Tmptdiff128'.

A diferencia de los sensores convencionales, el DVS Tmptdiff128 no utiliza una señal de reloj para realizar la lectura de los píxeles. En lugar de eso, detecta cambios en la intensidad logarítmica de la luz y los comunica instantáneamente como 'eventos'. Estos eventos registran las coordenadas x-y de los píxeles activos en el sensor y se transmiten a través de una interfaz AER. Cada píxel puede generar dos tipos de eventos: 'On', que indica un aumento en la intensidad de la luz, y 'Off', que señala una disminución, como se ilustra en la Figura 10.2.

10.2. Características técnicas

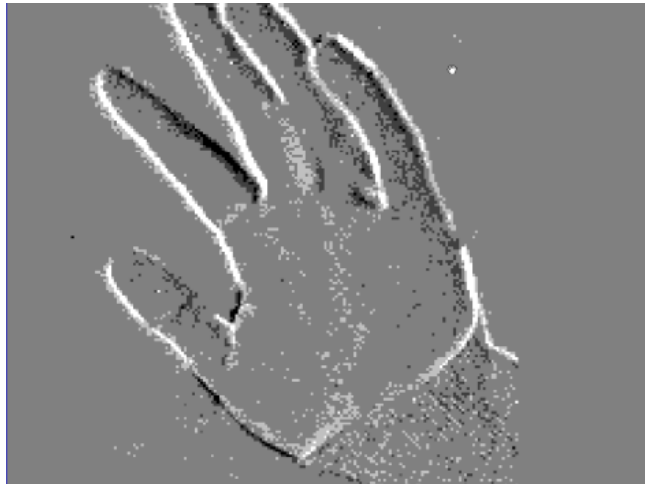


Figura 10.2: Contorno de una mano captada por el DVS en el software jAER. Pueden apreciarse los dos tipos de eventos: eventos "on" (píxeles en blanco), los cuales marcan un aumento relativo de la intensidad lumínica, y eventos "off" (píxeles en negro), marcando una disminución relativa de la intensidad lumínica [4].

Los píxeles de este sensor ajustan su ganancia de manera individual, otorgándole al sensor un amplio rango dinámico intraescena de 120 dB. Como resultado, la salida del sensor no es una imagen convencional, sino una representación abstracta de los bordes en movimiento en la escena, donde el fondo se elimina automáticamente.

Gracias a la integración del preprocesamiento visual en el chip, el procesamiento de datos resulta computacionalmente menos exigente que en los métodos tradicionales de procesamiento de imágenes.

10.3 Principio de funcionamiento

A diferencia de los sensores de imagen convencionales que capturan fotogramas estáticos compuestos por muchos píxeles, los píxeles en un DVS funcionan de manera diferente:

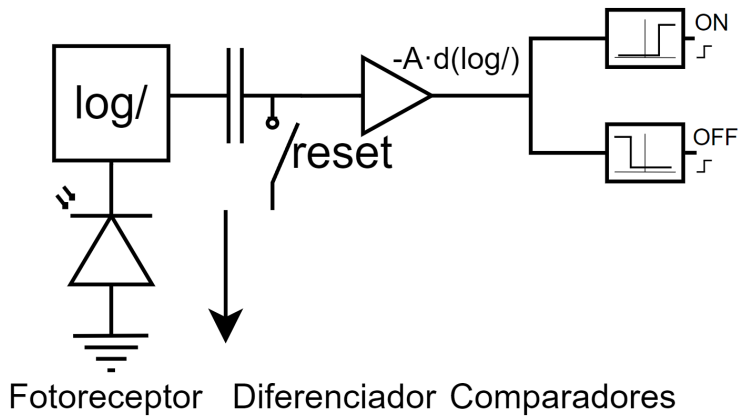
En lugar de capturar y transmitir toda la información de la intensidad de la luz de cada píxel en cada fotograma, los píxeles en un DVS responden solo a los cambios significativos de intensidad en la luz. Cuando un píxel detecta un cambio en la intensidad de la luz que supera un cierto umbral (T), envía la dirección AER asociada a ese píxel. Esta dirección AER incluye un bit de polaridad, el cual indica que el cambio puede ser positivo (aumento de la intensidad) o negativo (disminución de la intensidad).

La dirección AER enviada por cada píxel contiene información sobre qué píxel experimentó el cambio en la intensidad (fila y columna), así como si fue un aumento o una disminución en la intensidad.

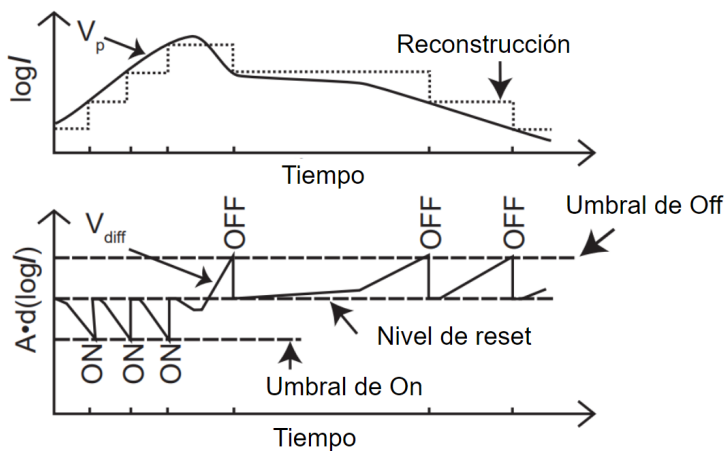
Estos eventos se transmiten de forma asíncrona, lo que significa que no están atados a un fotograma o a una tasa de actualización tradicional. En cambio, se envían tan pronto como ocurre el cambio de intensidad, lo que proporciona una precisión de tiempo mucho mayor en comparación con los sistemas de captura de imagen tradicionales.

En la Figura 10.3a se presenta el esquema abstracto de un píxel, donde se pueden apreciar las distintas fases de las que se compone este circuito: si la luminosidad en ese píxel varía lo suficiente para superar un umbral T , el comparador cambiará su estado. En la (Figura 10.3b) se representan internamente y se emiten los eventos ON y OFF en respuesta a una señal de entrada.

10.3. Principio de funcionamiento



(a) Esquemático del circuito de un píxel.



(b) Principio de operación.

Figura 10.3: **(a)** Esquema abstracto del píxel, que responde con eventos a cambios en la intensidad logarítmica. **(b)** Cómo se representan internamente y se emiten los eventos *ON* y *OFF* en respuesta a una señal de entrada. Imagen basada en [4].

En esencia, el DVS 'Tmptdiff128' (Figura 10.4) es una matriz de 128x128 píxeles, donde cada píxel lleva asociada una coordenada X e Y. Cuando uno de los 16 384 píxeles detecta un cambio en la luminosidad, este genera un paquete AER, el cual contiene la información del píxel activo, junto con la polaridad del mismo.

A modo de ejemplo, en la Figura 10.5 se presenta el mapeo del píxel posicionado en la coordenada (56,78) con polaridad positiva:

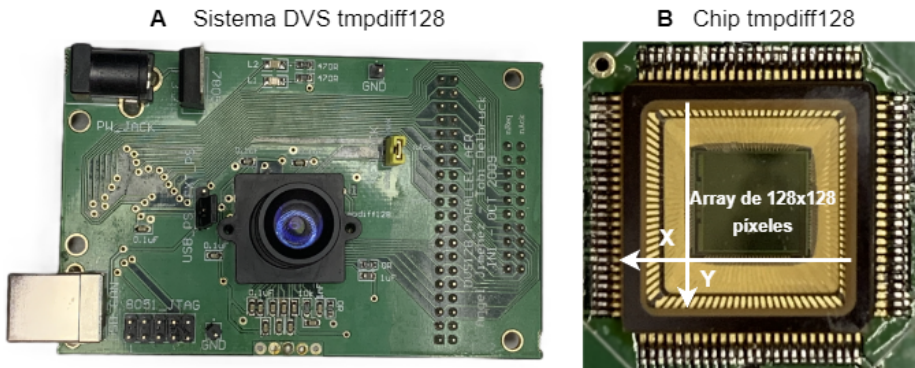


Figura 10.4: Características del sensor de visión dinámico Tmpdiff128. **(A)** DVS con lente e interfaz AER. **(B)** Fotografía del chip. Los píxeles generan eventos AER, compuestos por las coordenadas X e Y, así como con la polaridad del píxel (ON/OFF).

15	14					8	7	6						0	
-	Coordenada X					p	Coordenada Y								
15	14					8	7	6						0	
-	0	1	1	1	0	0	0	1	1	0	0	1	1	1	0

Figura 10.5: Mapeo del píxel situado en la posición (56,78). Bits 0-6: Coordenada Y. Bit 7: Polaridad. Bits 8-14: Coordenada X. Bit 15: Sin asignación.

Teniendo en cuenta estos conceptos, es posible utilizar la retina para monitorizar qué regiones presentan mayores cambios en la luminosidad, lo cual es clave para desarrollar la rutina de seguimiento implementada en esta tesis doctoral.

IV

Métodos

11

Métodos

“Nuestra mayor debilidad radica en rendirnos. La manera más segura de tener éxito es intentarlo una vez más.”

Thomas A. Edison

11.1 Introducción

En este capítulo, se detalla de forma exhaustiva y cronológica los métodos desarrollados durante la tesis doctoral, estructurados en tres fases distintas que marcan el avance progresivo hacia la arquitectura final de la plataforma robótica.

La 1ª fase se centra en el diseño del sCPG. Aquí se exploran los conceptos teóricos subyacentes y se somete a la red resultante a distintas simulaciones. Durante esta etapa inicial, no se utilizan datos reales, permitiendo un enfoque más teórico y experimental para establecer una base sólida para el desarrollo posterior.

En la 2ª fase, se aborda la modificación y optimización de la red, marcando un punto determinante en la investigación. Se implementa la red

en plataformas FPGA y se empieza a integrar el uso de datos en tiempo real del FSR. Este enfoque práctico permite no solo afinar la red, sino también adaptarla para interactuar de manera más efectiva con el entorno real, lo cual es esencial para la aplicación práctica de la plataforma robótica.

Finalmente, la 3ª fase representa la culminación de esta tesis doctoral. Aquí se implementa la red WTA, junto con el sensor de visión neuromórfico, lo que supone un gran avance en la capacidad perceptiva de la plataforma. Se diseñan también los módulos de control de los actuadores y las PCBs de interconexión, elementos clave para la integración y el funcionamiento efectivo del sistema. La fase concluye con la implementación completa de la plataforma robótica, demostrando la viabilidad y eficacia del modelo desarrollado a lo largo de la tesis.

11.2 Fase I

Como primer enfoque de esta tesis doctoral se planteó el diseño de técnicas de neuromodulación y adaptación mediante la simulación de información sensorial externa. Esta aproximación tiene como objetivo principal regular la frecuencia de oscilación de un CPG.

Para lograr esto, se diseñó una estructura neuronal compuesta por dos CPGs interconectados que regulan sus oscilaciones de manera distinta:

- El primero, denominado CPG_{12} , ajusta su frecuencia de oscilación basándose en la información sensorial recibida a través de un FSR simulado.
- Por otro lado, el segundo CPG, denominado CPG_{AB} , regula su actividad rítmica mediante la neuromodulación proporcionada por el primero.

Es importante resaltar que en esta estructura, el CPG_{12} tiene un rol específico como neuromodulador del CPG_{AB} . Esto significa que su función principal es influir en la actividad del CPG_{AB} , pero no se utilizará direc-

11.2. Fase I

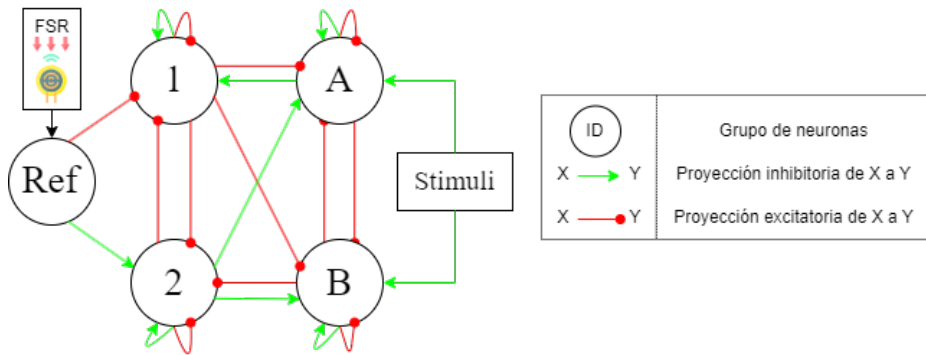


Figura 11.1: Concepto de la topología de la red neuronal adaptativa.

tamente para generar señales de movimiento en la plataforma robótica. Por lo tanto, aunque el CPG_{12} juega un papel determinante en la regulación y modificación del comportamiento del CPG_{AB} , su actividad por sí sola no se traduce en acciones motoras de la plataforma. En cambio, es la salida del CPG_{AB} la que se utiliza efectivamente como señal de movimiento para la plataforma robótica.

11.2.1 Modelo neuronal asociado

El modelo empleado para la implementación de la red neuronal es el modelo LIF, que se detalla exhaustivamente en el capítulo 5. Este modelo se caracteriza por un conjunto de ecuaciones específicas derivadas de las ecuaciones 5.11, 5.12, 5.13 y 5.14, las cuales describen la dinámica de los potenciales de las neuronas en respuesta a las entradas y a su estado anterior, lo que permite simular el comportamiento dinámico de las neuronas en la red:

$$\frac{dV}{dt} = \frac{V_r - V + R(I_{exc} - I_{inh})}{\tau} \quad (11.1)$$

$$\frac{dI_{inh}}{dt} = \frac{-I_{inh}}{\tau_i} \quad (11.2)$$

$$\frac{dI_{exc}}{dt} = \frac{-I_{exc}}{\tau_e} \quad (11.3)$$

Donde V es el potencial instantáneo de la neurona, V_r es el potencial de reposo de la neurona, R la resistencia de la membrana de la neurona, I_{exc} e I_{inh} son las corrientes de excitación e inhibición inyectadas a la neurona, τ es la constante de tiempo que modela la carga y descarga del potencial de la membrana, τ_i la constante de tiempo que modela el valor de la corriente de inhibición inyectada, y τ_e la constante de tiempo que modela el valor de la corriente de excitación inyectada.

Las ecuaciones 11.1, 11.2 y 11.3 ilustran cómo el potencial de membrana y las corrientes de excitación e inhibición decaen con el tiempo. Esto implica que, en ausencia de nuevos estímulos, el potencial de la neurona disminuirá gradualmente hasta alcanzar el nivel de reposo.

11.2.2 Diseño de la red neuronal

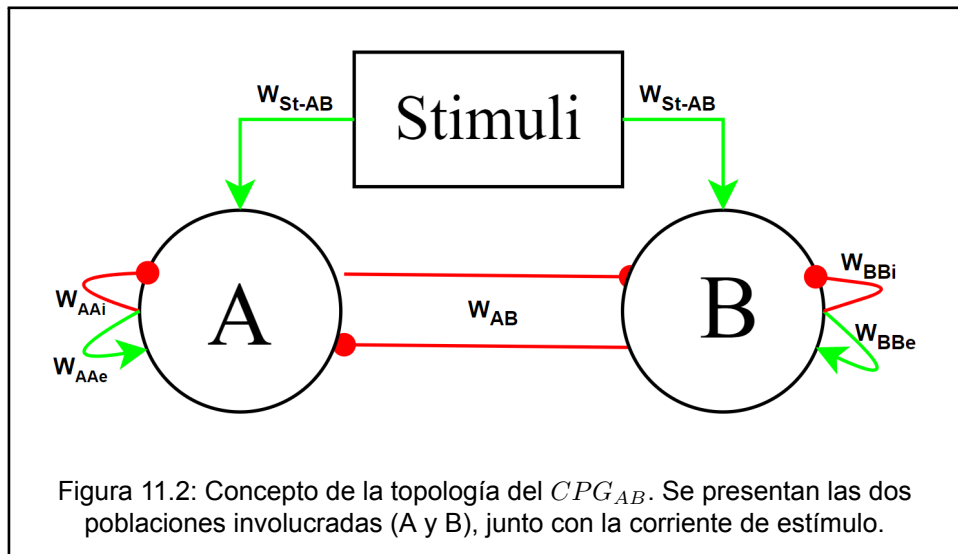
El diseño de la red neuronal podría dividirse en tres apartados bien diferenciados: el diseño del CPG_{AB} , el diseño del CPG_{12} y el diseño de la red de realimentación.

11.2.2.1 Diseño del CPG_{AB}

La Figura 11.2 muestra la estructura del CPG_{AB} , que consiste en dos grupos de neuronas, denominados A y B, cada uno con 500 neuronas. Adicionalmente, cuenta con una corriente de estímulo constante (*Stimuli*). Al igual que en las redes neuronales biológicas, las conexiones entre estas neuronas no tienen una probabilidad del 100 % de conexión [199].

11.2. Fase I

En este modelo, se han implementado probabilidades en las conexiones sinápticas para simular más fielmente las redes neuronales biológicas: hay un 75 % de probabilidad de que se realicen las conexiones inhibitorias y un 25 % de que se realicen las excitatorias para prevenir un exceso de actividad en la red, reduciendo el consumo en una posible implementación en hardware.



Además, las neuronas de una población tienen una probabilidad del 75 % de formar conexiones inhibitorias con neuronas de la otra población. Esta disposición se establece para garantizar una inhibición efectiva de una población cuando la otra está excitada, lo que resulta en una oscilación en la frecuencia de disparo global de ambas poblaciones. Esta interacción es clave para el funcionamiento dinámico del sistema, permitiendo un equilibrio entre la activación y la inhibición en el modelo del CPG_{AB} .

A pesar de que el modelo base para el diseño de la red es el modelo LIF, mencionado en el apartado anterior y empleado por el CPG_{12} , el CPG_{AB} introduce una sutil modificación a este modelo. Este cambio consiste en inyectar una corriente constante externa a ambas poblaciones para garantizar una oscilación mínima en el CPG, convirtiendo a las neuronas

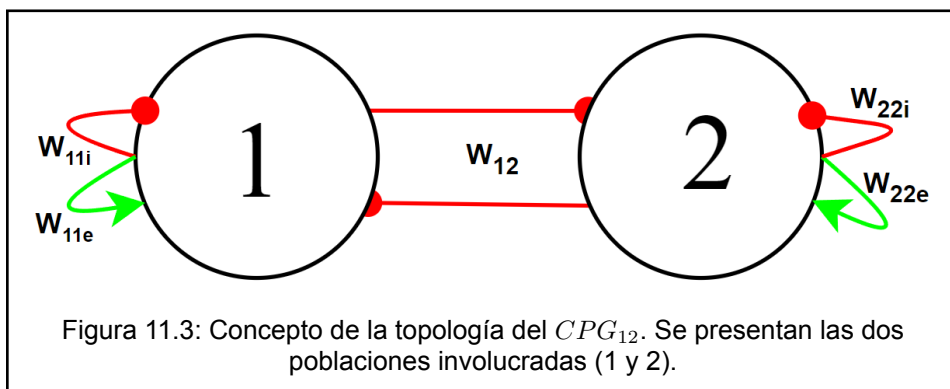
de ambas poblaciones en neuronas tónicas¹. Como resultado de esta alteración, la expresión matemática 11.1 del modelo LIF se modificaría para reflejar este cambio:

$$\frac{dV}{dt} = \frac{Vr - V + R(I_{exc} - I_{inh} + I_{st})}{\tau} \quad (11.4)$$

Donde I_{st} sería la corriente mínima inyectada a las neuronas de las poblaciones A y B, con un valor suficiente para asegurar el patrón rítmico en el CPG sin que las conexiones inhibitorias lleguen a impedir el estímulo en las neuronas.

11.2.2.2 Diseño del CPG_{12}

Por otro lado, el CPG_{12} (Figura 11.3) también está compuesto por dos poblaciones de 500 neuronas cada una, interconectadas de la misma forma que el CPG_{AB} .



Este CPG, a diferencia del anterior, no busca mantener una oscilación constante entre ambas poblaciones, sino provocar que una de ellas domine sobre la otra en función de la información obtenida del sensor y de la realimentación del CPG principal.

¹En neurociencia, el término 'tónico' se utiliza para describir un patrón de señales o actividad neuronal que es constante o sostenida a lo largo del tiempo, a diferencia de una actividad 'fásica', que es breve y a menudo responde a estímulos específicos.

11.2. Fase I

11.2.2.3 Diseño de la población de referencia (P_{Ref})

La población de referencia (P_{Ref}), se configura como una distribución de *Poisson*² con una frecuencia variable en función de la tensión medida por el sensor FSR en un momento dado. P_{Ref} está formada por 50 neuronas, y estas se conectan con las poblaciones P_1 y P_2 . La probabilidad de conexión entre P_{Ref} y estas poblaciones se ajusta según los valores proporcionados por el FSR, siguiendo el esquema ilustrado en la Figura 11.4.

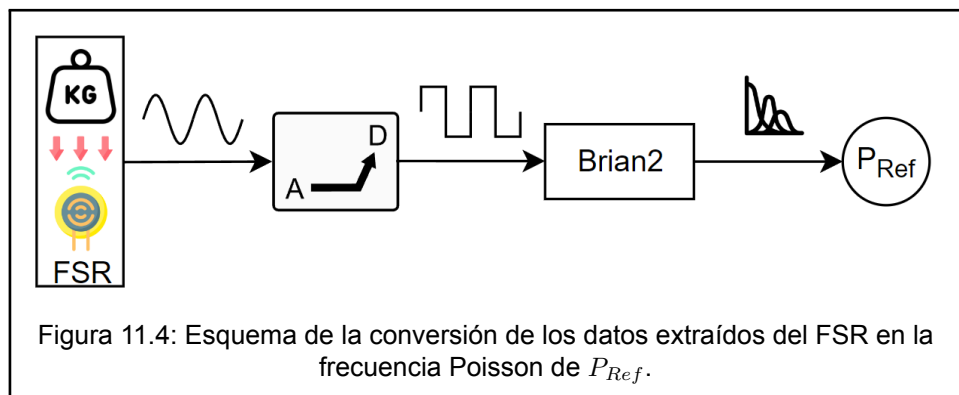


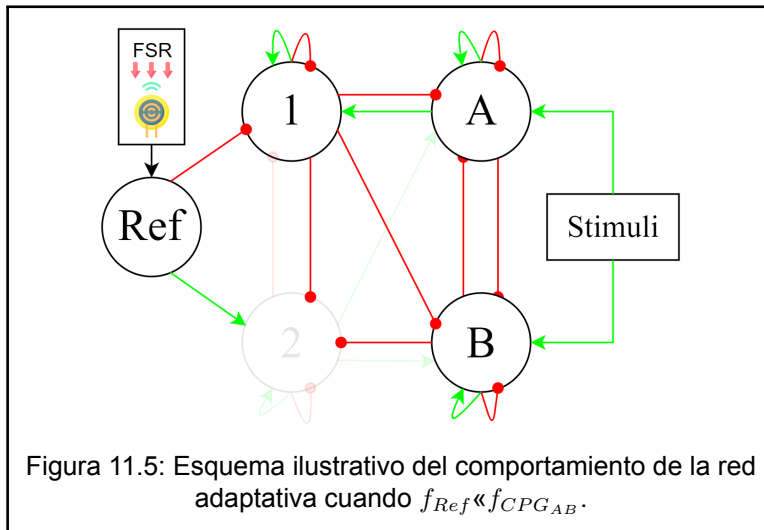
Figura 11.4: Esquema de la conversión de los datos extraídos del FSR en la frecuencia Poisson de P_{Ref} .

Por lo tanto, siguiendo el esquema de la Figura 11.1, se presentan los dos casos extremos que pueden producirse en la red neuronal:

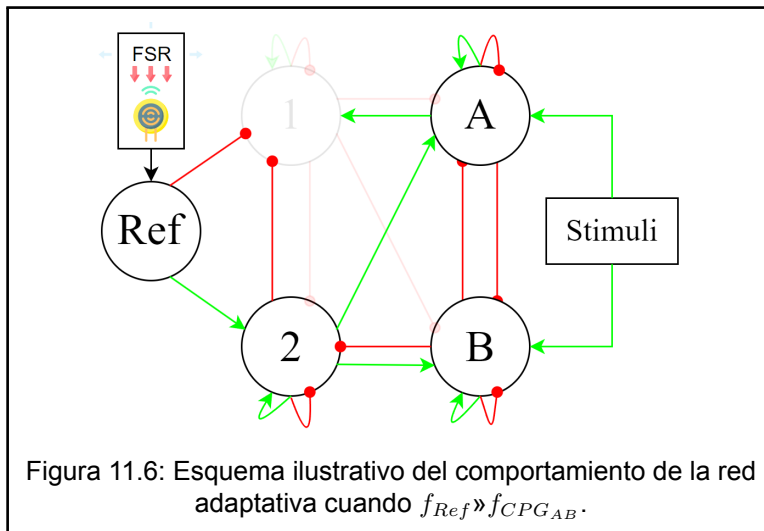
El sensor arroja el valor mínimo establecido. Bajo estas circunstancias, donde $f_{P_{Ref}} \ll f_{CPG_{AB}}$ (Figura 11.5), las 50 neuronas de P_{Ref} disparan con una frecuencia de 10Hz, por lo que tendrán nula influencia sobre las poblaciones del CPG_{12} . En esta situación, dado que P_1 está más estimulada debido a la influencia de P_A y mínimamente inhibida por P_{Ref} , aumenta su actividad, lo que resultará a su vez en la inhibición del CPG_{AB} , llevando a una disminución de la frecuencia del mismo, pero sin llegar a la inactividad, gracias a la corriente de estímulo I_{St} .

El sensor arroja el valor máximo establecido. En este caso, donde $f_{P_{Ref}} \gg f_{CPG_{AB}}$ (Figura 11.6), las neuronas disparan con una frecuencia

²La distribución de *Poisson* es una distribución de probabilidad que modela la tasa de eventos discretos en un intervalo de tiempo específico [200].



máxima de 171Hz a las poblaciones del CPG_{12} y, dado que la excitación recibida es mucho mayor que la inhibición que incide sobre P_2 por parte de P_B , el CPG_{AB} será excitado, haciendo que aumente su frecuencia.

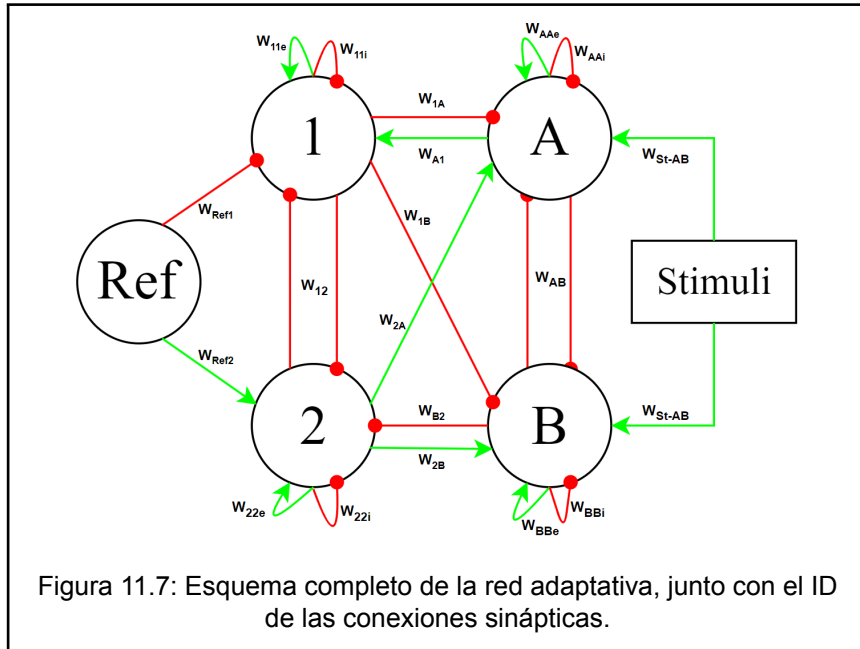


De igual forma, los valores intermedios aumentarán o disminuirán la frecuencia del CPG_{AB} , en función de la presión ejercida en el FSR.

11.2. Fase I

11.2.2.4 Estructura y pesos sinápticos

Finalmente, en la Figura 11.7 se presenta el diseño de la red completa para su simulación en Brian2, junto con la identificación de cada conexión neuronal.



Adicionalmente, en la tabla 11.1 se proporcionan el valor de los pesos sinápticos asociados a cada conexión neuronal.

Peso sináptico	Valor
W_{AB}	0,5 nA
$W_{AAe}, W_{BBe}, W_{11e}, W_{22e}$	4 nA
$W_{AAi}, W_{BBi}, W_{11i}, W_{22i}$	1,5 nA
W_{Ref1}, W_{Ref2}	1 nA
$W_{A2}, W_{A1}, W_{1A}, W_{1B}, W_{2A}, W_{2B}$	0,1 nA
W_{12}	3 nA

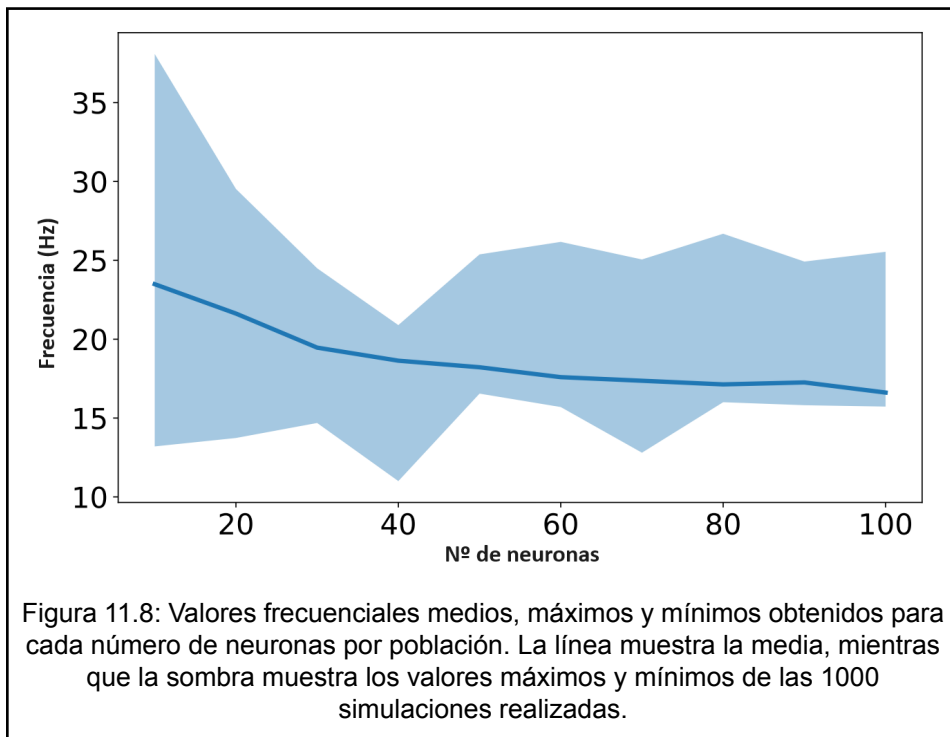
Tabla 11.1: Valores de los pesos sinápticos de la red de adaptación.

11.3 Fase II

Durante la primera aproximación a la implementación final, se realizó el diseño y la simulación de la red adaptativa con el objetivo de validar su funcionamiento. En etapas posteriores, enfocadas en la implementación de esta red en hardware neuromórfico, se hizo esencial no solo optimizar el número de neuronas en cada población, sino también ajustar sus parámetros y desarrollar el diseño de la lógica digital para la FPGA.

11.3.1 Optimización de la red neuronal

Inicialmente, se ejecutaron 1000 simulaciones de manera redundante en esta fase para establecer el número óptimo de neuronas por población requerido para alcanzar una oscilación estable en la salida del CPG_{AB} . Los resultados de estas simulaciones, incluyendo los valores máximos, mínimos y medios de la frecuencia de oscilación del CPG_{AB} , se presentan de manera detallada en la Figura 11.8.



11.3. Fase II

Es evidente que el incremento en el número de neuronas conduce a una mayor estabilidad en la frecuencia del CPG, así como a una reducción en la desviación estándar. Se observa que, a partir de un umbral de 40 neuronas, los patrones de oscilación se vuelven más estables. Al alcanzar o superar las 100 neuronas, el patrón de oscilación del CPG se estabiliza completamente, logrando una desviación estándar de tan solo 1,5 Hz. Debido a estos resultados, se optó por utilizar 100 neuronas por población en la configuración de la red.

11.3.2 Optimización de los parámetros neuronales

Por otro lado, se ejecutaron 1623 simulaciones adicionales hasta que se alcanzó el objetivo de ajustar con precisión los parámetros neuronales. Este proceso tenía como finalidad incrementar la diferencia entre los valores frecuenciales mínimos y máximos en el CPG. Al lograrlo, se mejoraría significativamente la capacidad de adaptación de la plataforma robótica a diferentes superficies. Los parámetros resultantes de estas simulaciones, que reflejan los ajustes realizados, se detallan en la tabla 11.2.

Parámetros neuronales	Valor
V_{reset}	$-55,0 \text{ mV}$
V_{rest}	$-55,0 \text{ mV}$
V_{Th}	$15,0 \text{ mV}$
τ_m	$6,0 \text{ ms}$
τ_{sync_e}	$5,0 \text{ ms}$
τ_{sync_i}	$8,75 \text{ ms}$
C_m	$0,1875 \text{ nF}$
Δ_t	$1,0 \text{ ms}$
I_{bias}	$2,2 \text{ nA}$

Tabla 11.2: Parámetros neuronales utilizados en la fase II.

11.3.3 Sistema digital

Para la comunicación entre los sensores y la plataforma neuromórfica se ha utilizado una placa de desarrollo FPGA Nexys 4 DDR [201] (Figura 11.9). Esta placa está equipada con una FPGA Xilinx Artix-7 XC7A100T-1CSG324C, la cual ofrece una capacidad de lógica programable de 100.000 celdas y 240 slices DSP, proporcionando suficiente potencia de procesamiento para implementar la arquitectura de procesamiento.

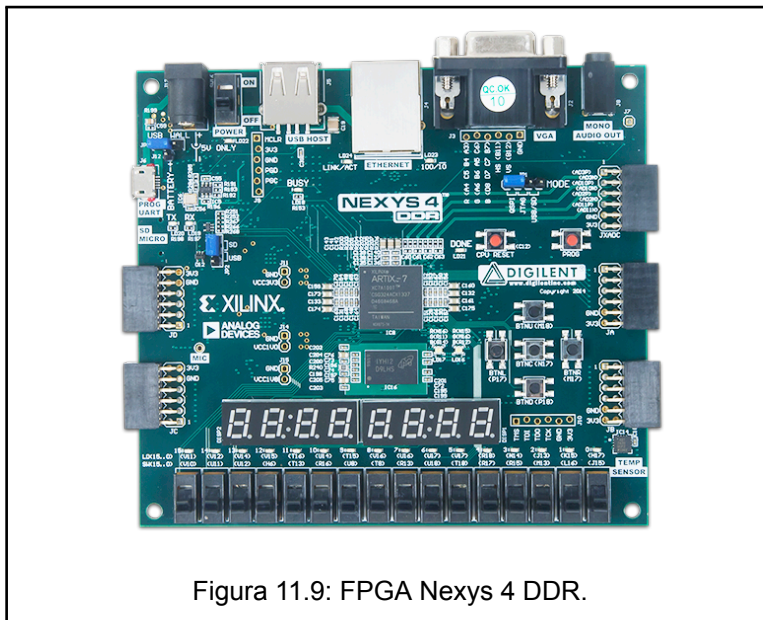


Figura 11.9: FPGA Nexys 4 DDR.

Este sistema tratará el valor obtenido de los FSR y traducirá esa información digital a *spikes*, para posteriormente enviarlos a través de los *SpINNLinks* [202] hasta *SpINNaker*. Una vez procesados por la red neuronal, esta enviará los *spikes* producidos por el sCPG a la FPGA, la cual modificará la velocidad de movimiento de la plataforma robótica en función de la respuesta del sCPG.

Para el envío de estos *spikes* es posible utilizar tanto PWM como PFM, ofreciendo cada uno de ellos un enfoque diferente en la transmisión de información mediante pulsos. Mientras que en PWM la información se codifica en el ancho de los pulsos, en PFM la información se transmite a

11.3. Fase II

través de la variación de la frecuencia de los pulsos. A continuación, se estudiará brevemente cuál de estos métodos es el más acertado para optimizar la eficiencia y precisión en la transmisión de la señal, teniendo en cuenta las características específicas del sistema de control de la plataforma robótica.

11.3.3.1 PWM VS PFM

La modulación por frecuencia de pulso (PFM) y la modulación por ancho de pulso (PWM) son dos técnicas distintas para transmitir información a través de canales de comunicación. En esencia, ambas modulaciones trabajan modificando una señal portadora en función de una señal de mensaje, donde reside la información esencial. Esta modificación resulta en una nueva señal, adecuadamente adaptada para su transmisión.

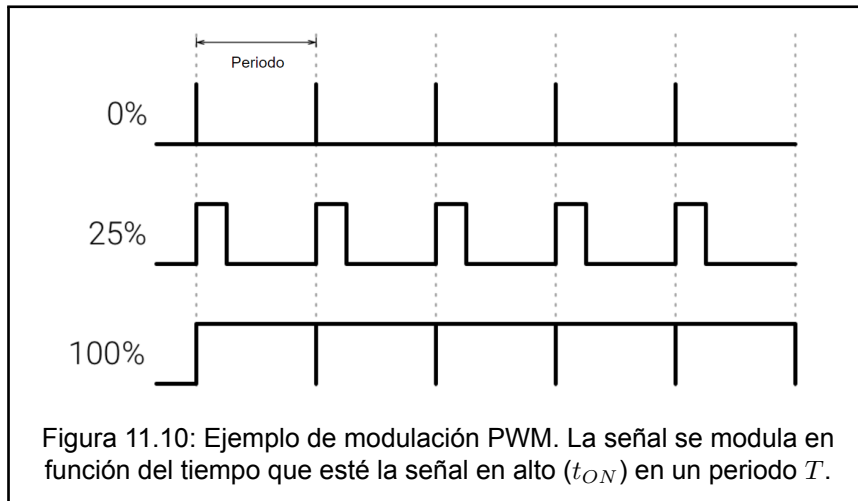
Específicamente, tanto PFM como PWM se utilizan comúnmente para controlar actuadores, incluyendo motores de corriente continua, servomotores y motores paso a paso. De las dos, PWM es la más utilizada en aplicaciones de control motor. Su uso es ampliamente reconocido y se ha convertido en el método predilecto en la mayoría de las situaciones que involucran el control de motores, respaldado por numerosas fuentes.

Por otro lado, PFM, aunque efectiva, no ha alcanzado el mismo nivel de reconocimiento y uso generalizado, por lo que es menos común encontrar dispositivos que implementen modulación PFM.

11.3.3.1.1 PWM

La modulación por ancho de pulso es una técnica de modulación utilizada para codificar un mensaje en una señal de pulsos. Esta técnica se basa en variar el ancho (duración) de los pulsos de una señal periódica (generalmente una onda cuadrada) para transmitir información o para controlar la cantidad de energía que se envía a una carga.

El principio fundamental de la modulación PWM es el ajuste de la propor-



ción del tiempo durante el cual la señal está en su nivel alto en relación con el tiempo en que está en su nivel bajo (Figura 11.10). Esta proporción se conoce como ciclo de trabajo, y es lo que define la cantidad de energía o información transmitida. En aplicaciones prácticas, PWM se utiliza ampliamente para controlar motores, regular la intensidad de luces LED, y en sistemas de comunicación para transmitir señales de manera eficiente. La ventaja principal de PWM es que permite controlar la cantidad de energía que se entrega a un dispositivo sin generar mucha pérdida de energía en forma de calor, lo cual es especialmente útil en aplicaciones de electrónica de potencia.

Para controlar motores mediante PWM dentro de una SNN, primero se integra la señal de salida del pulso durante un periodo fijo T_{PWM} . Este proceso determina el valor que se establecerá en el generador de PWM, que se sitúa antes de la etapa de potencia. A continuación, el generador de PWM ajusta el ciclo de trabajo basándose en este valor integrado, permitiendo así un control preciso del motor, aunque introduciendo latencia debido al proceso de integración.

11.3.4 Modulación por frecuencia de pulsos (PFM)

La modulación por frecuencia de pulsos modifica la frecuencia de un tren de pulsos para controlar la energía transferida a una carga. A diferencia de la modulación PWM, que mantiene una frecuencia constante y varía el ancho de los pulsos, PFM varía la frecuencia de los pulsos mientras mantiene constante su ancho.

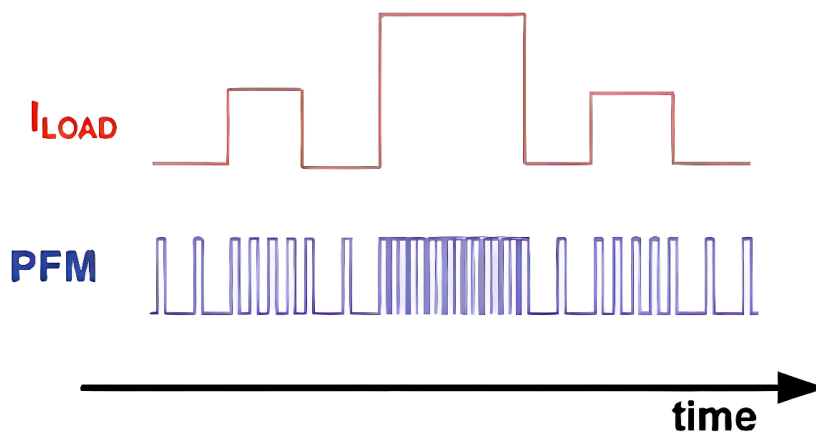


Figura 11.11: Ejemplo de modulación PFM. La señal se modula en función de la frecuencia de la señal.

Aunque la modulación PFM es menos conocida y utilizada en la actualidad, se ha considerado la más adecuada para cuantificar los *spikes* generados por una red neuronal. Esto se debe a que la salida de SpiN-Naker se compone de direcciones AER distribuidas en el tiempo. La manera más simple de contabilizar los spikes que llegan a la FPGA en un intervalo de tiempo específico es mediante el uso de PFM. A través de la FPGA, es posible identificar cuáles y cuántas direcciones AER se reciben en un período determinado. Esto permite, a su vez, ajustar el patrón de movimiento o la orientación de la plataforma robótica basándose en esta información.

11.3.4.1 Relación biológica

Al revisar los estudios biológicos en neurociencia, se observa que no existe una preferencia clara ni directrices específicas para elegir entre PWM y PFM en la toma de decisiones. Sin embargo, diversos autores en el ámbito de la ingeniería consideran que la modulación PFM se alinea más estrechamente con los procesos naturales [203, 204].

En términos biológicos, los músculos están inervados por neuronas motoras tipo α y γ , donde las neuronas motoras α se encargan de las fibras musculares extrafusales y las γ de las intrafusales [205]. Además, la actividad muscular está directamente relacionada con la frecuencia de descarga de las motoneuronas, según lo indicado en la referencia [206].

Dado este contexto biológico, se puede argumentar que la modulación PFM representa un enfoque más adecuado y biológicamente coherente para el control de motores en el contexto de las redes neuronales pulsanter.

11.3.4.2 Diseño del sistema digital

El sistema digital completo se diseñará utilizando la FPGA previamente mencionada, que implementará un código HDL. Este código permite la creación de modelos de circuitos digitales a través de la especificación de las conexiones entre sus componentes. La Figura 11.12 ilustra un esquema general y simplificado del módulo superior que se implementa en la FPGA. El procedimiento se desarrolla de la siguiente manera:

Inicialmente, el módulo *A/D Converter* transforma la señal analógica (de 0 a 5V) en una señal digital de 8 bits (que varía de 0 a 255), representando el valor instantáneo emitido por el FSR. Luego, el módulo *PFM Generator* crea una señal de reloj variable (FSR Clock), que fluctúa entre 10Hz (para un valor de 0V) y 171Hz (para 5V). Posteriormente, el módulo *AER in* produce *spikes* continuos con una frecuencia que depende de la señal *FSR Clock*. Estos *spikes* se transmiten a través de los

11.3. Fase II

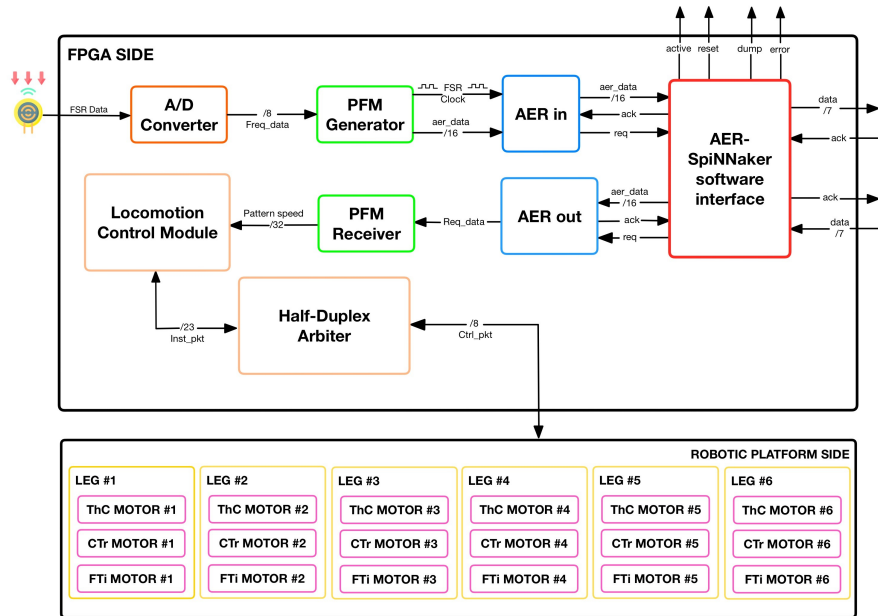


Figura 11.12: Esquema del módulo principal implementado en la FPGA para el control de la plataforma robótica.

SpiNNLinks hasta *SpiNNaker*.

Después de que la red neuronal procese los *spikes* entrantes (como se detalla en la subsección 11.2), se inicia el proceso inverso. El módulo *AER out* recibe los *spikes* del sCPG y los procesa. A continuación, este módulo envía una señal *Req_data* al módulo *PFM Receiver*, estableciendo un enlace entre las señales *req* y *ack* en la comunicación *SpiNNaker-FPGA*. La señal *Req_data* se activa solo cuando se recibe una señal *req* y se envía un *ack*, asegurando la correcta recepción de un *spike* (ver capítulo 7). Esta funcionalidad permite el monitoreo externo de la frecuencia del sCPG fuera de *SpiNNaker*.

Para concluir, los módulos *Locomotion Control Module* y *Half-Duplex Arbiter* gestionan el protocolo de comunicación entre los actuadores y la FPGA, modulando su velocidad de acuerdo con el proceso descrito.

11.4 Fase III

Concluida la etapa de optimización e implementación en hardware neuromórfico, la tercera y última fase de esta tesis representa el punto culminante de esta tesis doctoral. En esta fase, se aborda la integración y aplicación práctica de todos los elementos desarrollados previamente, enfocándose en la implementación de la red WTA y la incorporación del sensor de visión neuromórfico. Este avance significativo en la percepción sensorial de la plataforma se complementa con el diseño y creación de PCBs para la interconexión eficiente de los distintos componentes del sistema.

El objetivo principal de este último enfoque es demostrar no solo la viabilidad técnica, sino también la eficacia operativa del modelo desarrollado. A través de la integración de los elementos de neuromodulación, adaptación sensorial y control físico, esta fase busca establecer un paradigma en la robótica neuromórfica que sea capaz de interactuar de manera efectiva y autónoma en un entorno real. La implementación completa de la plataforma robótica, con su sistema de percepción y control, no solo marca la conclusión de este proyecto de tesis, sino que también sienta las bases para futuras investigaciones y aplicaciones en el campo de la robótica y la inteligencia artificial.

11.4.1 Ajuste fino de los parámetros neuronales

En la fase final del proceso, se realizó un nuevo ajuste de los pesos sinápticos y de los parámetros de todas las poblaciones neuronales involucradas. Este ajuste implicó la realización de más de 5000 iteraciones adicionales, con el objetivo principal de identificar aquellos parámetros y pesos que facilitan un funcionamiento óptimo de la red neuronal en su totalidad. Este proceso contribuyó significativamente a la reducción, casi a la eliminación completa, de los casos en los que se presentaban simu-

11.4. Fase III

laciones con inestabilidad persistente³. Adicionalmente, se efectuaron cambios en las nomenclaturas de las poblaciones dentro del CPG_{12} : P_1 fue renombrada como P_{INH} y P_2 como P_{EXC} . Como resultado de estos cambios, el CPG_{12} también será renombrado a CPG_{EI} . Estas modificaciones tienen el objetivo de mejorar la claridad en cuanto a las funciones específicas que estas poblaciones desempeñan dentro de la red.

En cuanto a los cambios en las características de la red, se establecieron parámetros neuronales independientes para las poblaciones del CPG_{AB} , CPG_{EI} y P_{Ref} , con el fin de eliminar los casos de inestabilidad persistente en la gran mayoría de simulaciones (en concreto, se pasó de un 70 % de éxito en la fase II a un 98 % en la fase III⁴). A continuación, en las tablas 11.3 y 11.4 se presentan los cambios realizados en los parámetros neuronales:

Fase II			Fase III	
Parámetros neuronales	Valor		Parámetros neuronales	Valor
V_{reset}	-55.0 mV		V_{reset}	-70.0 mV
V_{rest}	-55.0 mV		V_{rest}	-65.0 mV
V_{Th}	15.0 mV		V_{Th}	-50.0 mV
τ_m	6.0 ms	→	τ_m	6.0 ms
τ_{Sync_e}	5.0 ms		τ_{Sync_e}	7.0 ms
τ_{Sync_i}	8.75 ms		τ_{Sync_i}	17.1 ms
C_m	0.1875 ms		C_m	0.25 ms
τ_{refrac}	2.0 ms		τ_{refrac}	0.001 ms
I_{bias}	2.2 nA		I_{bias}	0.1 nA

Tabla 11.3: Diferencia en los parámetros neuronales entre la fase II y la fase III para el CPG_{AB} y P_{Ref} .

La implementación de cambios en las poblaciones no solo resultó en una mayor tasa de éxito, sino que también mejoró significativamente la

³La expresión 'inestabilidad persistente' en este contexto significa que el CPG no logra alcanzar una oscilación estable durante el tiempo establecido en la simulación actual.

⁴Se realizaron 400 simulaciones adicionales con los nuevos parámetros neuronales, obteniendo 393 estables y 7 con inestabilidad persistente.

Fase II		→	Fase III	
Parámetros neuronales	Valor		Parámetros neuronales	Valor
V_{reset}	-55.0 mV		V_{reset}	-65.0 mV
V_{rest}	-55.0 mV		V_{rest}	-65.0 mV
V_{Th}	15.0 mV		V_{Th}	-64.91 mV
τ_m	6.0 ms		τ_m	0.1 ms
τ_{Synce}	5.0 ms		τ_{Synce}	0.1 ms
τ_{Synci}	8.75 ms		τ_{Synci}	0.1 ms
C_m	0.1875 ms		C_m	0.1 ms
τ_{refrac}	2.0 ms		τ_{refrac}	0.0 ms
I_{bias}	0.0 nA		I_{bias}	0.0 nA

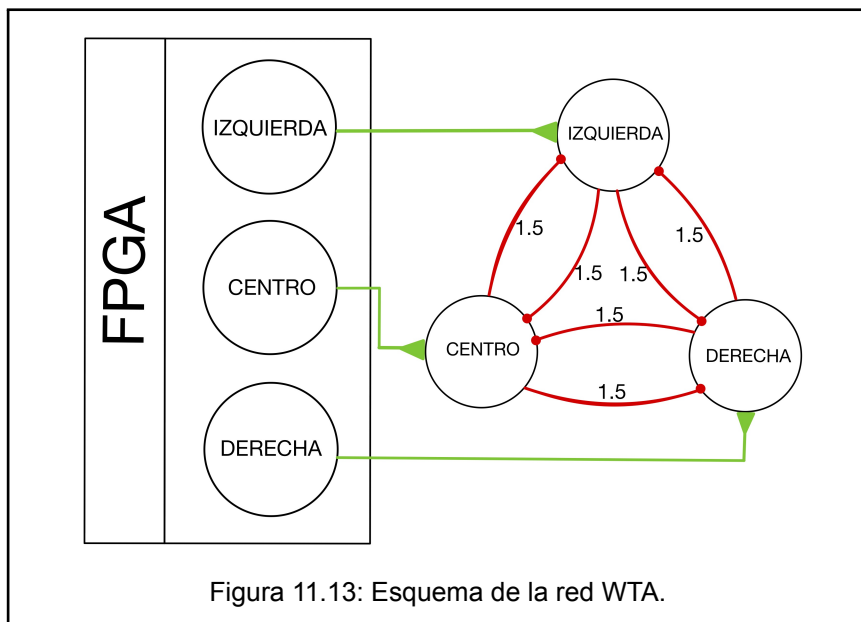
Tabla 11.4: Diferencia en los parámetros neuronales entre la fase II y la fase III para el CPG_{EI} .

respuesta de la red neuronal a las señales de entrada del FSR. Esto se debe principalmente a la adaptación de los parámetros neuronales en el CPG_{EI} para asegurar que la relación entrada:salida en estas poblaciones sea de 1:1.

Con esta configuración, se garantiza que el comportamiento de las poblaciones neuronales esté estrechamente alineado y sea coherente con la entrada proporcionada por el FSR. Además, también es importante destacar que se ha reducido en un 95.45 % la corriente inyectada en el CPG_{AB} , pasando de 2.2 nA a tan solo 0.1 nA.

11.4.2 Diseño de la red Winner-Take-All

Las neuronas de la red WTA reciben la actividad generada por la retina DVS y su salida será utilizada para seleccionar la dirección a seguir por la plataforma robótica. Estas poblaciones incorporan los mismos parámetros que las del CPG_{AB} , a excepción de la corriente inyectada (tabla 11.5). Por este motivo, la actividad o inactividad de estas neuronas se debe únicamente a la actividad de entrada que reciban estas poblaciones de la FPGA. La topología de esta red puede verse en la Figura 11.13, y su interacción con el DVS es la siguiente:



Dado que la retina es una matriz de 128×128 píxeles, para seleccionar la dirección de la plataforma robótica se ha decidido dividir la coordenada X en tres ventanas: los primeros 42 píxeles están vinculados a la dirección derecha, los 44 siguientes al centro y los 42 últimos a la izquierda. De esta forma, cuando un píxel detecte cambios en la luminosidad, se enviará a SpiNNaker la dirección AER asociada a ese píxel. Dependiendo de su coordenada X , esta dirección se utilizará para excitar una de las tres poblaciones en SpiNNaker. Los pesos de las conexiones sinápticas

están configurados para que la población que genere mas actividad sea la ganadora e inhiba fuertemente a las dos anteriores; gracias a esto, a la plataforma robótica solo se transmitirán los eventos AER de una única población ganadora a la vez, lo que le permitirá decidir qué dirección debe seguir.

Parámetros neuronales	Valor
V_{reset}	-70.0 mV
V_{rest}	-65.0 mV
V_{Th}	-50.0 mV
τ_m	12.5 ms
τ_{Sync_e}	7.0 ms
τ_{Sync_i}	17.1 ms
C_m	0.25 nF
τ_{refrac}	0.001 ms
I_{bias}	0.0 nA

Tabla 11.5: Parámetros neuronales de la red WTA implementada en SpiNNaker.

11.4.3 Implementación del sensor DVS en el sistema

Tal y como se detalló en la sección 10.3, el sensor de visión dinámico (DVS) incluye una matriz de 128x128 píxeles, donde cada píxel lleva asociada una coordenada X e Y, tal y como se muestra en la Figura 11.14.

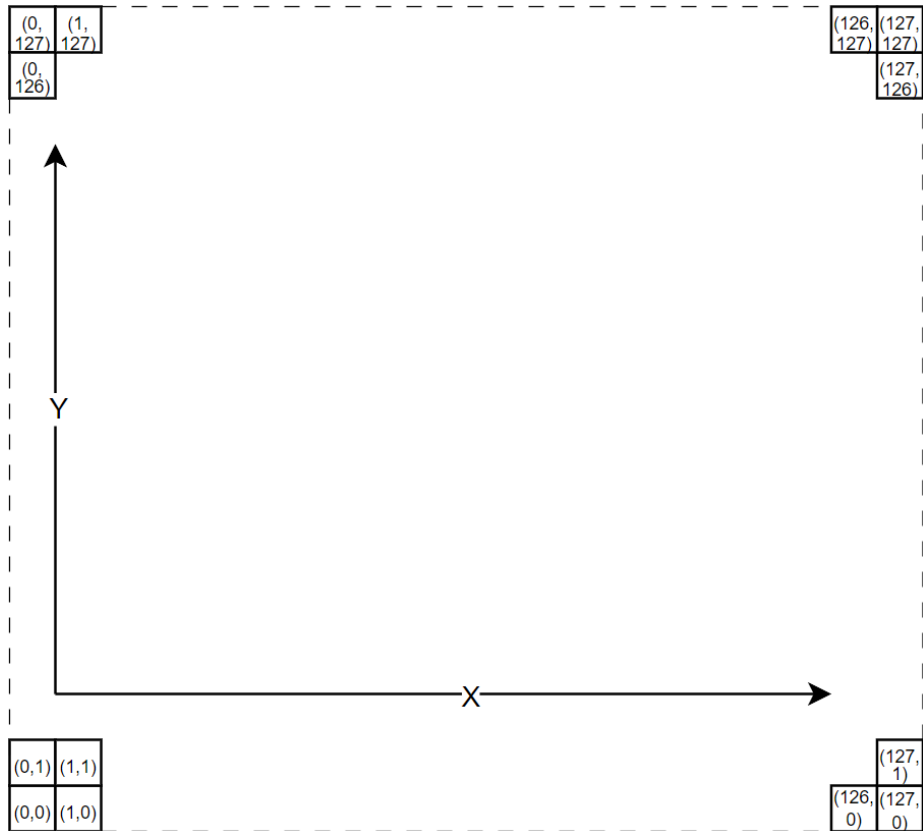


Figura 11.14: Esquema de distribución de píxeles del sensor de visión dinámico.

11.4.3.1 Concepto de diseño

La implementación del DVS tiene como objetivo principal que la plataforma robótica identifique y se mueva hacia la dirección con la mayor cantidad de estímulos visuales. Para lograr esto, se optó por dividir el eje X en tres zonas claramente diferenciadas: izquierda, centro y dere-

cha (Figura 11.15). Basándose en el volumen de estímulos percibidos en cada región, SpiNNaker determina cuál de ellas presenta la mayor cantidad de estímulos y, en consecuencia, guía a la plataforma robótica hacia esa dirección específica.

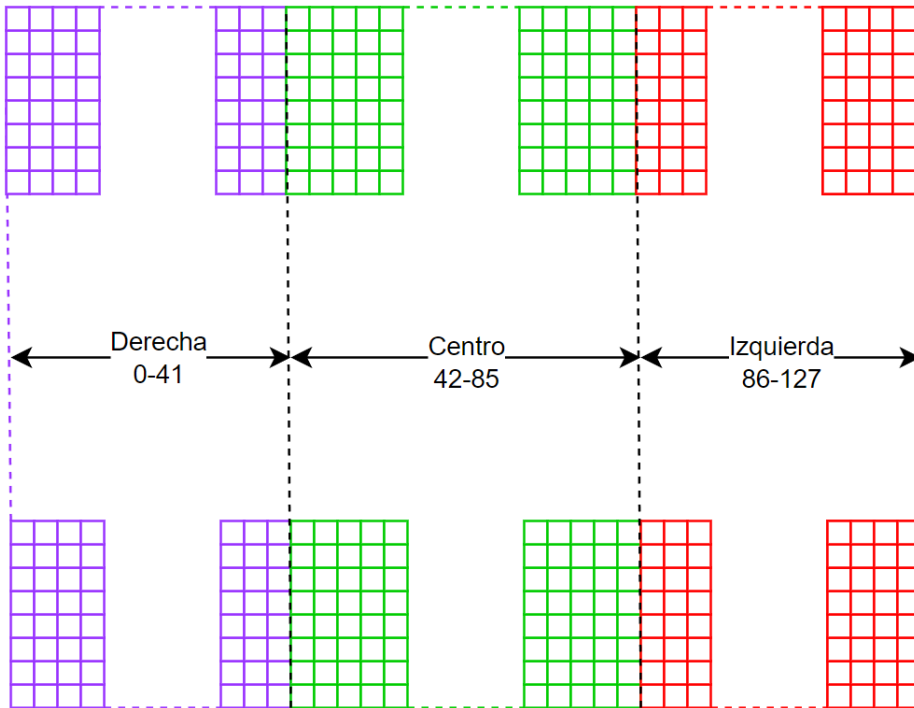


Figura 11.15: Distribución de los píxeles del DVS para la selección de la dirección de la plataforma robótica.

11.4.3.2 Interfaz DVS-SpiNNaker

Para comunicar el sensor DVS con SpiNNaker, se han diseñado e implementado distintos módulos en HDL. El esquema general de estos módulos se presenta en la Figura 11.16. El proceso de comunicación es el siguiente:

El módulo *AER Receiver*, que implementa el protocolo AER descrito en el capítulo 7, es responsable de recibir las direcciones asociadas a píxeles que detectan cambios de luminosidad. Este módulo selecciona solamente la coordenada X de dichas direcciones y la transfiere a través de un

11.4. Fase III

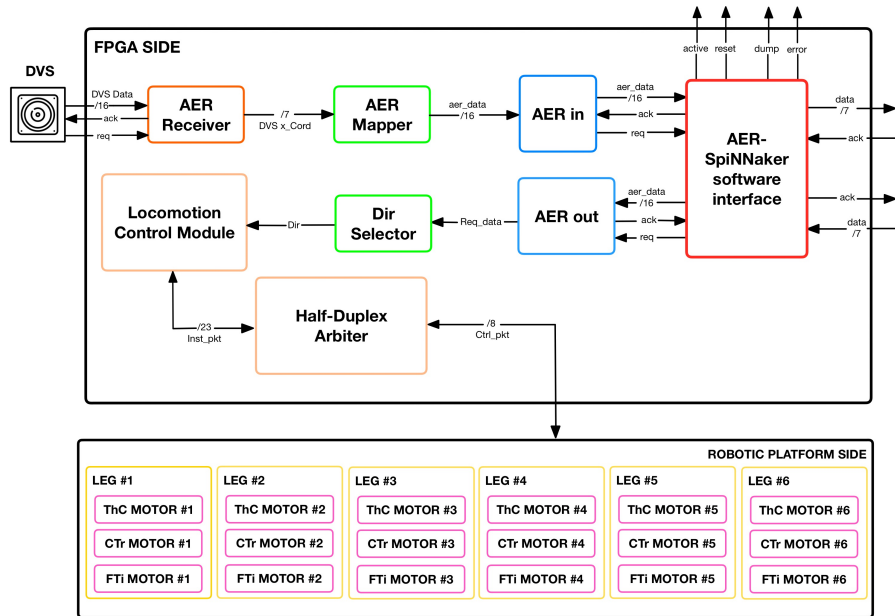


Figura 11.16: Esquema del módulo principal implementado en FPGA para la comunicación DVS-SpiNNaker.

bus síncrono al módulo *AER Mapper*. El módulo *AER Mapper*, por su parte, adapta estos datos al formato de 16 bits que requiere SpiNNaker y envía 20000 direcciones AER por segundo al módulo *AER in*. Este último módulo, que se comunica con SpiNNaker a través de SpiNNLinks, transfiere los datos para su procesamiento por la red WTA en SpiNNaker, tal como se explica en la subsección 11.4.2. Posteriormente, en la FPGA, el módulo *Dir Selector* clasifica las señales recibidas y selecciona la dirección adecuada en función de la salida de la red WTA. Finalmente, los módulos *Locomotion Control Module* y *Half-Duplex Arbiter* ejecutan funciones de control similares a las realizadas para el FSR.

11.4. Fase III

El diseño de la red involucra un total de 579 neuronas, distribuidas en 9 poblaciones diferentes. La tabla 11.6 detalla las conexiones sinápticas realizadas entre ellas.

Conexión sináptica	Valor (nA)	Probabilidad de conexión
$W_{IZQ}, W_{CEN}, W_{DER}, W_{FSR}$	2	1
W_{WTA}	1.5	1
W_{RefInh}	4	0.75
$W_{ReExc}, W_{IIE}, W_{EEe}$	2	0.25
$W_{IHi}, W_{EEi}, W_{AAi}, W_{BBi}$	1.5	0.75
W_{IE}	5	0.75
W_{IA}, W_{IB}, W_{BE}	0.5	0.75
W_{AI}, W_{EB}, W_{EA}	0.3	0.25
W_{AAe}, W_{BBe}	5	0.25
W_{AB}	0.25	0.75
W_{St-AB}	0.1	-

Tabla 11.6: Pesos sinápticos de la red neuronal adaptativa completa.

Por otro lado, en la Figura 11.18 se muestra un esquema general de la lógica implementada en la plataforma FPGA, tanto para el control de la velocidad con el FSR, como para el control de la dirección con el sensor DVS.

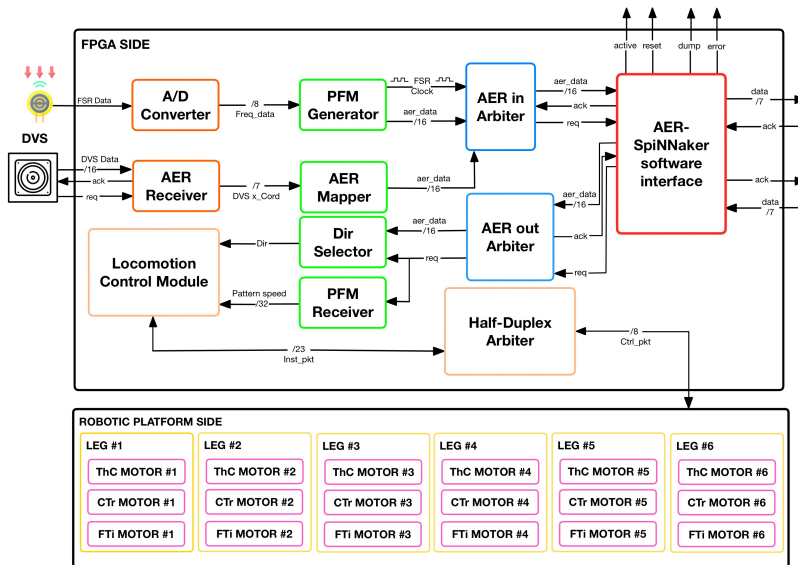


Figura 11.18: Esquema del módulo principal implementado en FPGA para la comunicación de los sensores FSR y DVS con SpiNNaker.

V

Resultados

12

Análisis de resultados

“Si he visto más lejos, es por estar parado sobre los hombros de gigantes.”

Isaac Newton

12.1 Introducción

En este capítulo, al igual que en el de métodos, se detallan de forma exhaustiva y cronológica los resultados obtenidos durante la tesis doctoral, estructurados en tres fases distintas, con la finalidad de mostrar progresivamente los avances realizados, hasta el modelo final de la red neuronal adaptativa obtenido en la fase III.

La 1ª fase se centra en los resultados obtenidos en las simulaciones realizadas en Brian2 con la red CPG, donde se somete cada área de la red de adaptación a distintas pruebas, con el fin de verificar la oscilación estable y sin cambios en el tiempo.

La 2ª fase se enfocará en analizar los resultados obtenidos al trasladar la red del simulador a la plataforma neuromórfica SpiNNaker. Se presentarán diversas pruebas ejecutadas de manera similar en ambas plata-

formas con el objetivo de verificar si existe similitud entre los resultados simulados y los implementados en el hardware. Además, se expondrán los resultados logrados mediante la optimización de la red y la implementación en tiempo real de los datos del FSR.

Finalmente, en la 3ª fase se muestran los resultados de la implementación del DVS y la red WTA, resultando en un salto significativo en la capacidad perceptiva de la plataforma robótica. Esta última fase detalla la efectividad de toda la red completa, subrayando cómo los desarrollos realizados contribuyen a la viabilidad y eficacia del modelo propuesto en esta tesis.

12.2 Fase I

Para comprobar si la arquitectura desarrollada se comporta de la forma esperada, en primer lugar se estudiará de forma aislada el funcionamiento del CPG_{AB} , asegurando que se produce una oscilación mantenida pero sin cambios en el tiempo.

Tras esto, examinaremos el funcionamiento del mismo CPG una vez se interconecte con el CPG_{12} , realizando pruebas con distintos estímulos para analizar los resultados obtenidos.

Finalmente, se realizará la conexión de la arquitectura completa, analizando los distintos escenarios en los que podría encontrarse la red en base a los datos de presión externos.

12.2.1 Análisis del CPG_{AB} en Brian 2

La Figura 12.1 ilustra la topología del CPG_{AB} . Como se señaló previamente, todas las neuronas en las poblaciones A y B reciben constantemente un estímulo, denominado I_{St} , con un valor fijo de $2,2nA$.

Este valor específico fue determinado a través de experimentos que demostraron ser el mínimo necesario para lograr una oscilación estable en

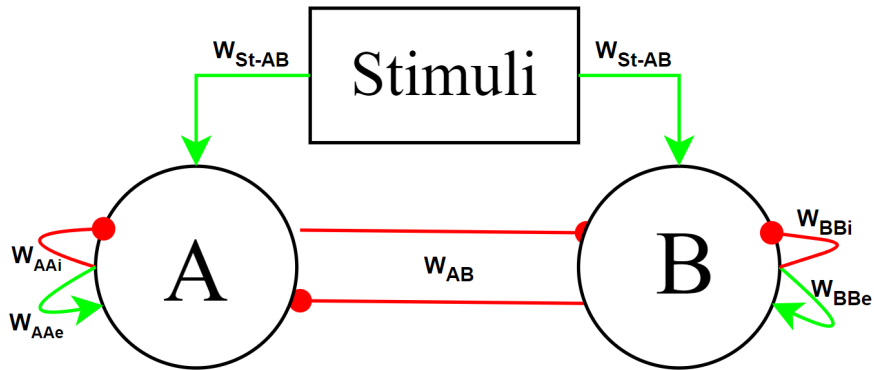


Figura 12.1: Concepto de la topología del CPG_{AB} . Se presentan las dos poblaciones sometidas al primer estudio (A y B), junto con la corriente de estímulo.

el CPG. Aunque es posible utilizar valores mayores para I_{St} , estos generan, como era de esperar, una frecuencia de oscilación más alta y un incremento significativo en el ruido de la simulación, como se muestra en las figuras 12.2 y 12.3. Por consiguiente, se estableció el valor de $2,2nA$ para facilitar la observación del impacto de la retroalimentación en el funcionamiento del CPG.

Aunque las conexiones neuronales se han formado basándose en una cierta probabilidad¹, es determinante fijar con precisión los valores de los pesos sinápticos para todas estas conexiones, con el fin de garantizar la oscilación adecuada del CPG. Específicamente, se ha definido un peso de $0,5nA$ para las conexiones inhibitorias entre las distintas poblaciones neuronales. En cuanto a las conexiones dentro de cada población (intrapoblacionales), los pesos se han fijado en $4nA$ para las conexiones excitatorias y en $1,5nA$ para las inhibitorias, teniendo en cuenta el número de neuronas en cada población.

Con esta breve prueba se demuestra que, mientras la corriente de estímulo I_{St} se mantenga constante en el tiempo, la frecuencia de oscilación del CPG también mantendrá su frecuencia de oscilación constante.

¹25 % de probabilidad de conexión para las excitatorias y un 75 % para las inhibitorias.

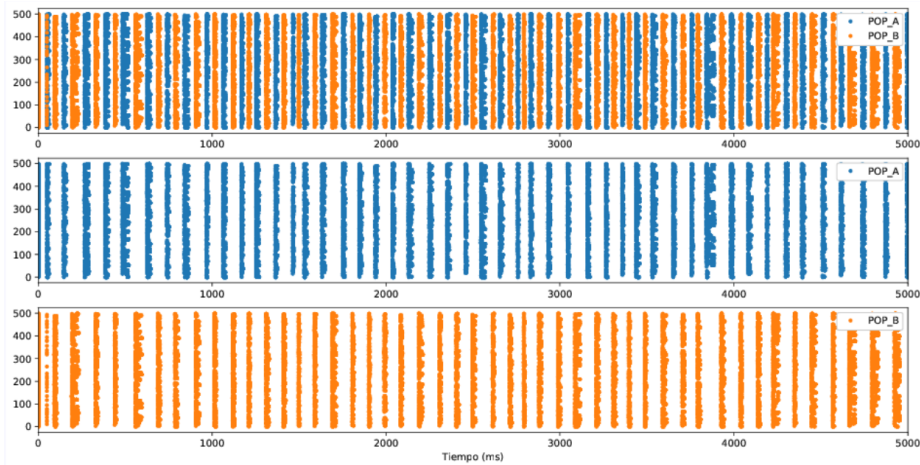


Figura 12.2: Simulación del CPG_{AB} con un valor de I_{St} de $10nA$. Puede observarse el aumento en la frecuencia de oscilación junto con un aumento en la cantidad de ruido.

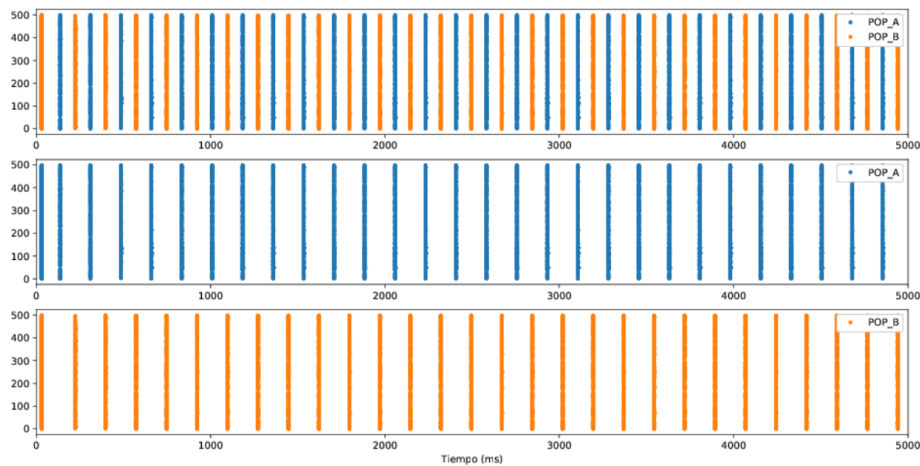


Figura 12.3: Simulación del CPG_{AB} con un valor de I_{St} de $2,2nA$. Puede observarse una menor frecuencia de oscilación y casi la eliminación total del ruido.

12.2.2 Análisis de la red de realimentación en Brian 2

El siguiente paso para estudiar el funcionamiento de la arquitectura neuronal sería añadir conexiones entre el CPG principal (CPG_{AB}) y el secundario (CPG_{12}). En este caso, tal y como se puede apreciar en la Figura 12.4, se han realizado únicamente las conexiones que van desde

12.2. Fase I

el CPG_{12} al CPG_{AB} , con la intención de poder observar con claridad el efecto que tiene la realimentación sobre el CPG.

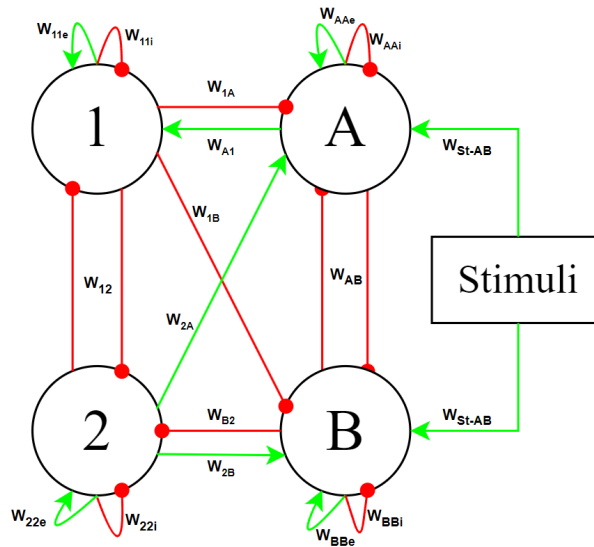


Figura 12.4: Conexionado del CPG_{AB} y el CPG_{12} para simulaciones aisladas.

En este escenario, se han implementado conexiones inhibitorias desde la P_1 hacia el CPG_{AB} , mientras que desde P_2 se han establecido conexiones exclusivamente excitatorias hacia el mismo CPG. Tanto las conexiones excitatorias como las inhibitorias presentan pesos sinápticos de $0,1nA$. Por otro lado, el CPG_{12} comparte las mismas características en cuanto a conexiones intrapoblacionales e interpopulacionales, probabilidades de conexión y valores de pesos sinápticos con el CPG_{AB} . El propósito de estas conexiones es que, dependiendo de la población dominante en el CPG_{12} , se produzca una modificación en la frecuencia de oscilación del CPG_{AB} . En específico, si la población P_1 es la dominante, se espera una disminución en la frecuencia de oscilación del CPG, mientras que si P_2 predomina, se anticipa un aumento en dicha frecuencia.

12.2.2.1 Comportamiento de la población 1 en Brian 2

Para lograr la dominancia de la población P_1 en el contexto donde no hay estímulos externos que afecten al CPG_{12} , se ha introducido una pequeña corriente constante positiva en P_1 . Debido a esta intervención, tal y como se muestra en la Figura 12.5, se produce una ligera disminución en la frecuencia de oscilación del CPG_{AB} , la cual desciende hasta $10,4Hz$, representando una reducción del 10,34 % en la frecuencia de oscilación original.

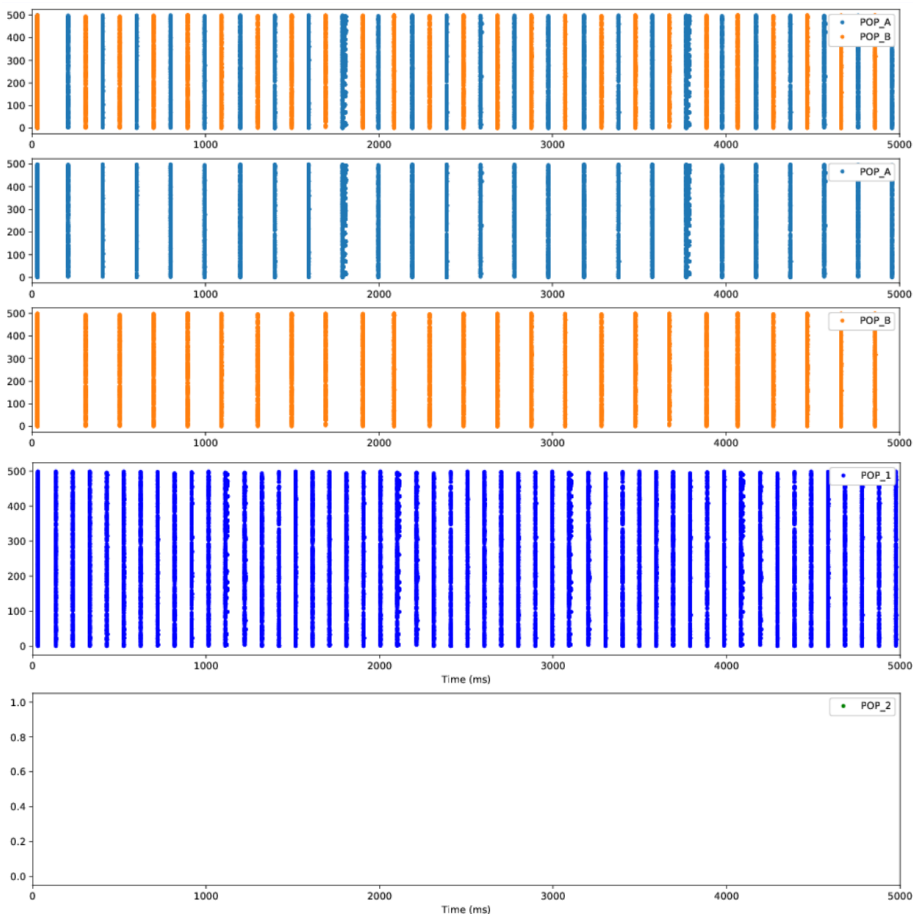


Figura 12.5: Simulación del CPG_{12} en la que se le ha inyectado corriente constante a P_1 .

12.2. Fase I

12.2.2.2 Comportamiento de la población 2 en Brian 2

Se ha realizado el mismo experimento que en la subsección anterior, pero inyectando la misma corriente constante positiva a P_2 (Figura 12.6). En este caso, P_2 domina totalmente sobre P_1 , por lo que el CPG_{AB} recibe más estímulos, aumentando su frecuencia hasta los $14Hz$, un incremento del 20,69 %.

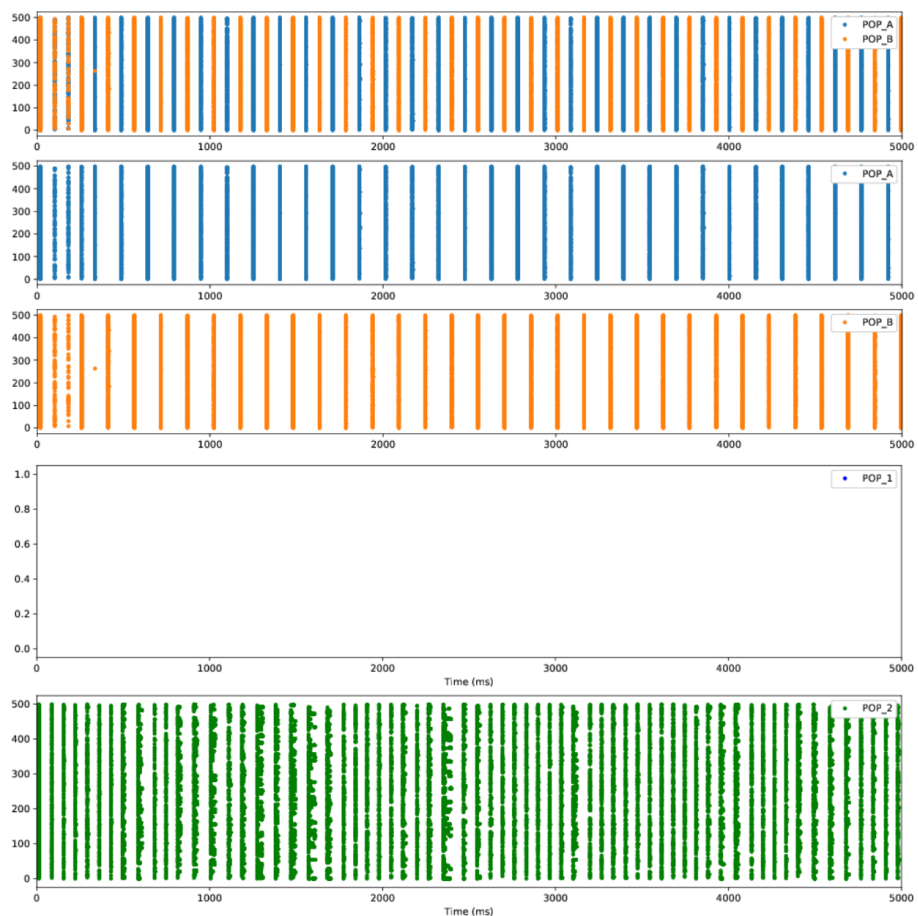


Figura 12.6: Simulación del CPG_{12} en la que se le ha inyectado corriente constante a P_2 .

12.2.3 Análisis del efecto del sensor sobre la arquitectura neuronal en Brian 2

Para completar la red de realimentación se ha utilizado la población P_{Ref} , cuyo patrón de eventos de salida sigue una distribución de *Poisson*. Esta población emitirá pulsos inhibitorios y excitatorios a las poblaciones P_1 y P_2 , respectivamente. El esquema de la red se presenta nuevamente en la Figura 12.7.

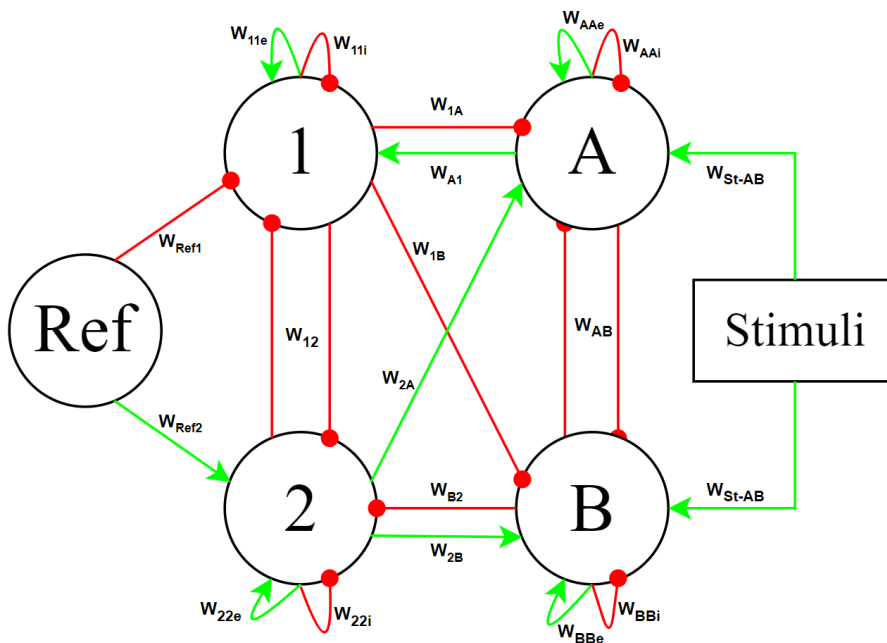


Figura 12.7: Simulación del CPG_{12} en la que se le ha inyectado corriente constante a P_2 .

Con la red de adaptación ya configurada, se llevaron a cabo varios experimentos en los que la red fue expuesta a diversas entradas simuladas del FSR. Estos experimentos tenían como objetivo demostrar la capacidad de adaptación de la oscilación del CPG en respuesta a las variaciones en las señales del FSR.

12.2. Fase I

12.2.3.1 Entrada única de 5V en Brian 2

Inicialmente, se simuló una entrada de $5V$ en el instante $t = 1000ms$, con una duración de $1000ms$. Durante este periodo, las neuronas de P_{Ref} dispararon a una frecuencia aproximada de $171Hz$. Una vez finalizado este intervalo de disparo, no se realizaron más entradas del sensor.

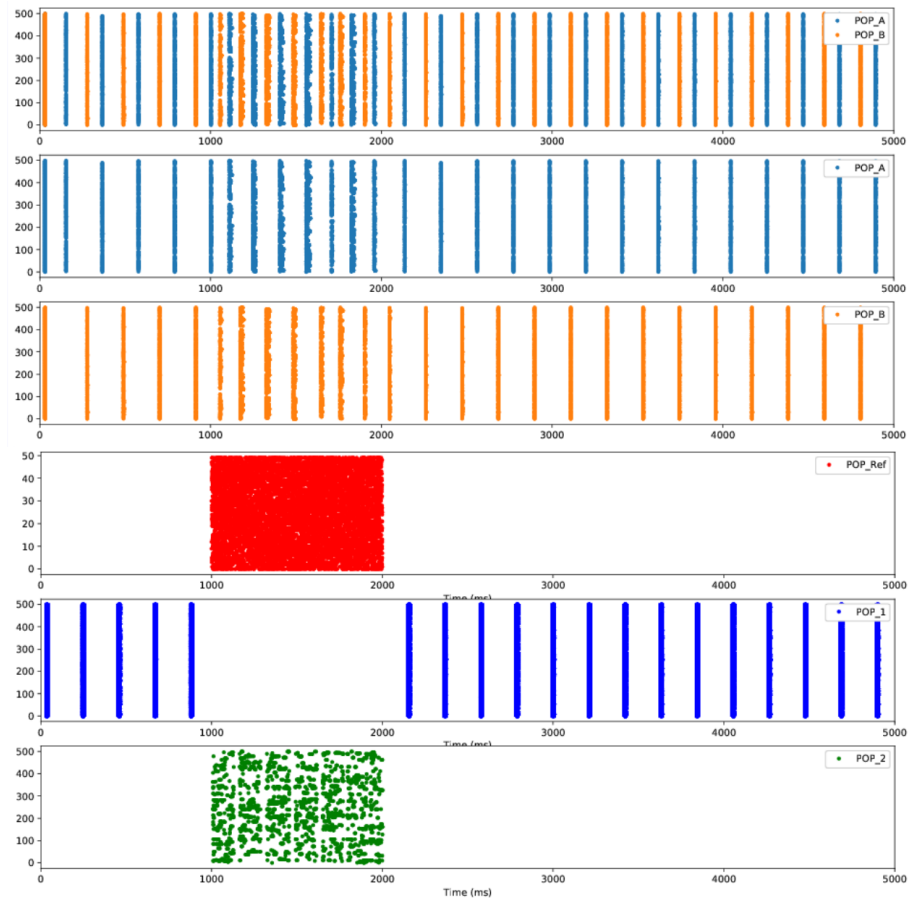


Figura 12.8: Resultado de la simulación ante un estímulo con distribución *Poisson* de $171Hz$ durante $1000ms$. En la primera gráfica puede observarse la variación de frecuencia del CPG_{AB} durante los $1000ms$ que dura el estímulo de P_{Ref} .

En la Figura 12.8 se presenta el resultado de esta simulación en las poblaciones del CPG_{12} . Al analizar la gráfica, se aprecia que en $t = 0ms$, en ausencia de información del sensor, P_{Ref} no presenta actividad. Esto

resulta en que P_1 sea excitada por el CPG_{AB} y P_2 inhibida y, a su vez, P_1 comienza a inhibir levemente al CPG_{AB} .

En el instante $t = 1000ms$, P_{Ref} empieza a disparar a una frecuencia de aproximadamente $171Hz$ por neurona, lo que induce una excitación instantánea en P_2 y la inhibición de P_1 . Durante este tiempo, P_2 excita al CPG_{AB} , lo que incrementa su frecuencia de oscilación.

Una vez finalizada la actividad de P_{Ref} en $t = 2000ms$, P_1 recupera su dominio sobre P_2 , y comienza nuevamente a inhibir levemente al CPG_{AB} . En la misma Figura 12.8, se puede ver el resultado de esta simulación en el CPG_{AB} , donde la frecuencia aumenta significativamente entre $t = 1000ms$ y $t = 2000ms$, alcanzando frecuencias mínimas de $8Hz$ y máximas de $15Hz$.

12.2.3.2 Entradas alternas de 0-5V en Brian 2

Se ha llevado a cabo otra simulación que consiste en alternar los picos máximos y mínimos de voltaje en el sensor. El objetivo de esta simulación es evaluar la capacidad de la red para adaptarse a cambios repentinos en los valores del FSR. Como se muestra en la Figura 12.9, las poblaciones P_1 y P_2 cambian su actividad en respuesta a la actividad de P_{Ref} , de manera similar a lo observado en la simulación anterior. En esta ocasión, el CPG_{AB} muestra variaciones en su frecuencia de oscilación, oscilando entre $8Hz$ y $15Hz$, pero con la particularidad de que estos cambios ocurren de forma alternada en el tiempo.

12.2.3.3 Entradas aleatorias en Brian 2

Finalmente, se realizaron diversas simulaciones que replicaron escenarios reales, en los cuales el sensor recopilaría datos del entorno. Estos datos variaron de forma más gradual en comparación con los escenarios previamente analizados.

En la primera simulación (ver Figura 12.10), se introdujeron diez valores aleatorios de voltaje para imitar las posibles mediciones del FSR. Estos

12.2. Fase I

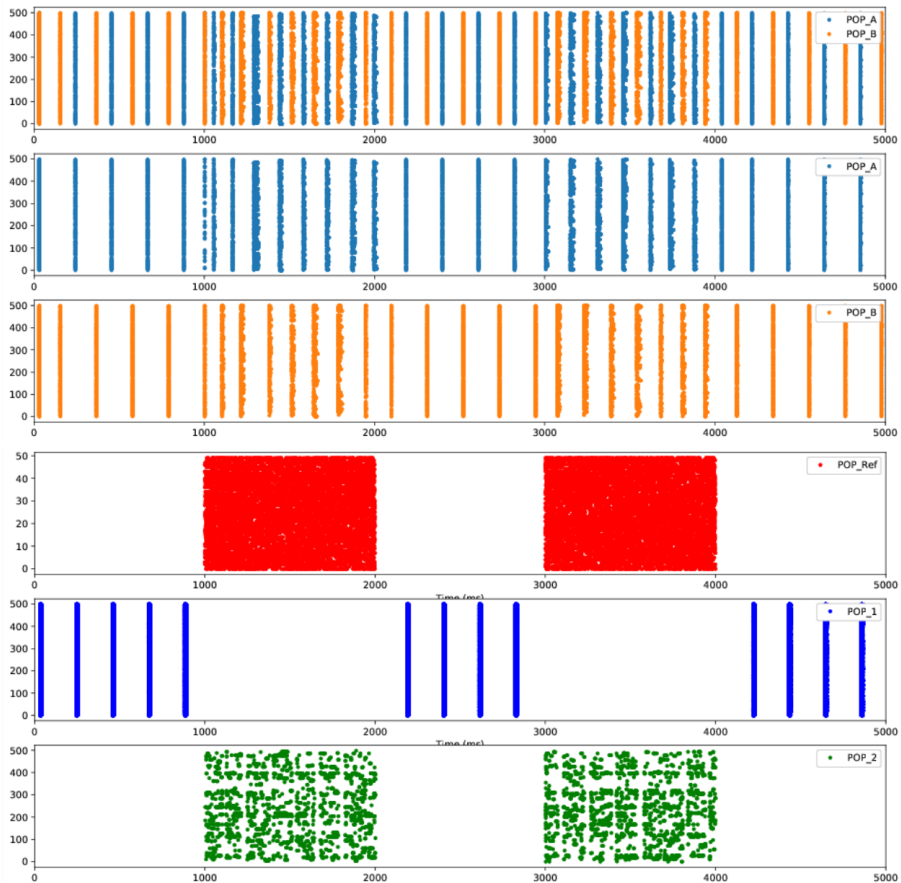


Figura 12.9: Resultado de la simulación ante un estímulo con distribución *Poisson* de 171Hz durante 1000ms . En la primera gráfica puede observarse la variación de frecuencia del CPG_{AB} durante los 1000ms que dura el estímulo de P_{Ref} .

valores se renovaron cada 500ms , lo que obligaba a la red a adaptarse continuamente. Los valores de frecuencia empleados para la distribución de *Poisson* fueron 171Hz , 40Hz , 80Hz , 30Hz , 5Hz , 130Hz , 50Hz , 76Hz , 20Hz y 150Hz

La segunda simulación (ilustrada en la Figura 12.11) consistió en someter al CPG_{AB} a un incremento constante en la frecuencia de oscilación. Se introdujeron aumentos de 20Hz cada 500ms en la frecuencia de la población P_{Ref} . Para generar un escenario de mayor estrés, los últimos valores de frecuencia inyectados excedieron en hasta un 17 % el máximo

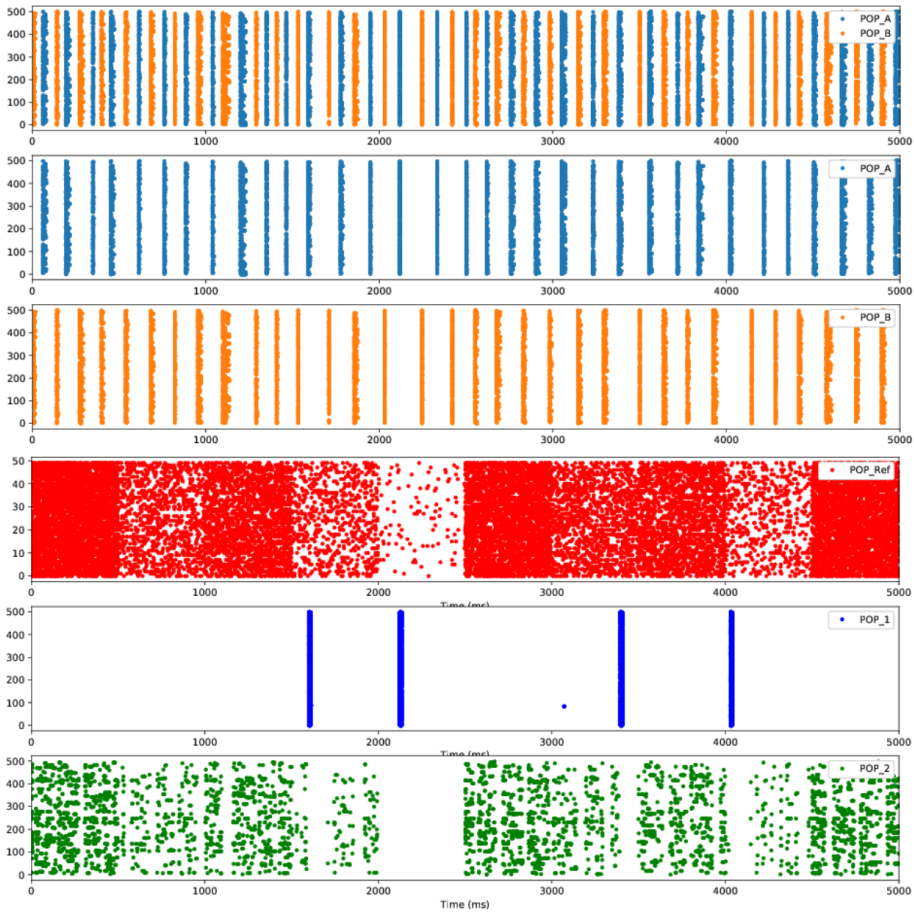


Figura 12.10: Variación de la salida del CPG_{AB} en función de P_{Ref} . En este caso se simula un terreno irregular en el que se van obteniendo valores de tensión aleatorios.

teórico de $171Hz$.

Estos experimentos, aunque se limitan a ser simulaciones teóricas, son fundamentales para demostrar la viabilidad y eficiencia de la red neuronal en la regulación de la locomoción de la plataforma robótica. Cada prueba, con sus respectivas complejidades y desafíos, ha servido para evaluar la adaptabilidad y respuesta de la red ante diferentes escenarios. Lo notable es que, en todas estas pruebas, se ha logrado consistentemente alcanzar la oscilación característica de los CPGs. Esto no solo subraya la robustez y flexibilidad de la red, sino que también abre nue-

12.2. Fase I

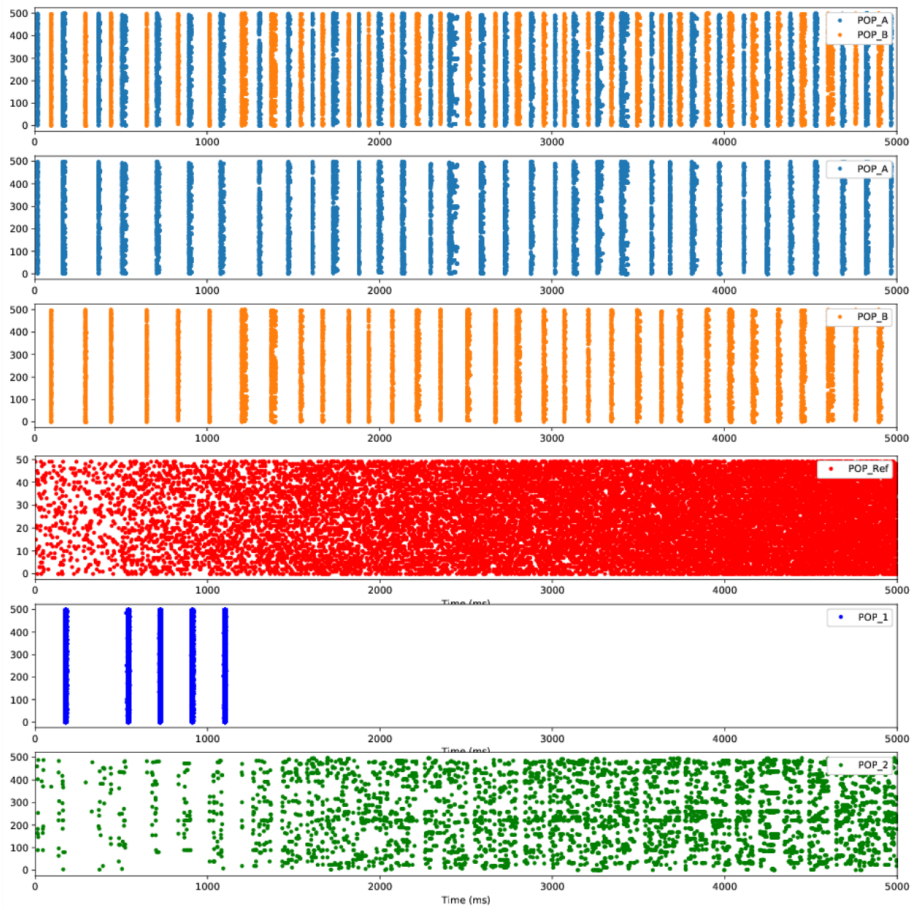


Figura 12.11: Variación de la salida del CPG_{AB} en función de P_{Ref} . Se simula un incremento constante en la fuerza establecida por el robot sobre el terreno, lo que provoca un aumento en la frecuencia del CPG_{AB} simultáneamente.

vas posibilidades para su aplicación en sistemas robóticos más complejos y dinámicos, donde la precisión en la locomoción y la capacidad de adaptación a condiciones cambiantes son esenciales.

12.3 Fase II

Después de optimizar la cantidad de neuronas en cada población y sus parámetros, como se detalla en la sección 11.3, se repitieron las pruebas realizadas anteriormente en la Fase I. Sin embargo, en esta ocasión, las pruebas se llevaron a cabo tanto en el simulador Brian 2 como en la plataforma neuromórfica SpiNNaker. El objetivo de esta repetición era verificar si los resultados obtenidos en ambos entornos, el simulador y la plataforma, presentaban similitudes significativas.

12.3.1 Implementación de los nuevos parámetros neuronales en Brian 2

Las figuras 12.12 y 12.13 presentan brevemente los resultados de las simulaciones realizadas en Brian 2, utilizando los parámetros neuronales ajustados.

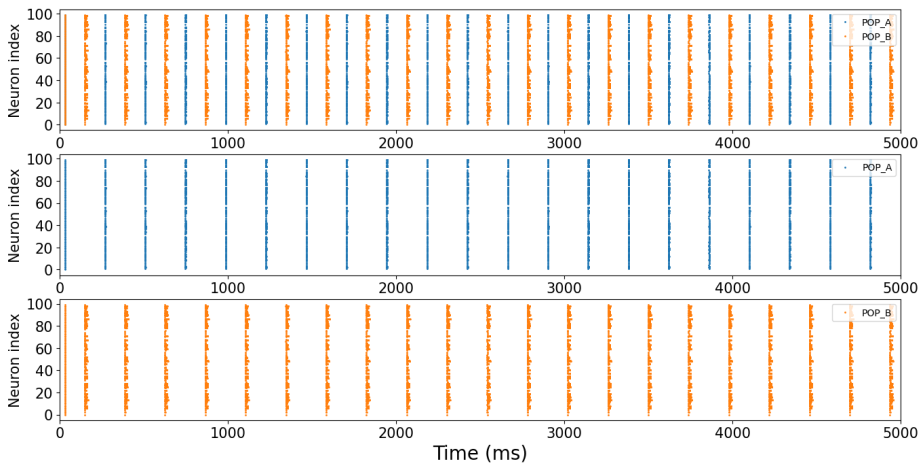


Figura 12.12: Simulación en Brian 2 con los nuevos parámetros neuronales, así como con la reducción del número de neuronas.

Como puede apreciarse, los resultados en estas simulaciones son similares a los obtenidos anteriormente, consiguiendo que la frecuencia del CPG aumente o disminuya en función de la entrada simulada del sensor FSR, pero con una reducción del 80 % de neuronas por población.

12.3. Fase II

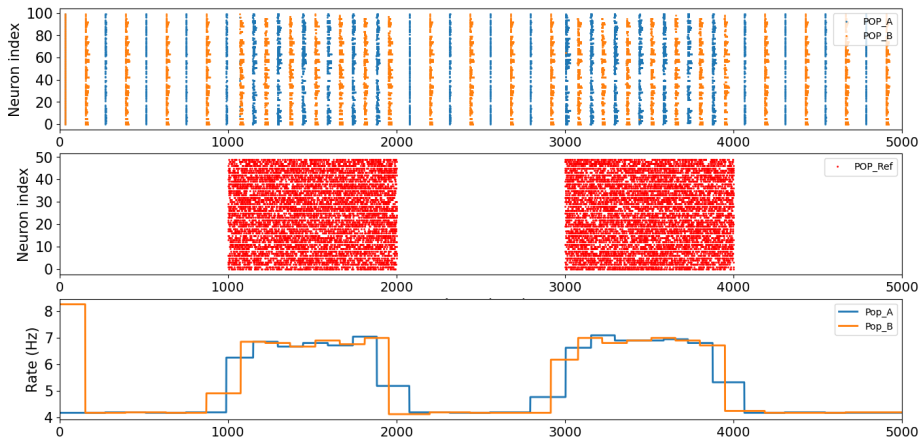


Figura 12.13: Simulación de entradas máximas en Brian 2 con los nuevos parámetros neuronales, así como con la reducción del número de neuronas.

12.3.2 Implementación de la red adaptativa en SpiNNaker

Se replicaron los experimentos de la Fase I en la plataforma SpiNNaker, con el objetivo de verificar si los resultados son consistentes con los obtenidos en Brian 2. Para demostrar el eficaz desempeño del modelo en esta plataforma neuromórfica, se llevaron a cabo las mismas pruebas que en Brian 2.

Inicialmente, se comprobó la capacidad del CPG para generar oscilaciones sostenidas sin estímulos externos. Posteriormente, el CPG fue expuesto a las entradas extremas del sensor. Por último, se evaluó su respuesta frente a entradas aleatorias y progresivamente crecientes.

12.3.2.1 Análisis del CPG_{AB} en SpiNNaker

Para la implementación del CPG_{AB} en SpiNNaker también se utilizaron 100 neuronas por población, con una corriente de estímulo I_{St} constante de $2,2nA$. La frecuencia de oscilación del CPG fue de $11,62Hz$ (Figura 12.14), lo que coincide con la frecuencia de oscilación medida en la simulación de Brian 2. Aunque se perdieron algunos spikes en el trazado en comparación con Brian 2, estos pueden atribuirse al soporte limitado de cálculos en punto flotante en SpiNNaker. Además, el patrón parece

consistente y con poco ruido.

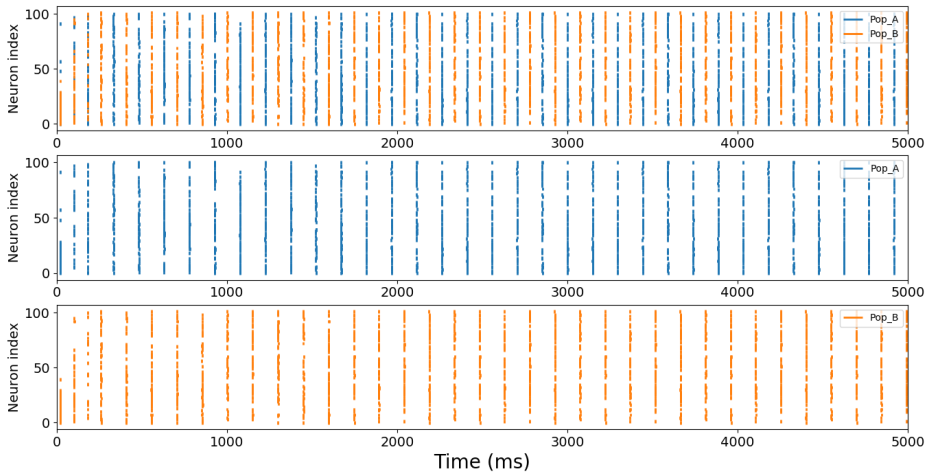


Figura 12.14: Simulación en SpiNNaker del CPG_{AB} con una frecuencia de oscilación constante.

12.3.2.2 Entradas alternas de 0-5V en SpiNNaker

De igual forma que con el simulador, se ha simulado un cambio repentino del valor del sensor, representado por la frecuencia del generador *Poisson* en P_{Ref} (Figura 12.15). Esta frecuencia se estableció en $0,1Hz$ durante el primer segundo de la simulación y osciló entre ese valor y $171Hz$ durante los siguientes cuatro segundos, para apreciar ambos regímenes del CPG. Se puede observar como, con un baja frecuencia de P_{Ref} , P_1 es la dominante en el CPG_{12} , provocando una frecuencia de oscilación media de $12,5Hz$. Con una frecuencia de $171Hz$ en P_{Ref} , P_2 será la dominante y la frecuencia de oscilación media del CPG_{AB} asciende hasta los $27,5Hz$.

12.3.2.3 Entradas aleatorias en SpiNNaker

Finalmente, las figuras 12.16 y 12.17 muestran la respuesta del CPG en SpiNNaker a frecuencias crecientes y a valores aleatorios de P_{Ref} , respectivamente. En ambos casos se observa como el patrón de oscilación del CPG se adapta perfectamente a los cambios producidos en la

12.3. Fase II

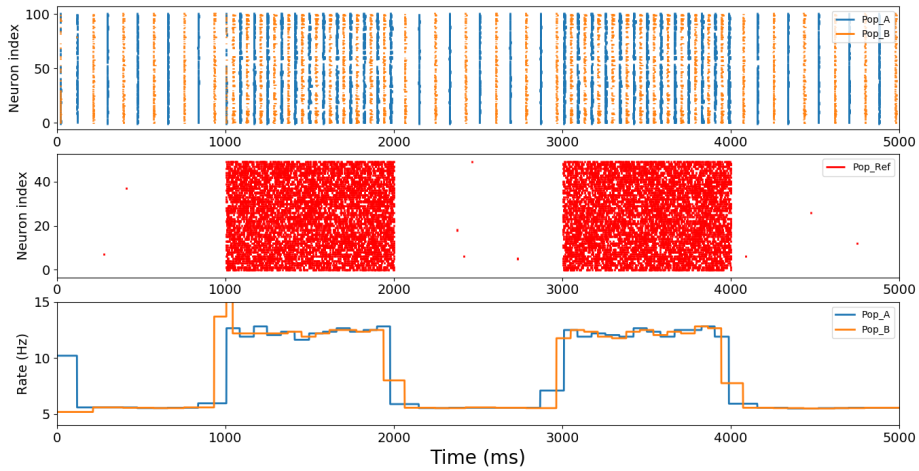


Figura 12.15: Resultado en SpiNNaker del CPG sometido a valores extremos de estímulo de entrada ($0,1Hz$ durante la mayor parte del tiempo, y $171Hz$ entre $t = 2000ms$ y $t = 3000ms$, y entre $t = 4000ms$ y $t = 5000ms$). La tercera gráfica muestra la frecuencia de oscilación de cada población del CPG_{AB} en Hz . Por lo tanto, la frecuencia de oscilación global del CPG será la combinación de ambas.

frecuencia de P_{Ref} .

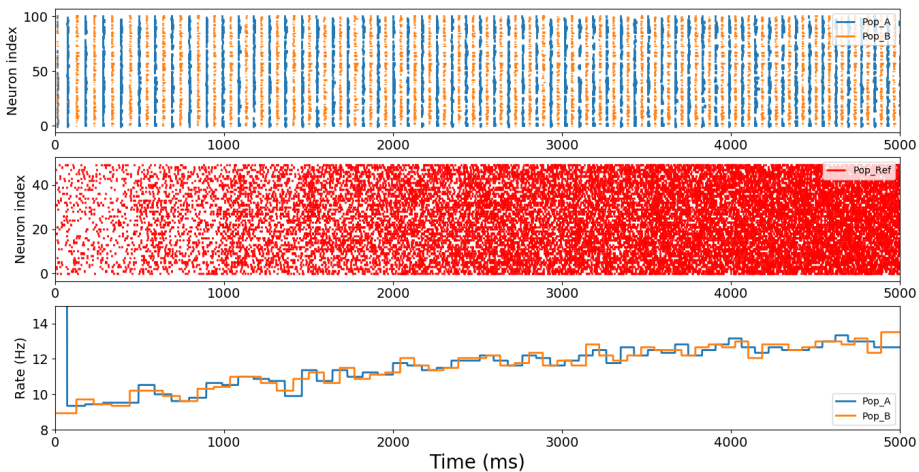


Figura 12.16: Resultado en SpiNNaker del CPG sometido a valores crecientes de la frecuencia de P_{Ref} , con incrementos de $20Hz$ cada $500ms$. La tercera gráfica muestra la frecuencia de oscilación de cada población del CPG en Hz . Por lo tanto, la frecuencia de oscilación global del CPG será la combinación de ambas.

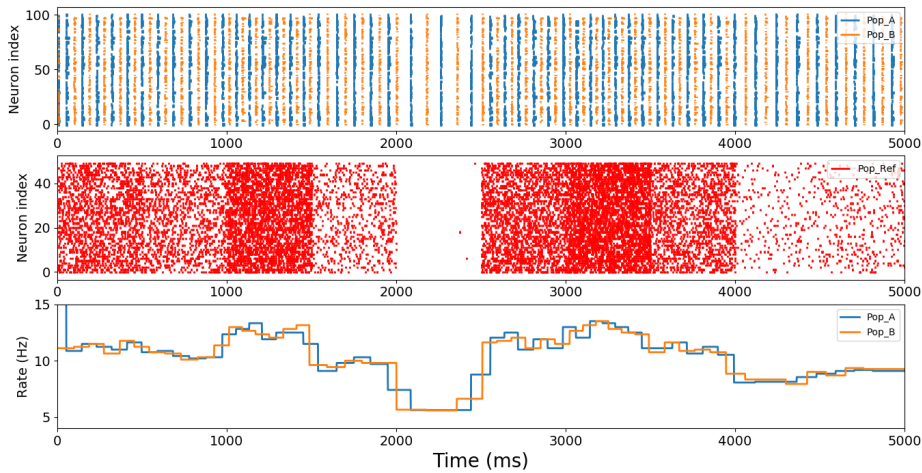


Figura 12.17: Resultado en SpiNNaker del CPG sometido a diez valores aleatorios de la frecuencia de entrada P_{Ref} . La tercera gráfica muestra la frecuencia de oscilación de cada población del CPG en Hz . Por lo tanto, la frecuencia de oscilación global del CPG será la combinación de ambas.

12.3.3 Comparación de los resultados obtenido en Brian 2 y SpiNNaker

La Figura 12.18 muestra la comparación de los resultados obtenidos entre ambas plataformas: las simulaciones de Brian 2 y la plataforma SpiNNaker. En todos los experimentos se utilizaron los mismos estímulos de entrada por parte de P_{Ref} para ambos casos, los cuales variaron entre $0Hz$ y $171Hz$. El rango de frecuencia generada por ambos CPGs fueron muy similares: en concreto, en el caso de Brian 2 la frecuencia de oscilación del CPG obtuvo unos valores mínimos y máximos de $9,5Hz$ y $14,9Hz$, mientras que en SpiNNaker fueron de $11,62Hz$ y $27,5Hz$, respectivamente.

12.3.4 Evaluación de los modelos de sCPG con datos reales del FSR

Los valores obtenidos del FSR son fundamentales para determinar el tipo de superficie sobre la cual la plataforma robótica camina. Estos valores actúan como retroalimentación, permitiendo al robot identificar si

12.3. Fase II

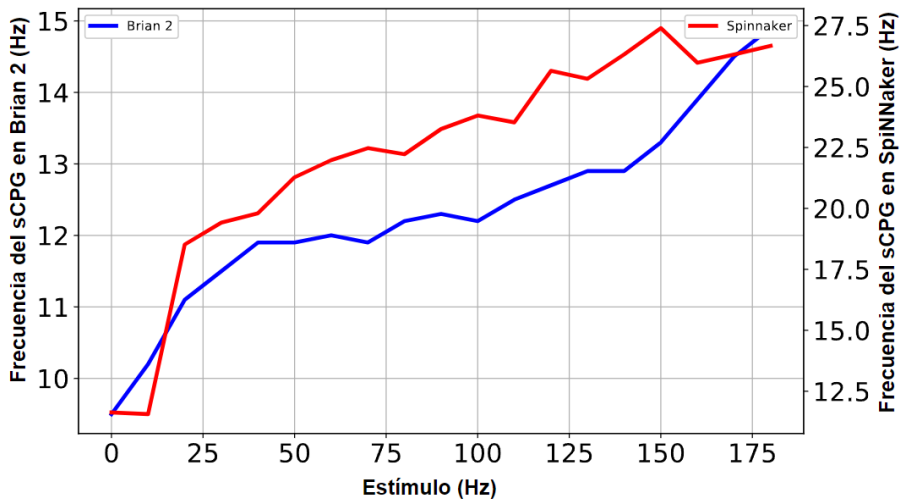


Figura 12.18: Comparación de los resultados obtenidos tanto en la simulación de Brian 2 como en la implementación de SpiNNaker. El gráfico muestra la tasa generada por el sCPG cuando el estímulo de entrada (P_{Ref}) cambia de $0Hz$ a $180Hz$. La traza azul muestra los resultados de Brian 2 (eje Y izquierdo) y la traza roja los resultados de SpiNNaker (eje Y derecho).

está pisando una superficie blanda o rígida, tal como se ilustra en la Figura 12.19.

Existe una correlación directa entre la frecuencia del sCPG y la salida del FSR, lo que influye en la velocidad del robot. Específicamente, si el robot detecta una superficie blanda, la velocidad de balanceo de la pata se reduce, mientras que en superficies rígidas, esta velocidad aumenta. Sin embargo, la amplitud del movimiento de la pierna se mantiene constante en ambos casos.

Para validar estas hipótesis, se realizaron experimentos usando estímulos reales a partir de un sensor FSR integrado en una de las patas del robot. Los ensayos se llevaron a cabo en dos tipos de terrenos: madera y arena, con el fin de contrastar un suelo firme frente a uno más irregular. La configuración experimental se detalla en la Figura 12.20. Se ejecutaron cinco pruebas en cada superficie, y los resultados promedio están presentados en la Figura 12.19. En los experimentos, las mediciones de voltaje medio cuando la pata del robot estaba completamente en contac-

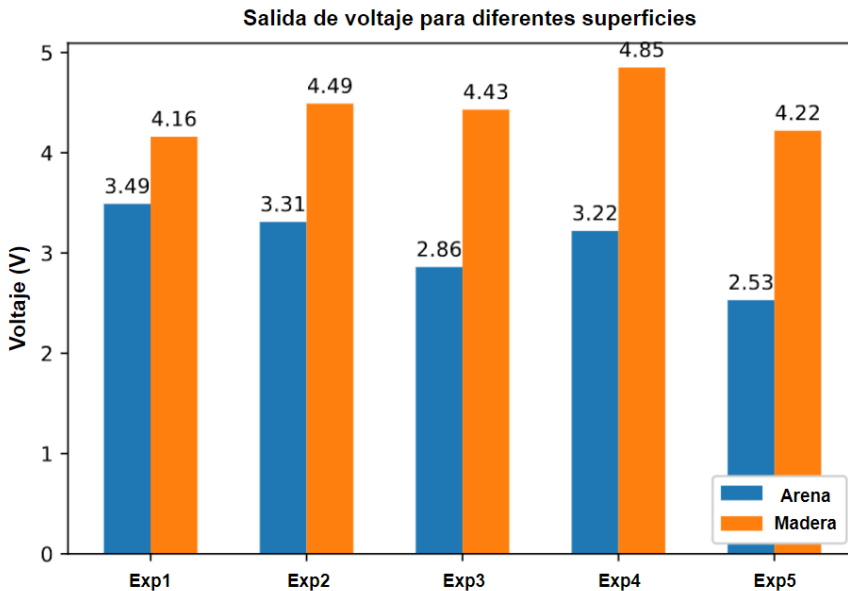


Figura 12.19: Voltaje medio de salida del sensor FSR para dos tipos de terrenos. En cinco experimentos diferentes, se midió el voltaje mientras una pata se movía sobre dos superficies distintas: arena y madera. Las barras azules representan los resultados obtenidos en la superficie de arena, y las barras naranjas, los de la superficie de madera. En la parte superior de cada barra se indica el valor medio del voltaje. La desviación estándar fue de 0,34 en la superficie de arena y de 0,24 en la de madera. Los resultados muestran que el voltaje medio es más bajo en la superficie de arena en comparación con la más rígida, madera.

to con la superficie mostraron que, en la madera, los valores superaban los 4 voltios, mientras que en la arena, estos valores eran inferiores a 3.5 voltios.

La relación entre el voltaje medido por el sensor FSR y la frecuencia de disparo del conjunto de neuronas en P_{Ref} permite modular la frecuencia de oscilación del CPG_{AB} . Este mecanismo tiene un impacto directo en la velocidad de marcha de la plataforma robótica, permitiendo ajustes significativos en función de las lecturas del FSR. Para validar y entender mejor este comportamiento, se han llevado a cabo experimentos tanto en simulaciones como en implementaciones prácticas.

Los resultados de las simulaciones en Brian 2 se presentan en las figuras

12.3. Fase II



Figura 12.20: Izquierda: Captura de la configuración utilizada para realizar los experimentos con el FSR. En [207] se presenta un vídeo de los experimentos realizados, donde la pata se movía manualmente. Derecha: fotografía de una de las patas para mostrar dónde están ubicados los motores y el sensor FSR.

12.21 y 12.22. Estas figuras ilustran cómo las variaciones en la señal del FSR afectan a la frecuencia de oscilación del CPG_{AB} y, por ende, a la velocidad de marcha del robot en un entorno controlado. Por otro lado, las figuras 12.23 y 12.24 muestran los resultados de las implementaciones en SpiNNaker, destacando cómo el sistema se comporta en un entorno más realista y práctico, confirmando las hipótesis y observaciones hechas en las simulaciones de Brian 2.

Por otro lado, como se muestra en la Figura 12.25, la diferencia de frecuencia generada por P_{Ref} en arena y madera es de un 28,4 %, mientras que los resultados muestran que la oscilación del CPG_{AB} es un 17 % menor cuando la pata está en terreno arenoso que cuando está en madera. La diferencia de frecuencia muestra que es posible adaptar la frecuencia del sCPG a las condiciones externas dadas por la salida del FSR.

Otro experimento realizado consistía en introducir una transición entre terrenos, con el fin de demostrar la variabilidad de la frecuencia del sCPG. Para este propósito, solo se tomaron los valores máximos obtenidos de P_{Ref} , los cuales corresponden a los valores obtenidos cuando la pata se encontraba totalmente apoyada en el terreno. Igual que los experimentos anteriores, este se ha realizado tanto en Brian 2 como en SpiNNaker (figuras 12.26 y 12.27, respectivamente).

12. Análisis de resultados

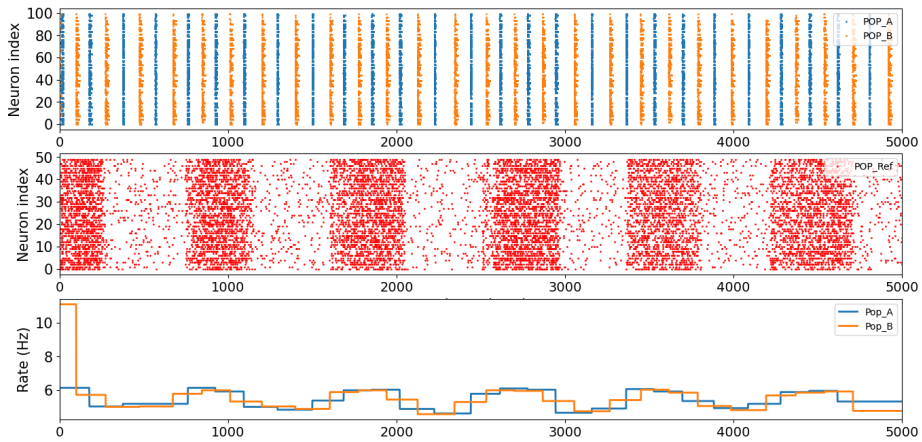


Figura 12.21: Simulación en Brian 2 del CPG para valores del FSR alojado en una pata al caminar sobre un terreno arenoso. La primera gráfica muestra los spikes de salida del CPG_{AB} , la gráfica intermedia muestra los spikes generados por P_{Ref} y la última muestra la frecuencia de disparo de cada población del CPG en Hz . Por lo tanto, la tasa de salida global del CPG será la combinación de ambas.

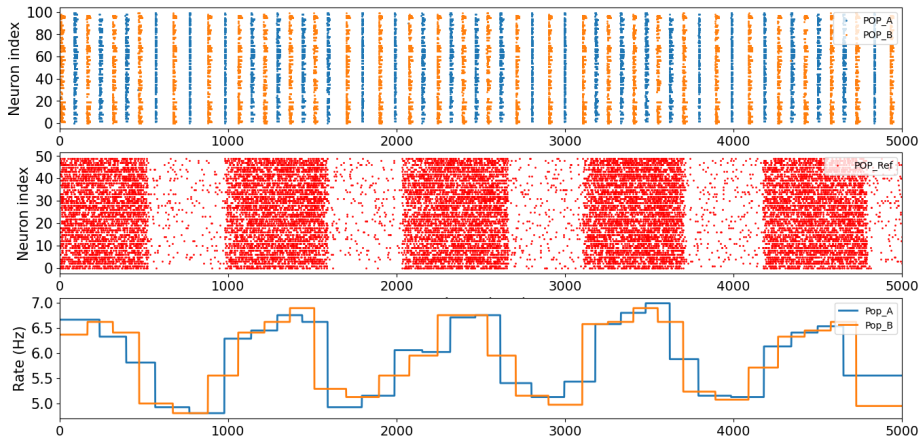


Figura 12.22: Simulación en Brian 2 del CPG para valores del FSR alojado en una pata al caminar sobre un terreno de madera. La primera gráfica muestra los spikes de salida del CPG_{AB} , la gráfica intermedia muestra los spikes generados por P_{Ref} y la última muestra la frecuencia de disparo de cada población del CPG en Hz . Por lo tanto, la tasa de salida global del CPG será la combinación de ambas.

12.4. Fase III

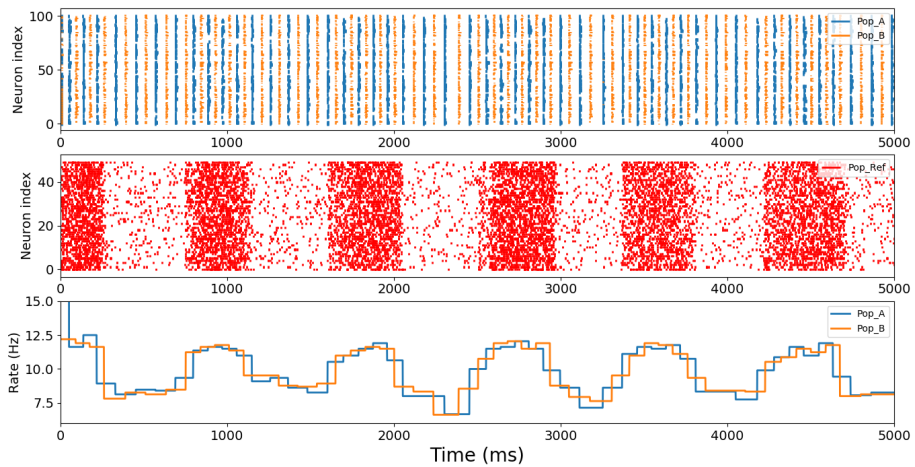


Figura 12.23: CPG implementado en SpiNNaker cuando la pata toca un terreno arenoso. La primera gráfica muestra los spikes de salida del CPG_{AB} , la gráfica intermedia muestra los picos generados por P_{Ref} y la última muestra la frecuencia de disparo de cada población del CPG en Hz . Por lo tanto, la tasa de salida global del CPG será la combinación de ambas.

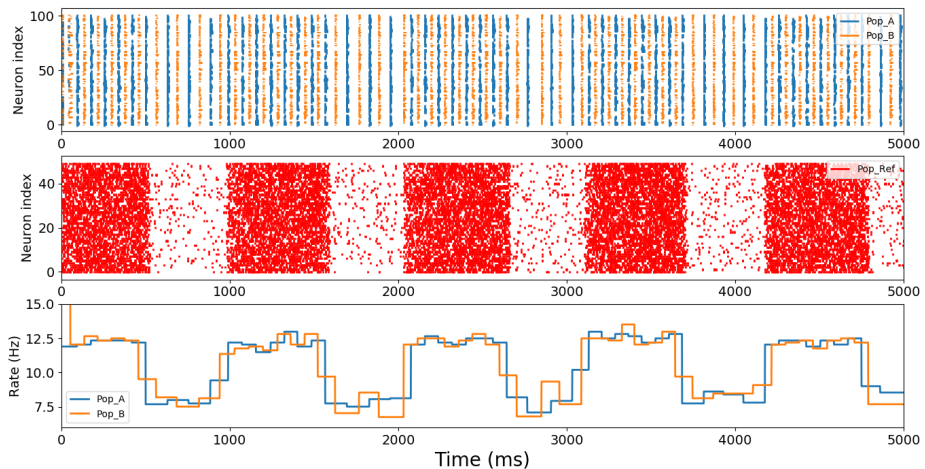


Figura 12.24: CPG implementado en SpiNNaker cuando la pata camina por un terreno de madera. La primera gráfica muestra los spikes de salida del CPG_{AB} , la gráfica intermedia muestra los picos generados por P_{Ref} y la última muestra la frecuencia de disparo de cada población del CPG en Hz . Por lo tanto, la tasa de salida global del CPG será la combinación de ambas.

12.4 Fase III

En la fase anterior del estudio, se evidenció la capacidad de adaptación del sCPG frente a estímulos externos mediante la utilización de datos

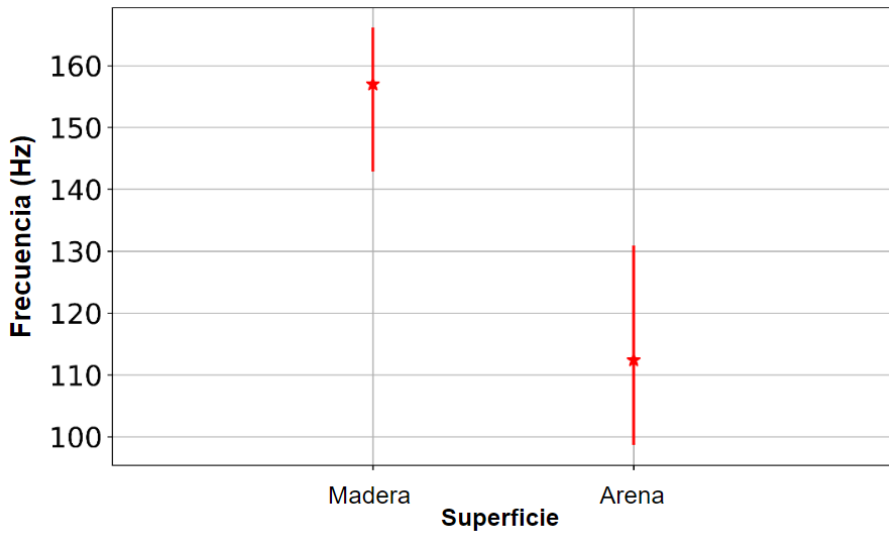


Figura 12.25: Frecuencia generada por P_{Ref} para dos superficies: arena y madera. Las barras muestran la frecuencia máxima, mínima y media. La frecuencia media para madera es de $157Hz$, mientras que para arena es de $112,4Hz$.

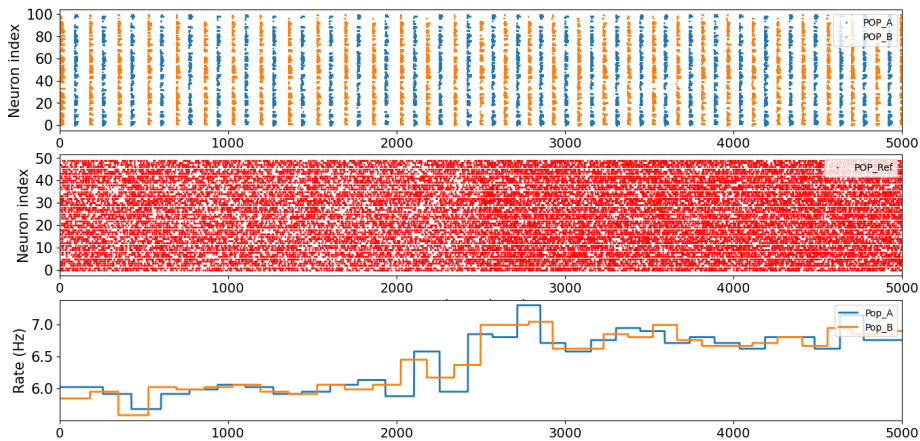


Figura 12.26: Rendimiento del sCPG simulado con Brian 2 durante una transición de un terreno arenoso a uno de madera. Los primeros $2500ms$ corresponden a un terreno arenoso, donde la frecuencia promedio es de $6Hz$, mientras que en los últimos $2500ms$ la frecuencia aumenta hasta $7Hz$ en un terreno de madera.

proporcionados por el FSR. Este proceso incluyó una comparativa entre los resultados derivados del uso del simulador Brian 2 y de la plataforma

12.4. Fase III

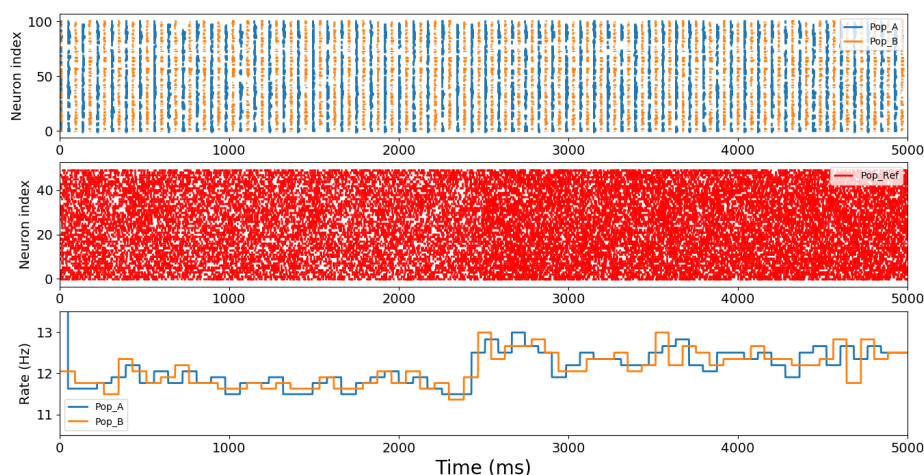


Figura 12.27: Rendimiento del sCPG implementado en SpiNNaker durante una transición de un terreno arenoso a uno de madera. Los primeros $2500ms$ corresponden a un terreno arenoso, donde la frecuencia promedio es de $12Hz$, mientras que en los últimos $2500ms$ la frecuencia aumenta hasta $13Hz$ en un terreno de madera.

neuromórfica SpiNNaker. Dicha comparativa demostró la viabilidad de transferir la red entre ambos entornos, conservando una similitud significativa en los resultados. Adicionalmente, se logró una reducción del 80 % en el número de neuronas, facilitando la transición del modelo SpiNN-5 al SpiNN-3, más adecuado para su integración en sistemas robóticos.

Sin embargo, se detectó que, aunque el sCPG variaba su frecuencia de oscilación en respuesta a cambios en el terreno, tanto en Brian 2 como en SpiNNaker, dicha variación se restringía a un espectro limitado de 2 a 4 Hz. Esto subrayó la necesidad de un ajuste adicional en los parámetros neuronales para incrementar el rango de diferenciación de frecuencias.

Por otro lado, pese a que los resultados se fundamentaron en datos directamente obtenidos del sensor FSR, corroborando las hipótesis iniciales, todavía no se había establecido una comunicación en tiempo real entre el FSR y SpiNNaker en esa etapa del proyecto.

Por consiguiente, en la fase final del proyecto, se implementaron los ajustes mencionados en la sección 11.4. Esto incluyó la integración completa

de la FPGA, encargada de la transmisión en tiempo real de los datos del FSR hacia el sistema SpiNNaker. Adicionalmente, se ampliaron las capacidades de la red neuronal para incluir procesamiento y selección de dirección, empleando el sensor DVS. En este proceso, se implementó la configuración ilustrada en la Figura 11.17 en SpiNNaker, y se aplicó el esquema de la Figura 11.18 en la plataforma FPGA.

12.4.1 Subred adaptativa del FSR

Se llevaron a cabo cuatro pruebas en las que se evaluó el comportamiento de la red cuando se sometía al FSR a una presión variable, consecuencia del cambio de superficie de la plataforma robótica.

12.4.1.1 Presión mínima en el FSR

El primer experimento incluye el estímulo mínimo posible entrante en la plataforma, el cual es de 10Hz , generado cuando se produce una mínima o nula presión sobre el FSR, caminando sobre arena. Los resultados se muestran en la Figura 12.28.

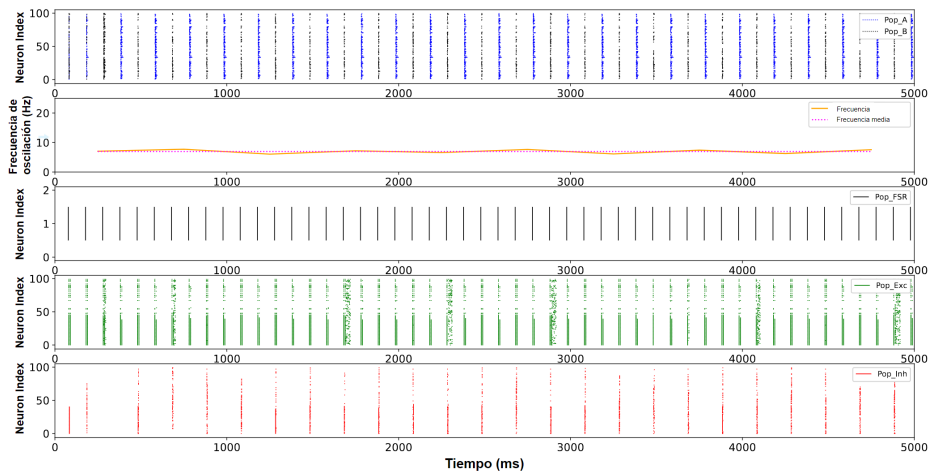


Figura 12.28: Comportamiento de la red CPG sometida a estímulos mínimos del FSR. Gracias a la acción reguladora de las poblaciones P_{Exc} y P_{Inh} , la frecuencia de salida del CPG_{AB} se mantiene estable alrededor de los 7Hz .

En este experimento puede observarse cómo la salida de P_{Inh} y P_{Exc} son

12.4. Fase III

muy similares. Este fenómeno ocurre debido a que en la configuración actual de la red, el CPG_{AB} tiene una mínima excitación por parte de I_{St} , el cual ha pasado de $2,2 nA$ a $0,1 nA$ y, dado que el sensor está arrojando su valor mínimo, ambas poblaciones oscilan por la predominancia de la actividad. debido a este comportamiento, el CPG_{AB} mantiene una oscilación estable de $7Hz$.

12.4.1.2 Presión intermedia en el FSR

El segundo experimento incluye, como estímulo, frecuencias intermedias de $85 Hz$ emitidas por P_{FSR} (P_{Ref} en las simulaciones anteriores), resultado de someter al FSR a una presión intermedia, caminando sobre una superficie firme pero irregular, por lo que el sensor no es presionado completamente por la plataforma robótica. Los resultados se muestran en la Figura 12.29.

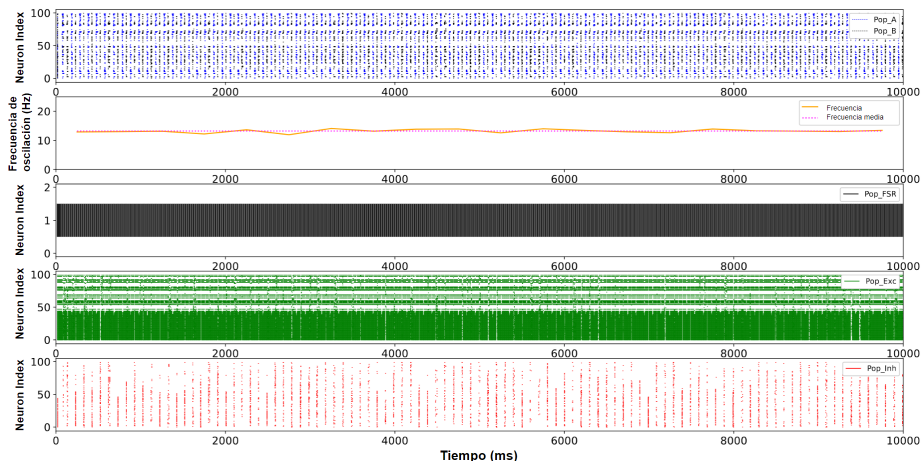


Figura 12.29: Comportamiento de la red CPG sometida a estímulos intermedios del FSR. Las frecuencias intermedias causan un aumento generalizado en las oscilaciones de todas las poblaciones, mientras que P_{Inh} y P_{Exc} mantienen la frecuencia del CPG estable durante toda la duración del experimento entorno a los $13 Hz$.

En este experimento la actividad de P_{Exc} aumenta considerablemente, consecuencia del aumento de presión en el FSR. Es interesante destacar como, aunque aumentar la frecuencia de P_{FSR} debería disminuir la

frecuencia de P_{Inh} , ocurre el efecto contrario debido al aumento en la frecuencia de actividad del CPG_{AB} , el cual excita, aunque débilmente, a P_{Inh} . La frecuencia de oscilación promedio del CPG es de 13 Hz .

12.4.1.3 Presión máxima en el FSR

Como tercer experimento, se sometió al CPG a una entrada constante de 171 Hz , consecuencia del paso de la plataforma robótica por una superficie completamente firme y regular, por lo que el sensor emite su máximo valor de voltaje. Los resultados se muestran en la Figura 12.30.

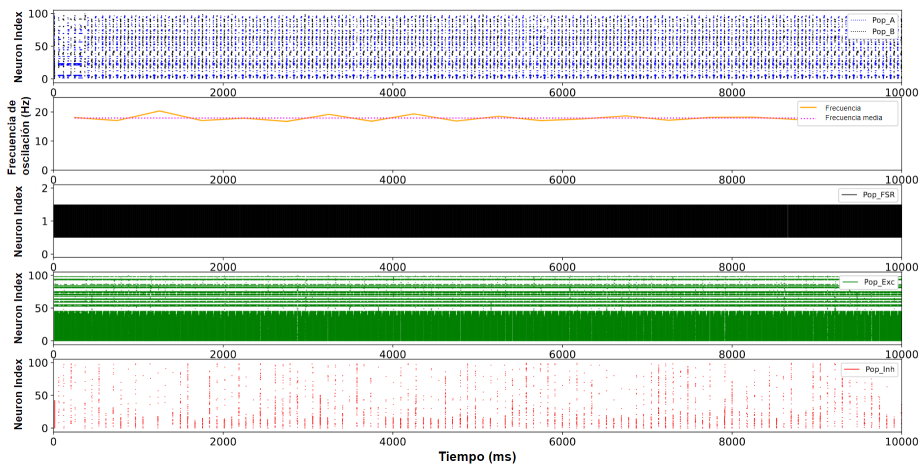


Figura 12.30: Comportamiento de la red CPG sometida a estímulos máximos del FSR. Es interesante observar cómo la significativa excitación causada por P_{Exc} sobre el CPG no solo aumenta su frecuencia hasta los 20 Hz , sino que también conduce a un aumento en el número de neuronas disparando en cada población del CPG durante los momentos en los que están activas.

En este caso, se puede observar cómo P_{Exc} es fuertemente excitada por P_{FSR} , mientras que la actividad de P_{Inh} se reduce considerablemente. En este escenario, no solo aumenta la frecuencia de oscilación del CPG_{AB} , que es alrededor de 20 Hz , sino también el número de disparos en un instante dado en cada población del CPG_{AB} .

12.4. Fase III

12.4.1.4 Presión variable en el FSR

Finalmente, en la Figura 12.31 se presenta un caso en el que la entrada de la población vinculada al FSR es oscilante, donde se puede observar cómo el CPG se adapta a la entrada aumentando o disminuyendo su frecuencia total (la frecuencia obtenida teniendo en cuenta ambas poblaciones del CPG) y su frecuencia individual (la frecuencia de cada neurona en cada población de forma individual en los instantes de tiempo en los que su población está disparando).

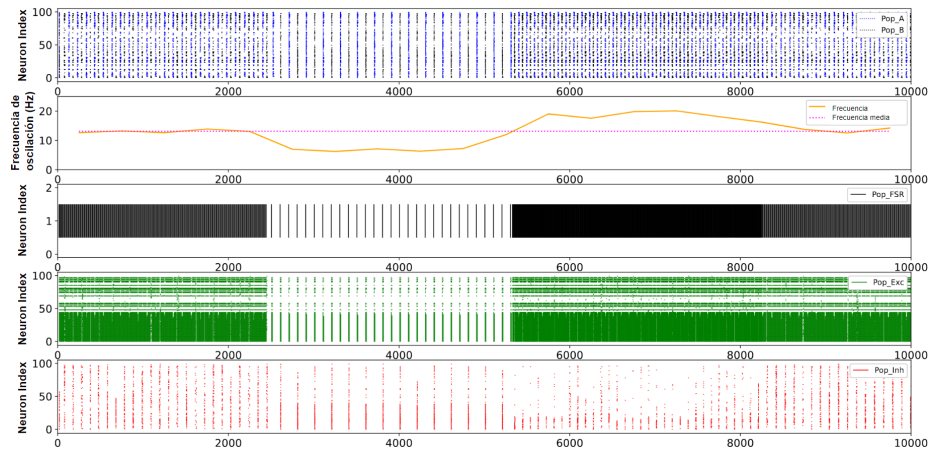


Figura 12.31: Comportamiento de la red CPG en respuesta a entradas variables. El CPG ajusta su frecuencia en función de la entrada asociada con el FSR, demostrando su adaptación a estímulos externos.

Con la respuesta del CPG y la implementación de los módulos FPGA presentados en la Figura 11.12, es posible monitorizar externamente la frecuencia de oscilación del CPG de forma constante, y adaptar el patrón de marcha de la plataforma robótica en consecuencia. En concreto, la plataforma robótica camina lentamente para valores de $f_{CPG_{AB}} < 10 \text{ Hz}$, trotará cuando $10 \text{ Hz} \leq f_{CPG_{AB}} < 15 \text{ Hz}$ y alcanzará la velocidad máxima solo si $f_{CPG_{AB}} \geq 15 \text{ Hz}$.

12.4.2 Subred adaptativa del DVS

Como se mencionó en la sección III del capítulo anterior, es posible recibir hasta 20 000 direcciones por segundo en SpiNNaker, cada una de ellas representando a una de las 128 direcciones posibles vinculadas a la coordenada X de la retina.

De igual forma que con el FSR en la sección anterior, se han realizado distintas pruebas para someter a la red a estímulos puntuales en cada región, con el fin de demostrar cómo la salida de la red WTA puede utilizarse para seleccionar la orientación de la plataforma robótica.

12.4. Fase III

12.4.2.1 Estímulo visual en la ventana izquierda

En el primer experimento, se generó una mayor actividad en la región izquierda del DVS, donde se puede observar que durante toda la simulación el estímulo se generó de manera intermitente en la región P_{IZQ} , generando picos de actividad mínima (por ejemplo, $t = 1000ms$, $t = 2000ms$, $t = 4000ms$) en algunos instantes debido a que el estímulo salía del rango visible de la retina. Los resultados se muestran en la Figura 12.32.

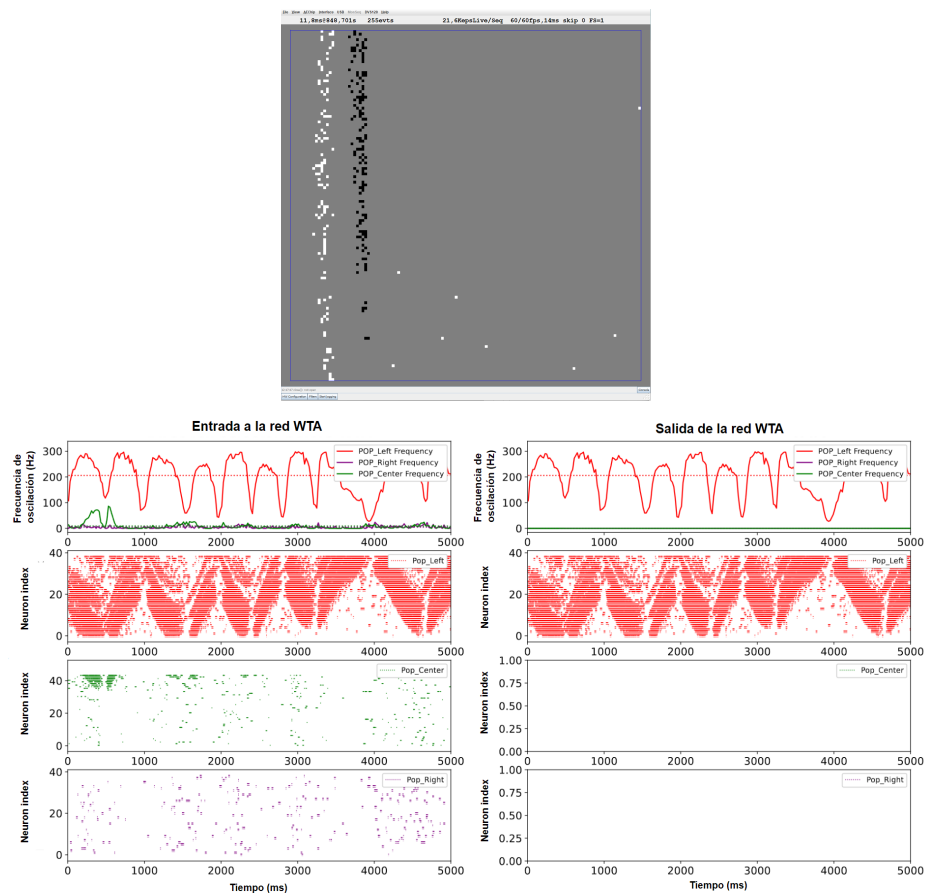


Figura 12.32: Comportamiento de la red WTA ante eventos generados en la región izquierda.

12.4.2.2 Estímulo visual en ventana central

Otra prueba realizada consistía en generar más actividad en la región central del DVS. Los resultados pueden verse en la Figura 12.33, y es posible comprobar que la detección de eventos en todas las ventanas no implica actividad en todas ellas en la salida de la red WTA. La actividad se mantendrá únicamente en la población que haya generado el mayor número de eventos en un tiempo dado.

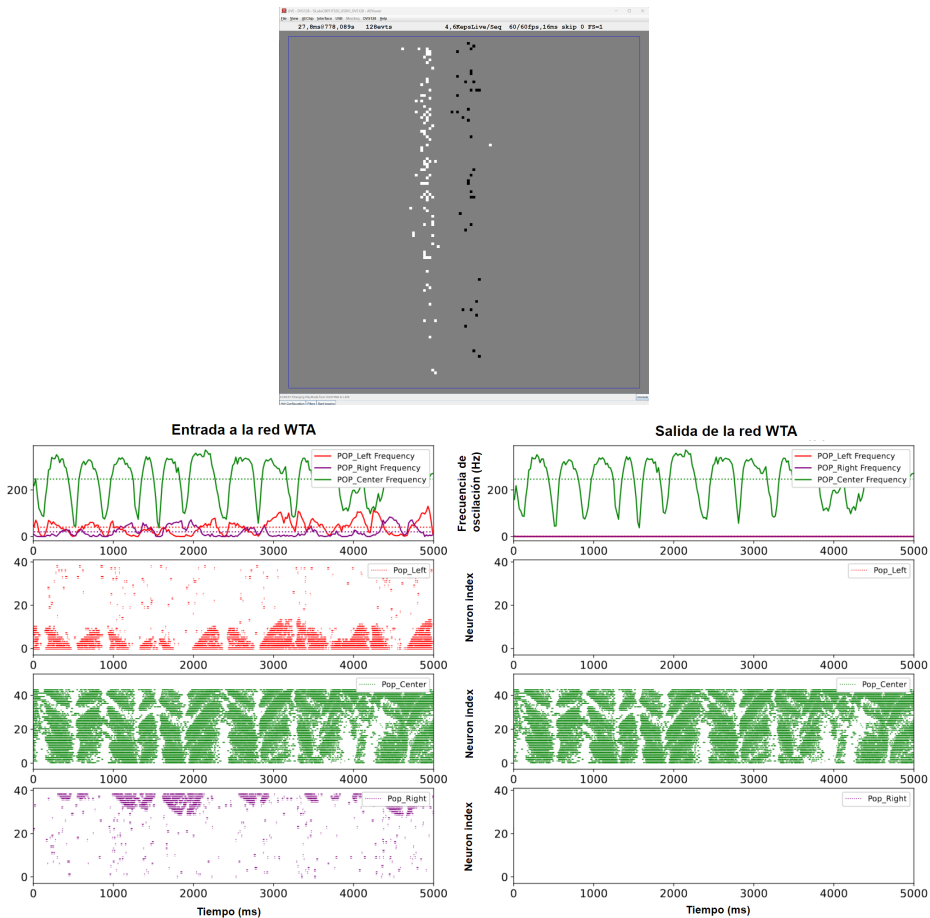


Figura 12.33: Comportamiento de la red WTA ante eventos generados en la región central.

12.4. Fase III

12.4.2.3 Estímulo visual en ventana derecha

El mismo experimento también se realizó generando una actividad aumentada en la región derecha del DVS. En este experimento se verifica nuevamente que, aunque la población central ha producido algunos eventos en los instantes $t = 800ms$ y $t = 3000ms$, la salida de la red WTA sólo contempla los eventos de la población ganadora. Los resultados pueden verse en la Figura 12.34.

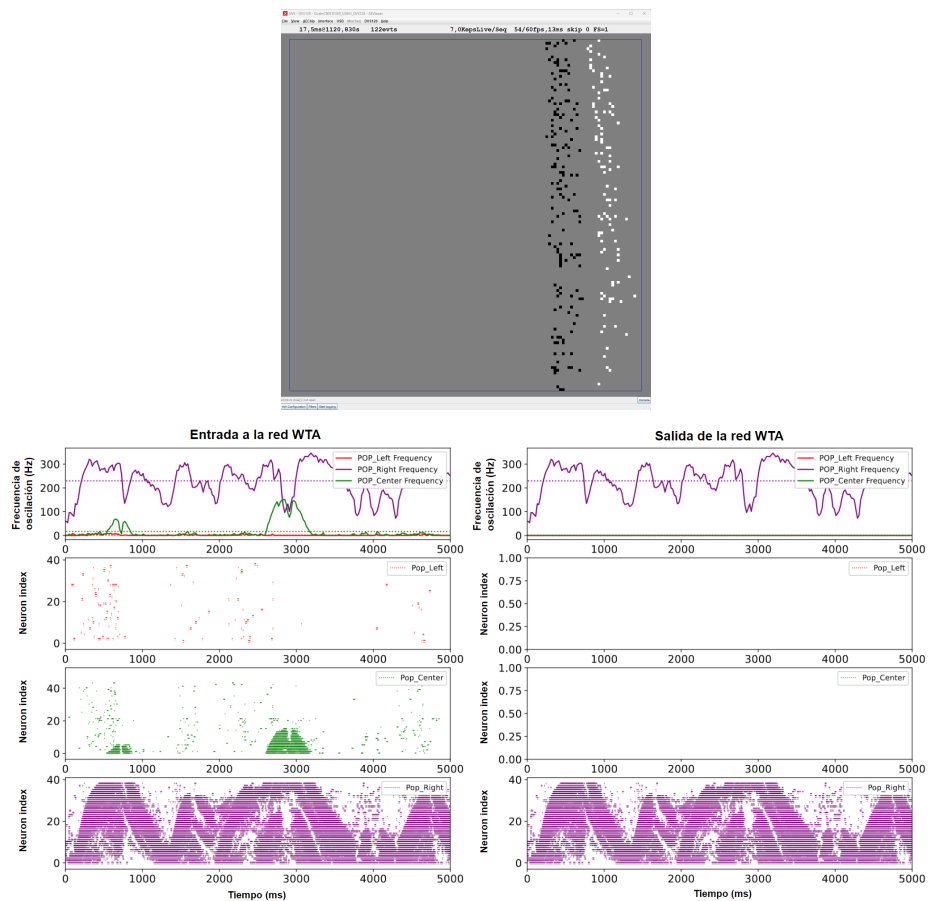


Figura 12.34: Comportamiento de la red WTA ante eventos generados en la región derecha.

12.4.2.4 Estímulo visual aleatorio

Finalmente, también se ha realizado un experimento en el cual la red WTA es sometida a actividad aleatoria sobre todo el campo visual del DVS. El resultado de este experimento puede verse en la Figura 12.35. Es interesante observar cómo la red muestra tolerancia al ruido instantáneo. Por ejemplo, aunque en $t = 1200ms$ la población ganadora en ese instante es P_{IZQ} , en la salida de la WTA, es P_{CEN} la que sigue siendo la población ganadora, ya que P_{IZQ} no ha generado suficiente actividad durante el tiempo necesario para superar la actividad previamente generada por P_{CEN} .

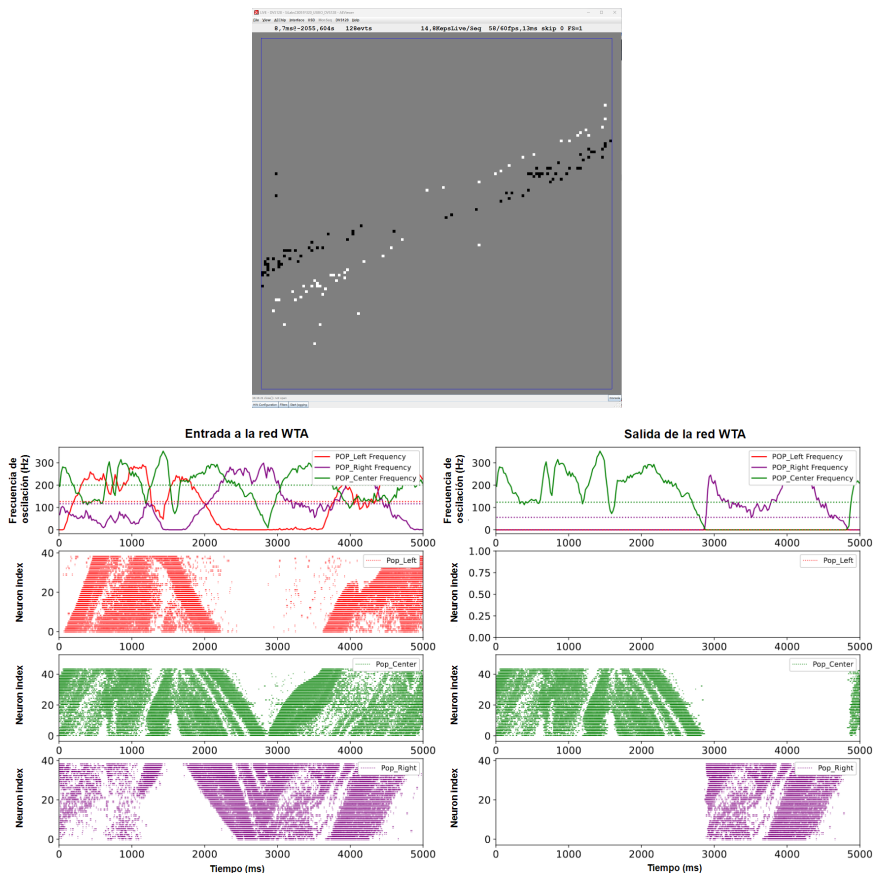


Figura 12.35: Comportamiento de la red WTA ante eventos generados aleatoriamente.

VI

Epílogo

13

Discusión

“Gracias por la aventura. Ahora ve a tener una nueva.”

Up - 2009

El conjunto de desarrollos y experimentos elaborados en esta tesis doctoral han permitido demostrar, por una parte, la capacidad de adaptación del modelo del sCPG a entradas variables del FSR, modificando su frecuencia de oscilación para aumentar o disminuir la velocidad de la plataforma robótica en consecuencia. Por otra parte, se ha demostrado la capacidad del modelo de la red WTA de seleccionar la región espacial que ha generado mayor número de cambios en los píxeles del DVS, para así orientar la plataforma robótica hacia la dirección que genera mayor actividad. Gracias a estas dos características, adaptación de rumbo y velocidad, ha sido posible implementar una rutina "presadepredador" totalmente funcional donde se fusionan las entradas de sensores analógicos y neuromórficos para que la red implementada en SpiN-Naker decida el comportamiento que debe tener la plataforma robótica. Además, los experimentos mostrados en el capítulo 12 han servido para testear el sistema progresivamente y validar su correcto funcionamiento.

Desde el punto de vista bioinspirado, las bases biológicas del CPG descritas en los primeros capítulos muestran las funciones que pueden llegar a desempeñar estas estructuras neuronales en los seres vivos, así como las características de las neuronas que forman este tipo de red biológica. Para la elaboración del CPG usado en esta tesis, se ha priorizado lo máximo posible la similitud de las características que lo definen con estas bases.

En el modelo biológico, las neuronas que componen el CPG pueden ser tónicas o fásicas. En el primer caso, las neuronas tónicas emiten el neurotransmisor a un ritmo constante pero bajo, mientras que las fásicas actúan casi al instante, emitiendo un pulso rápido de actividad para hacer que las neuronas post-sinápticas entren en acción rápidamente. En esta tesis se han implementado ambos tipos de neuronas: por un lado, el CPG_{AB} está compuesto en su totalidad por neuronas tónicas con una inyección de corriente mínima (0.1 nA) pero constante, lo que asegura que, aunque no se recibiera ningún tipo de estímulo por parte del FSR, la oscilación del CPG podría mantenerse en el tiempo, cumpliendo con la generación intrínseca que también presentan los CPG biológicos. Aun así, si se buscara eliminar por completo la actividad de la plataforma robótica (como por ejemplo, al detectar una superficie muy irregular, una alta actividad en un punto específico de su campo de visión, o añadir nuevos sensores que introduzcan nuevos estímulos), solo sería necesario implementar una pequeña corriente inhibitoria entrante en el CPG para eliminar su actividad .

Por otro lado, las poblaciones POP_{EXC} y POP_{INH} actúan como neuronas fásicas, ya que ante la mas mínima señal de excitación recibida, estas son capaces de alcanzar el umbral necesario para producir un disparo neuronal, excitando o inhibiendo al CPG_{AB} casi al instante. Con esta función se ha buscado imitar el control centralizado al que se someten los CPGs ya que, aunque estos pueden funcionar de manera autónoma, a menudo están sujetos a la modulación y el control por parte del sistema nervioso central para adaptarse a las necesidades del organismo.

Otra característica que presentan los CPGs biológicos es su plasticidad, lo que implica que pueden adaptarse y cambiar con el tiempo en respuesta a distintas señales y demandas. En esta tesis se ha implementado una plataforma robótica que se adapta a las condiciones de su entorno tal y como lo haría cualquier ser biológico. El FSR modifica la frecuencia de oscilación del CPG, lo que se traduce en una regulación de la marcha de la plataforma robótica en función del terreno en el que se encuentre, pudiendo asimilar este comportamiento al del sistema vestibular de los seres vivos. Gracias a este comportamiento, se cumple también la condición de variabilidad que presentan los CPGs biológicos, los cuales adaptan sus patrones en respuesta a diferentes condiciones o demandas. De la misma forma, el DVS se utiliza para seleccionar la orientación a seguir por la plataforma robótica, imitando la capacidad de decisión que poseen los seres vivos a la hora de tomar un rumbo en función de lo que capta su sistema óptico (presas, peligros, obstáculos, etc.).

Sin embargo, también hay características del CPG presentado en esta tesis que pueden diferir con respecto a los biológicos. Por un lado, el número de neuronas utilizados se ha estimado únicamente teniendo como prioridad la estabilidad del sistema completo, reduciendo casi a cero los escenarios en los que la salida de la red es inestable o no coherente. En los CPGs biológicos enfocados a la locomoción, se requiere una red neuronal relativamente compleja (dependiendo de factores como la especie o su patrón de movimiento), pero se estima que en mamíferos, como los seres humanos, estos CPGs pueden estar compuestos por miles de neuronas[208]. Por otro lado, mientras en este trabajo el CPG solo se encarga de ajustar la velocidad de la plataforma robótica, en el entorno biológico estas estructuras neuronales controlan simultáneamente la velocidad, la dirección, el equilibrio y otros aspectos de la locomoción en función de las necesidades específicas del organismo y las condiciones ambientales[209, 210].

Otro punto que debe comentarse y que está relacionado con el anterior, es el uso de una red WTA para la toma de decisión en cuanto a

la orientación de la plataforma robótica se refiere. Si bien estas redes pueden modelar ciertos aspectos de la competencia y la inhibición observados en algunas funciones cerebrales, no se asemejan de manera directa o específica a una estructura neuronal biológica particular del sistema nervioso, quedando esta función en el entorno biológico reservada, de nuevo, a los CPGs.

Además, hay trabajos que no solo regulan la frecuencia de la red, sino también su amplitud y su fase. Por ejemplo, en [211, 212] se hace uso de una red que permite controlar y modificar estos parámetros, cada una de forma independiente. Aunque no incluye ningún tipo de información sensorial que afecte al comportamiento de la red, estos trabajos se acercan un poco más a las funciones básicas que poseen los CPGs biológicos.

Finalmente, en cuanto a coste computacional y energético se refiere, la red presentada en este trabajo hace uso de un total de 579 neuronas, mientras que en otros trabajos como [213, 214] hacen uso de un número mucho menor de neuronas para controlar la plataforma robótica. Debido al elevado número de neuronas, han surgido problemas como el sobreajuste de los pesos sinápticos de la red, una necesidad de recursos hardware mayores, o el coste energético necesario para mantener la red en funcionamiento.

14

Trabajo futuro

*“En caso de que no nos veamos:
buenos días, buenas tardes y
buenas noches.”*

El show de Truman - 1998

Esta tesis doctoral deja abierta algunas posibles vías que pueden ser muy interesantes de explorar en futuros proyectos:

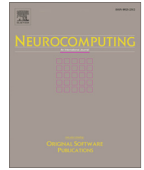
- Estudiar la incorporación de otros tipos de sensores, tanto neuromórficos como convencionales, con el fin de dotar a la plataforma de una mayor percepción de lo que ocurre en su entorno.
- Implementar pautas de aprendizaje en todos los niveles de la red neuronal, como por ejemplo el reconocimiento de patrones con la retina o el reconocimiento de sonidos con una cóclea neuromórfica.
- Optimización de la red neuronal, reduciendo el número de neuronas utilizadas y la cantidad de fallos producidos durante las simulaciones.
- Estudiar cómo los patrones de movimiento pueden ser almacenados y reproducidos en cualquier momento, utilizando un único estímulo como disparador de la secuencia de locomoción.

VII

Anexos

A

Neuromorphic adaptive spiking CPG towards bio-inspired locomotion



Neuromorphic adaptive spiking CPG towards bio-inspired locomotion

Pablo Lopez-Osorio^a, Alberto Patiño-Saucedo^b, Juan P. Dominguez-Morales^c,
Horacio Rostro-Gonzalez^{b,*}, Fernando Perez-Peña^{a,*}



^a School of Engineering, Universidad de Cádiz, Spain

^b Department of Electronics, DICIS-University of Guanajuato, Mexico

^c Robotics and Technology of Computers Lab., Universidad de Sevilla, Spain

ARTICLE INFO

Article history:

Received 10 May 2021

Revised 22 June 2022

Accepted 27 June 2022

Available online 30 June 2022

Communicated by Zidong Wang

Keywords:

Neurorobotics

SpiNNaker

Central pattern generator

Spiking neural network

Neuromorphic hardware

Adaptive-learning

ABSTRACT

In recent years, locomotion mechanisms exhibited by vertebrate animals have been the inspiration for the improvement in the performance of robotic systems. These mechanisms include the adaptability of their locomotion to any change registered in the environment through their biological sensors. In this regard, we aim to replicate such kind of adaptability through a sCPG. This sCPG generates different locomotion (rhythmic) patterns which are driven by an external stimulus, that is, the output of a FSR sensor to provide feedback. The sCPG consists of a network of five populations of LIF neurons designed with a specific topology in such a way that the rhythmic patterns can be generated and driven by the aforementioned external stimulus. Therefore, eventually, the locomotion of an end robotic platform could be adapted to the terrain by using any sensor as input. The sCPG with adaptation has been numerically validated at software and hardware level, using the Brian 2 simulator and the SpiNNaker neuromorphic platform for the latest. In particular, our experiments clearly show an adaptation in the oscillation frequencies between the spikes produced in the populations of the sCPG while the input stimulus varies. To validate the robustness and adaptability of the sCPG, we have performed several tests by varying the output of the sensor. These experiments were carried out in Brian 2 and SpiNNaker; both implementations showed a similar behavior with a Pearson correlation coefficient of 0.905.

© 2022 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introduction

It is well known that, in biology, rhythmic locomotion is produced by a neural structure called Central Pattern Generator (CPG) [1]. This structure is located at the spinal cord and it usually comprises of two neural populations which produce an alternating output of spikes. Eventually, these output spikes are used to activate the muscles fibers.

This approach of using CPGs can be borrowed to create locomotion in robotics. There are several possibilities to implement a CPG: using coupled-oscillators, using Artificial Neural Networks (ANNs) or using Spiking Neural Networks (SNNs). The closest biological implementation is to use an SNN. These networks are based on biological plausible neuron models and synaptic connections that implement biological features. The field of research called neuromorphic engineering aims to implement these networks on elec-

tronics, mimicking the way living beings have solved complex problems by using both analog and digital circuits. The neuromorphic robotics field puts together both the neuromorphic engineering and the roboticist communities [2]. The use of neural structures made of spiking neurons, coming from the neuromorphic engineering field, within robotics results in the need for less resources, less power consumption and a simplification of the algorithms [3] in comparison with traditional approaches based on ANN or coupled oscillators.

One of these neural structures is the CPG. This structure generates a rhythmic pattern at its output, which can be used within robotics to generate locomotion. Thus, a CPG creates gaits that are suitable to use within a robotic platform. These structures can generate a very stable pattern even without sensory information or brain activity [4].

As briefly shown, there is a growing community of researchers that are exploring the possibility of using spiking neurons within a CPG to create locomotion in robots [1]. Although there are some works that introduced local sensory feedback as [5,6], they used oscillators to model the behaviour of the CPG instead of spiking neuron models. Conversely, most of the previous works, spike-

* Corresponding authors.

E-mail addresses: pablo.osorio@uca.es (P. Lopez-Osorio), alpatisa@hotmail.com (A. Patiño-Saucedo), jpdominguez@atc.us.es (J.P. Dominguez-Morales), hrostro@ugto.mx (H. Rostro-Gonzalez), fernandoperez.pena@uca.es (F. Perez-Peña).

based related, in the literature present an open-loop CPG which does not include any sensory information: in [7], the actuation of a lamprey-like robot is done by using an open-loop CPG and neuromorphic hardware. Then, in [8,9], the authors proposed a CPG implemented on an Field Programmable Gate Array (FPGA) and the SpiNNaker platform [10]; these three works do not offer the possibility of changing the originally produced pattern in real time. However, they showed that implementing CPGs using spiking neurons uses less power. There is a more recent paper which allows both real time functioning and pattern variation but without including any sensory information [11]. In [12], the open-loop CPG is implemented on Loihi [13] using an astrocytic network, producing two different gaits with 24 motor neurons. In [14], authors propose the implementation of a CPG with the possibility of changing the amplitude, frequency and phase online without any sensory input required. The authors also pointed out that the architecture should include any sensory feedback to modify the behavior of the CPG. Regarding works that include sensory information, in [15], the CPG is built using coupled-oscillators instead of spiking neurons and the feedback to the CPG is included into the control loop of the equations of the CPG. In [16], 12 simulated neurons modulated by sensory feedback are used to build the CPG. Instead of what we propose, a SNN with a adaptation, they achieve different gaits by either moving the location or increasing the number of neural structures. Instead of what we propose, a SNN with adaptation, the authors achieve different gaits by either moving the location or increasing the number of neural structures. The neuron model used in that paper is Izhikevich [17] instead of the Leaky Integrate-and-Fire (LIF) model proposed in this work to reduce the computational resources used. The sensory information is used to adapt the time duration of each phase of the frequency switching to enable the actuators to reach the commanded position. Finally, in [18], a neuromorphic sensor has been used to select which predefined gait of the CPG should be activated. A very recent work by Strohmer et al. [19] suggests the use of a combination of non-spiking and spiking neurons to simulate a closed loop amplitude regulating network. The non-spiking neuron used in such work is in fact a spiking neuron (Leaky Integrate-and-Fire) with a threshold high enough to avoid firing. The spiking neuron model used is the adaptive exponential integrate-and-fire AdEx, a more complex model than the one used in this work and not easily implementable on neuromorphic hardware. The article focuses on investigating how combining spiking and non-spiking neurons can create a network of sensorimotor neurons capable of shaping the output of the network based on the analogue input. Although locomotion is very vaguely mentioned as an application of such a network, it is not an issue addressed in this paper. The authors also suggest the compatibility of their model with neuromorphic hardware. However, this compatibility is restricted only to the non-spiking neuron and to a cloud service called CloudBrain. In this regard, our proposal differs in different aspects, the first of them is the exclusive use of spiking neurons with a simpler model than the one presented in [19]. The second difference and contribution in this work is the use of information from a real sensor (Force Sensitive Resistor (FSR)) that feeds the network, which in the case of the locomotion of a legged robot will allow us to modify its dynamics depending on the terrain where the robot moves. The third and most important aspect is the validation of our model on neuromorphic hardware, specifically on the widely used SpiNNaker neuromorphic processor, using high level software tools that facilitate the reproducibility of this work by the scientific community. Finally, in this work we have carried out successful physical tests by injecting the force generated by the FSR sensor into different surfaces on one leg of the robot. This experiment can be replicated on the other legs of the robot and, together with a suitable interconnection of these, can generate the desired locomotion. In most

of these works, the use of an external input to the CPG changes the performance of it. This performance can be defined as a neuromodulation. It has been shown that this modulation is essential to alter the behavior of a neural structure by modifying the synaptic connections [20].

Finally, there are works where the main focus is on the learning process of the robotic platform: in [21], the authors proposed a rewarding-learning process to teach a hexapod robot how to walk without any previous knowledge. A couple of sensors (a standard camera and a gyroscope) are used to provide the rewarding signal to the neural network based on a CPG. Although they used a digital version model neuron of the LIF, they do not use neuromorphic hardware to implement the neural architecture; a Raspberry Pi is used instead. A similar approach based on reinforcement learning, but without using spiking neurons, was proposed to improve the locomotion of the NAO robot [22]. Another approach is used in [23], where the authors have two hexapod robots: an expert and a student. The student learns or imitates the gait of the expert by using a one-layer feedforward SNN and a Dynamic Vision Sensor (DVS) camera as input. However, the possibility for the robot to adapt to its environment is not implemented.

To summarize, the objective of this paper is to design and deploy a spiking architecture that makes possible the interaction of a spiking CPG with its environment. An external agent, i.e. a FSR, is introduced as the feedback stimulus to the network. This agent can modify the gait generated by the CPG. Therefore, the locomotion of a robotic platform (any legged robot) could be adapted to the terrain. Furthermore, the spiking network presented in this paper allows the introduction of the feedback sensory information on the loop of the CPG to provide adaptation. This adaptation SNN could be used with any sensor as input stimulus.

The rest of the paper is structured as follows: Section 2.1 introduces the materials used in this work, including the simulator and hardware used. The implemented methods are described in 2.2, together with the SNN model. Then, the results obtained are presented and discussed in Section 3. This section is divided into two different subsections: first, the experiments run using the software simulator and then, the same experiments run on the hardware platform. Finally, the conclusions are presented.

2. Materials and methods

2.1. Materials

This section describes both the software and hardware used to perform the experiments.

2.1.1. Brian 2

Brian 2 [24] is a neural simulator for SNNs written in Python programming language. Thus, it is a cross-platform which is available in different operating systems. In contrast to other SNN simulators such as NEURON [25] or PyNN [26], Brian 2 is highly flexible and it is easily adaptable with new non-standard neuron models and synapses. Brian 2 can be used to model and simulate complex problems faced by neuroscientists, as well as giving faster and more robust results before implementing the solution on a hardware platform.

2.1.2. SpiNNaker

SpiNNaker [27,10,28] is a massively parallel, multi-core computing system designed by the Advanced Processor Technologies (APT) Research Group from the University of Manchester. It was designed under the Human Brain Project (HBP) [29] for simulating parts of the brain by using SNNs. SpiNNaker machines consist of SpiNNaker chips, which have eighteen 200-MHz ARM968 processor

cores each [30]. This allows an asynchronous communication infrastructure for sending short packages (each of them representing a particular neuron firing) [31] identified using Address Event Representation (AER) [32]. Different SpiNNaker machines were built and commercialized, including SpiNN-3 and SpiNN-5, with 4 and 48 SpiNNaker chips each, respectively. They also include the spinnlinks [33], which allow real-time input/output interfacing with neuromorphic sensors and other neuromorphic platforms such as FPGAs [34–36]. A PyNN-based [26] software package called sPyNNaker [37] can be used to design and implement SNNs on these machines. The recently built million-core machine is at the School of Computer Science at the University of Manchester that can be used through the HBP portal. In this work, a SpiNN-5 machine was used to run the simulations proposed.

2.2. Methodology

We simulated our Spiking Central Pattern Generator (sCPG) model on a standard computer using the Brian 2 Simulator to characterize the network dynamics, to analyse the number of neurons per population needed and to adjust the network parameters. The simulation results guided the subsequent neuromorphic implementation using the SpiNNaker platform.

2.2.1. Neuron model

The LIF model is used to implement the neuron on both the software simulator and the SpiNNaker hardware platform [38]. The model is defined within the Eqs. (1) and (2).

$$\tau_m \frac{dV}{dt} = -(V - V_r) + RI(t) \tag{1}$$

$$\text{if } V(t) = V_{th} \text{ then } \lim_{\delta \rightarrow 0, \delta > 0} V(t + \delta) = V_r \tag{2}$$

where V is the membrane potential of the neuron, R represents the resistance of the membrane, τ_m the time constant of the neuron, V_r the resting potential, V_{th} the threshold and $I(t)$ is the stimulus.

2.2.2. Network model

The SNN model depicted in Fig. 1 was designed based on the neuron model presented in Section 2.2.1. The main objective of the proposed model is to generate a constant oscillation between

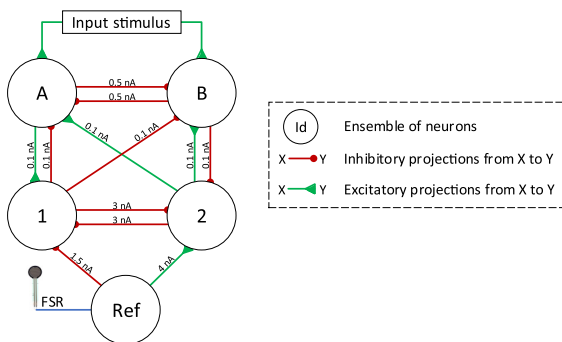


Fig. 1. Diagram of the proposed spiking neural network architecture. Populations A and B are self-excited and self-inhibited with a probability of 25% and 75% and weights of 4 nA and 1.5 nA, respectively. A and B are reciprocally inhibited with a probability of 75%, and are both injected with a constant external current in order to start generating the oscillation. Populations 1 and 2 are similar to A and B, and are connected to them in a way that greater spiking rates in the Ref population (higher values from the output of the FSR sensor) will produce higher oscillation frequencies in AB, whereas lower oscillation frequencies will be obtained for lower FSR values.

the spikes produced in ensembles A and B, whose frequency varies depending on the value read from the FSR sensor. Thus, the proposed CPG is able to automatically adapt its behavior depending on the input stimulus. The SNN shown in Fig. 1 is the core of the sCPG. The output spikes of populations A and B will be eventually used to interface the actuation structure of the legs of the robotic platform.

The SNN consists of different parts which are described next. It is important to note that each of the ensembles (also called populations) have the same number of neurons. The number of neurons in each population was set based on different experiments, which are presented in Section 3.

The main block of the architecture is the so-called CPG_{AB} , which consists of the populations A and B represented in Fig. 1. These populations are self-excited and self-inhibited with a probability of 25% and 75% and weights of 4 nA and 1.5 nA, respectively. Moreover, a 75% probability of having inhibitory synapses between neurons from the two aforementioned populations is also present. Thus, when one of the populations is producing spikes, the opposite is inhibited and, thus, generating the desired oscillation pattern. Populations A and B are injected with a constant external current in order to start generating the oscillation. Therefore, for these two populations Eq. 3 is used instead of Eq. 1.

$$\frac{dV}{dt} = \frac{V_r - V + R(I_{exc} - I_{inh} + I_{st})}{\tau_m} \tag{3}$$

Where I_{st} is the current injected to neurons in populations A and B. This value was set to 2.2 nA, which is sufficient for producing the desired rhythmic pattern.

Furthermore, populations 1 and 2 (CPG_{12}) both have the same number of neurons as those in CPG_{AB} , and are also interconnected in the same way. The projections between CPG_{12} and CPG_{AB} are depicted in Fig. 1, and follow the same aforementioned probabilities (25% for excitatory and 75% for inhibitory projections). The

Table 1

Neuron parameters for the proposed CPG in both the Brian 2 simulator and the SpiNNaker hardware platform.

Parameter	Value
u_{reset}	-55.0 mV
u_{rest}	-55.0 mV
u_{th}	15.0 mV
τ_m	6.0 ms
τ_{syn_e}	5.0 ms
τ_{syn_i}	8.75 ms
C_m	0.1875 nF
Δ_t	1.0 ms
I_{bias}	2.2 nA

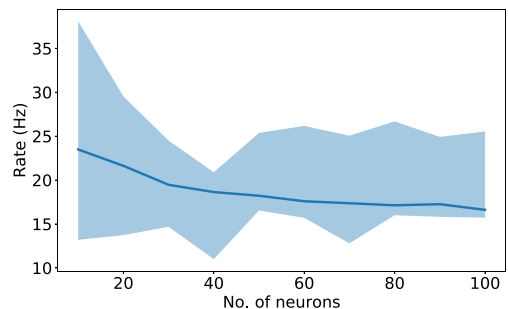


Fig. 2. Average, maximum and minimum rate values obtained for each number of neurons per population. The trace shows the mean, and, its shadow, the maximum and minimum rate of all the one thousand simulations performed.

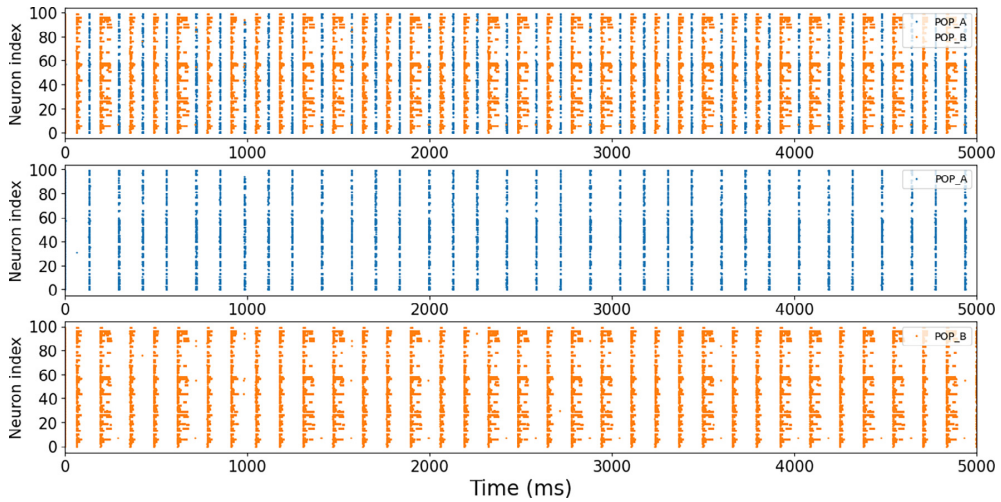


Fig. 3. Simulation of the CPG_{AB} with a I_{St} value of 10 nA. Increased oscillation frequency may be observed along with an increase in the amount of noise when compared to Fig. 4.

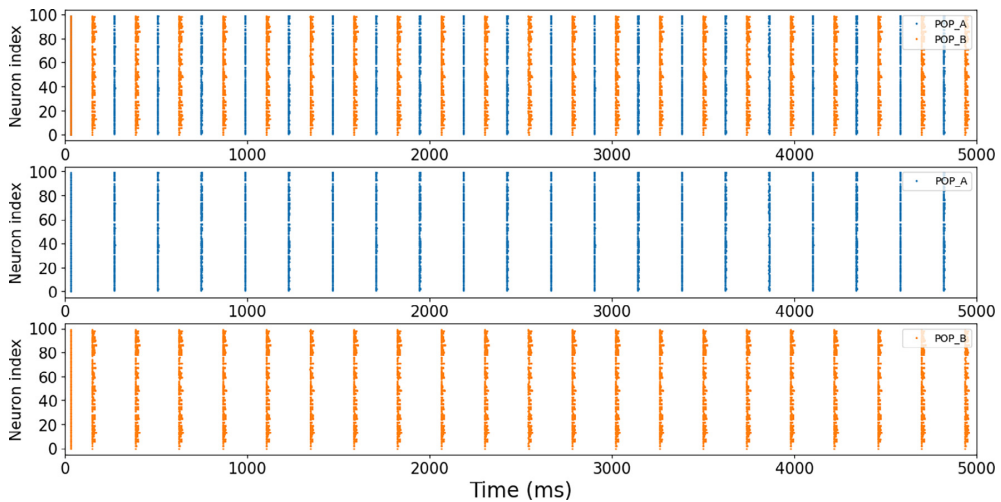


Fig. 4. Brian 2 simulation of the CPG_{AB} with a I_{St} value of 2.2 nA. A lower oscillation frequency and almost total noise removal can be observed when compared to Fig. 3.

weights between the different populations of the proposed model are specified in the same figure.

Finally, the reference population (*Ref*) was implemented as a Poisson distribution with variable frequency. In contrast to the populations from CPG_{AB} and CPG_{12} , the number of neurons in *Ref* was set to 50. These neurons are connected to populations 1 and 2 following the scheme presented in Fig. 1. This number of neurons was set to 50 since it was the optimum for producing biologically-plausible spiking rates in the population, with maximum and minimum frequencies close to the biological counterpart [39]. The spiking rate of the Poisson distribution depends on the values obtained from the FSR sensor used as input to population *Ref*. This sensor should be placed at the end of the leg of the robot. This sensor provides values between 0 and 5 V. Since the spinal cord ventral horn motor neuron alpha, which is the biological neuron taken as reference, has a spike rate between 10 and 171 Hz [39], a linear

regression was established in order to match the frequency of the Poisson distribution with the voltage value read from the sensor.

Based on these three blocks (CPG_{AB} , CPG_{12} and *Ref*) and the connections among them, the proposed behavior explained at the beginning of this section was achieved. Therefore, following Fig. 1, different scenarios can be analyzed. In the case where the spike rate of population *Ref* is greater than the oscillation frequency of CPG_{AB} , population 1 will be inhibited and population 2 will be excited. Since population 1 is excited from A, but the number of spikes is lower than the ones that are inhibiting the same population from *Ref*, population 1 will have very low activity. The opposite occurs in population 2, which will be inhibited from B but excited from *Ref* in a stronger way and, thus, having considerable more activity than population 1. The activity from population 2 will excite CPG_{AB} , increasing its oscillation frequency. Conversely, the opposite happens when the spike rate of *Ref* is

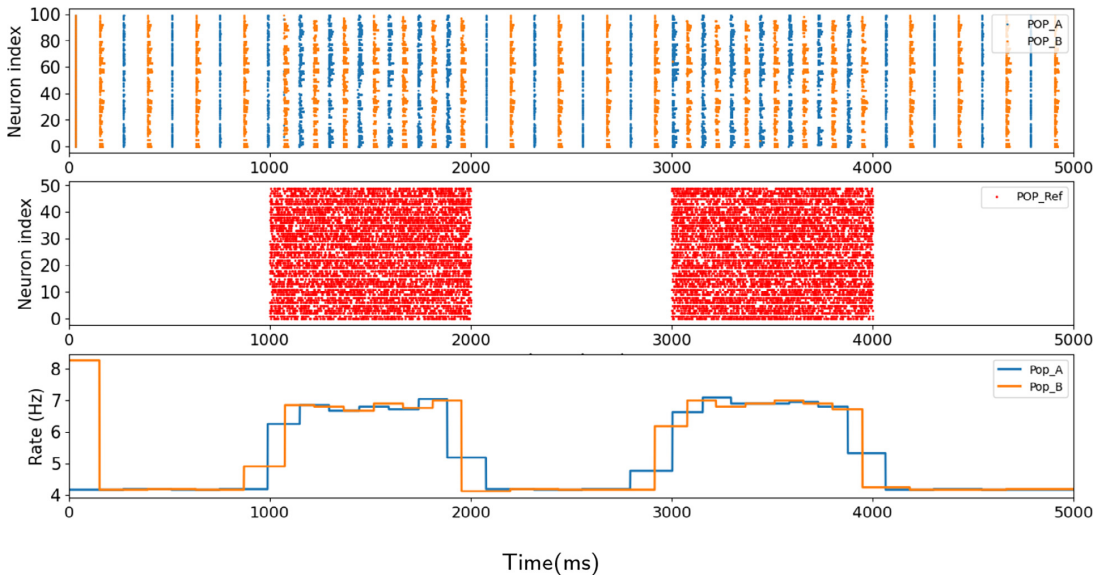


Fig. 5. Brian 2 simulation of the proposed SNN model when having a 5 V input from the FSR sensor for 1000 ms between $t = 1000$ ms and $t = 2000$ ms and between $t = 3000$ ms and $t = 4000$ ms (generating a Poisson distribution of 171 Hz in Ref). For the rest of the simulation time, the output of the sensor was set to 0 V. The third row shows the firing rate of each population of the CPG in Hz. Thus, the global output rate of the CPG will be the combination of both.

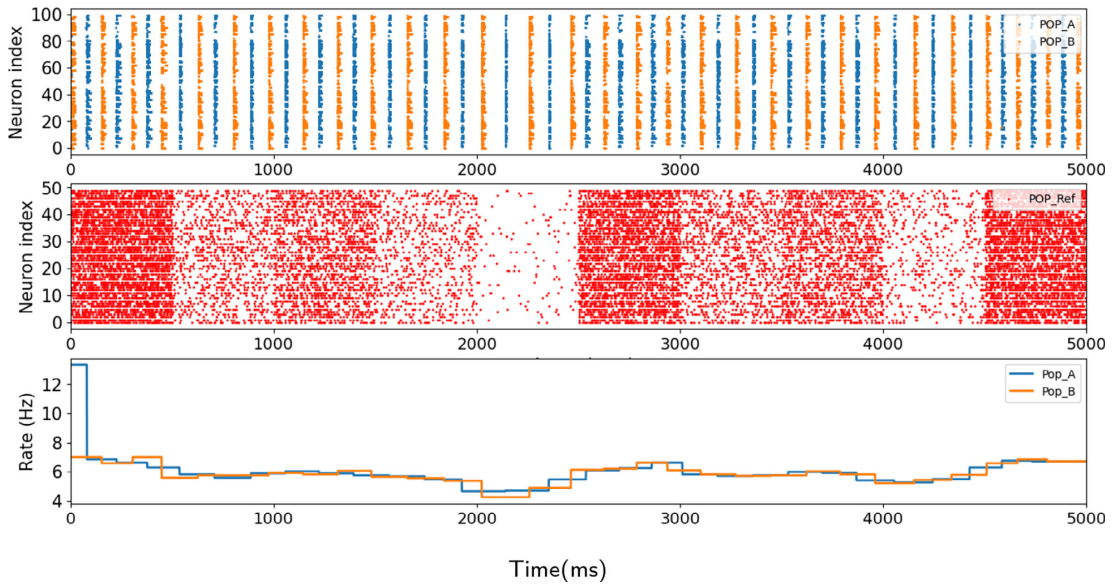


Fig. 6. Brian 2 results obtained when simulating random voltage readings from the FSR sensor (random values change every 500 ms). The third row shows the firing rate of each population of the CPG in Hz. Thus, the global output rate of the CPG will be the combination of both.

lower than the oscillation frequency of CPG_{AB} , where population 1 will have more activity than 2 and, therefore, inhibit CPG_{AB} in order to reduce its frequency. As a summary, CPG_{AB} is the main one that will eventually drive the motors. Then, CPG_{12} is used to interface the feedback sensor. The latter would adapt its frequency to the feedback sensor output and then, this frequency would be transmitted to the former. Therefore, the dynamical properties of the feedback sensor are not directly in contact with the main locomotion generator. As a result, the proposed network is able to adapt the frequency of the CPG based on an input stimulus. This model

was simulated in Brian 2 and emulated using SpiNNaker, and the results are shown in Section 3.

3. Results and discussion

3.1. Brian 2 simulations

The first experiment was performed to determine the number of neurons per population of the CPG that was needed to achieve a stable rate value along the simulations. The neuron parameters

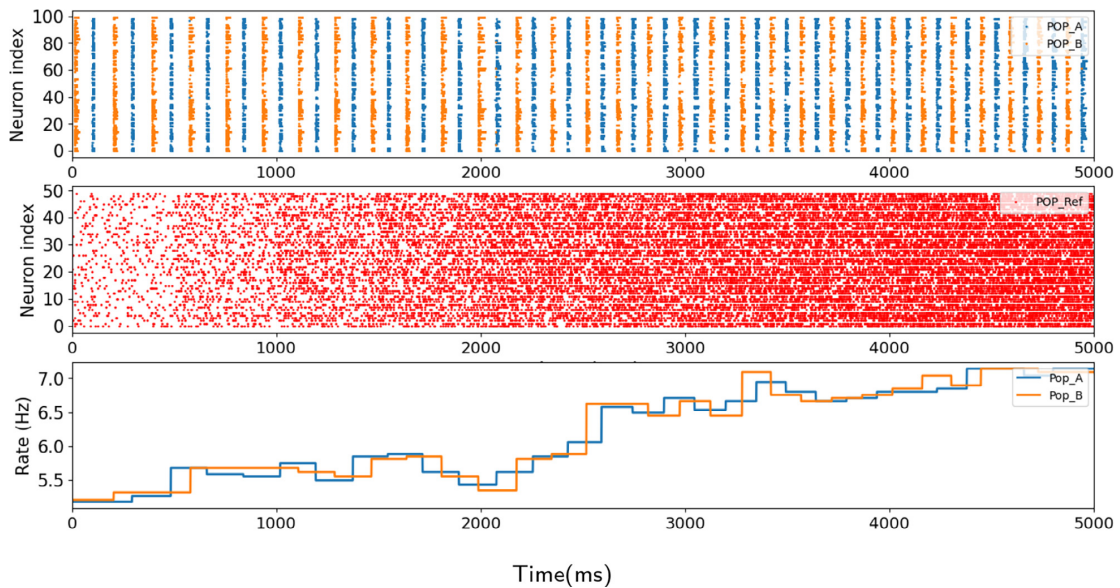


Fig. 7. Brian 2 results obtained when simulating a continuous increase in the frequency of Ref (increments of 20 Hz per 500 ms). The third row shows the firing rate of each population of the CPG in Hz. Thus, the global output rate of the CPG will be the combination of both.

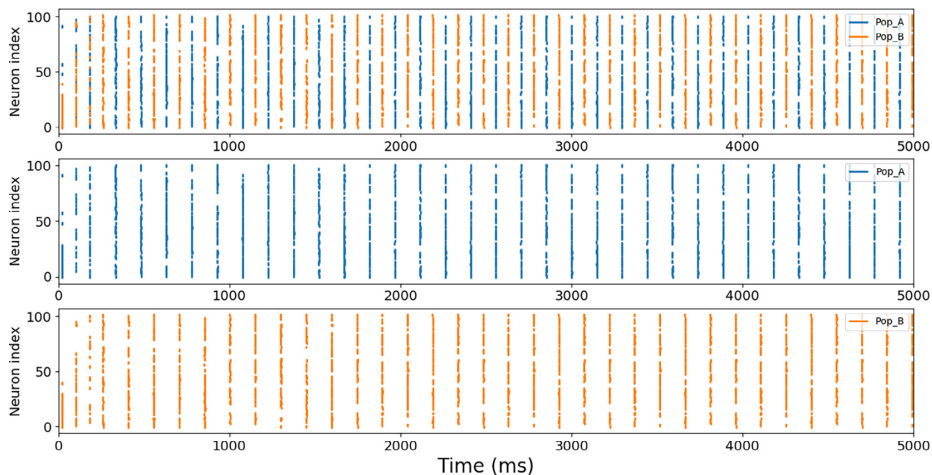


Fig. 8. SpiNNaker implementation of the CPG_{AB} with a constant oscillation frequency.

used for all the experiments are shown in Table 1. These parameters were optimized and found by means of a grid search algorithm [40]. Fig. 2 shows the maximum, minimum and mean values obtained for the rate of the simulations performed. A thousand simulations per number of neurons were run. As the number of neurons increased, the rate achieved is more stable and the standard deviation becomes lower. Starting from 40 neurons per population in the CPG_{AB} , the standard deviation is less than 1.5 and the behaviour is more stable. A hundred neurons per population shows the most stable output rate with the minimum deviation.

Once the number of neurons per population was fixed, to verify if the architecture presented in Section 2.2 behaved as expected, the operation of the CPG_{AB} in isolation was analyzed. This experiment ensured that it was able to produce a constant oscillation. Then, the operation of the same CPG was examined once intercon-

nected with the CPG_{12} , performing tests with different stimuli to analyze the results obtained. Finally, the entire architecture was connected and analyzed in different scenarios based on the external input from the FSR sensor.

3.1.1. CPG_{AB} analysis

The topology of the CPG_{AB} can be seen in Fig. 1, where green arrows denote excitatory connections and red arrows denote inhibitory connections. As mentioned in Section 2.2.2, I_{St} is a constant current injected to all neurons in populations A and B, with a fixed value of 2.2 nA.

This current is the minimum value required to produce the oscillatory pattern in the CPG. While the proposed topology can work with higher values of I_{St} , this generates a higher frequency of oscillation and a noticeable increase in the noise introduced in

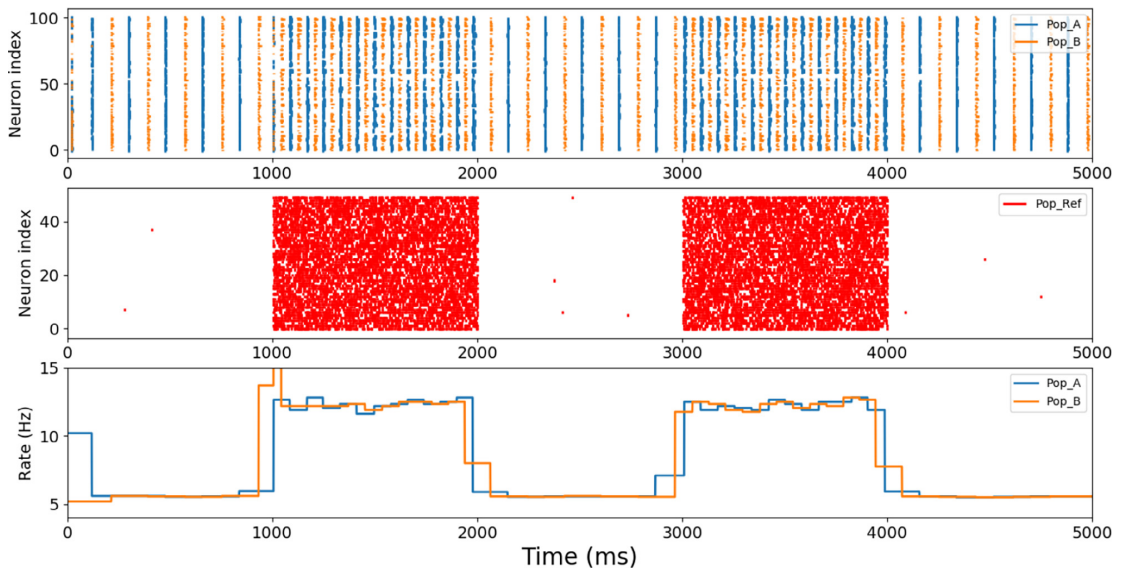


Fig. 9. Spinnaker implementation of the CPG for extreme values of the input stimulus (0.1 Hz during most of the time, and 171 Hz between $t = 2000$ ms and $t = 3000$ ms, and between $t = 4000$ ms and $t = 5000$ ms). The third row shows the firing rate of each population of the CPG in Hz. Thus, the global output rate of the CPG will be the combination of both.

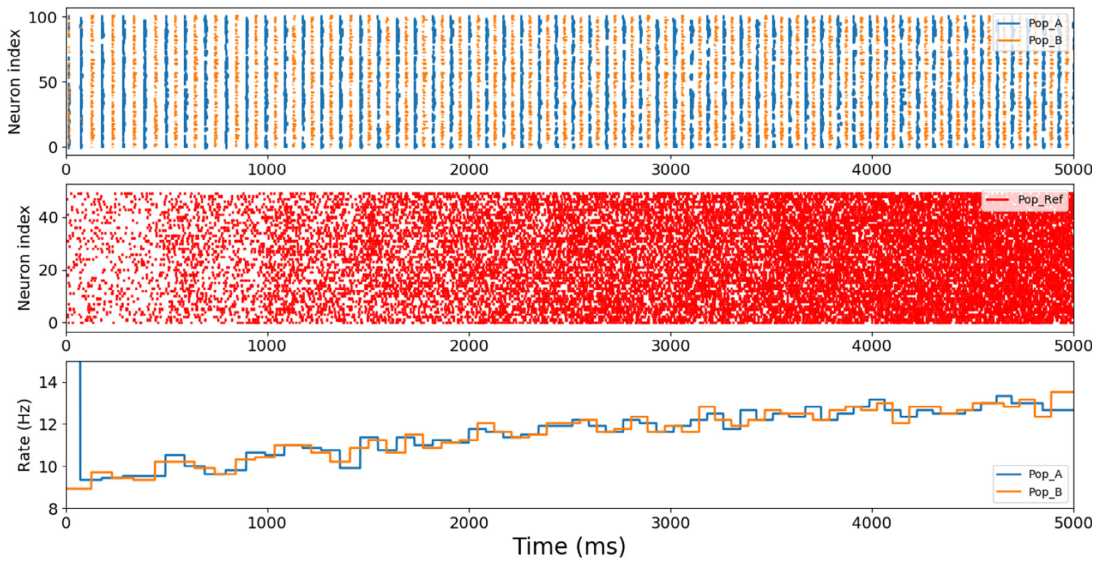


Fig. 10. Spinnaker implementation of the CPG for ten increasing values of the stimulus rate (increments of 20 Hz per 500 ms). The third row shows the firing rate of each population of the CPG in Hz. Thus, the global output rate of the CPG will be the combination of both.

the simulation (see Figs. 3 and 4). Thus, this value was set to 2.2 nA in order to be able to more easily appreciate the effect of feedback on the CPG. The results of the simulation performed are shown in Fig. 4, where a frequency of, approximately, 5.8 Hz was obtained for each population, while the total frequency of the CPG was 11.6 Hz.

3.1.2. Analysis of the effect of the sensor when used as input to the SNN

In order to study the robustness of the network against sudden changes in the oscillation frequency, an experiment where the val-

ues read from the sensor were alternating between maximum and minimum voltage peaks was performed. Initially, a 5 V input was simulated in 1 s and 2 s, both with a duration of 1 s. This made neurons in population *Ref* to fire at a frequency of, approximately, 171 Hz during this period. Before 1 s, between 2 s and 3 s, and after 4 s the sensor readings corresponded to 0 V. Fig. 5 shows the results of this simulation. It can be observed that, at time zero, since no information was being received from the sensor, *Ref* had no activity. Therefore, population 1 was excited by CPG_{AB} and population 2 was inhibited. In turn, population 1 slightly inhibited

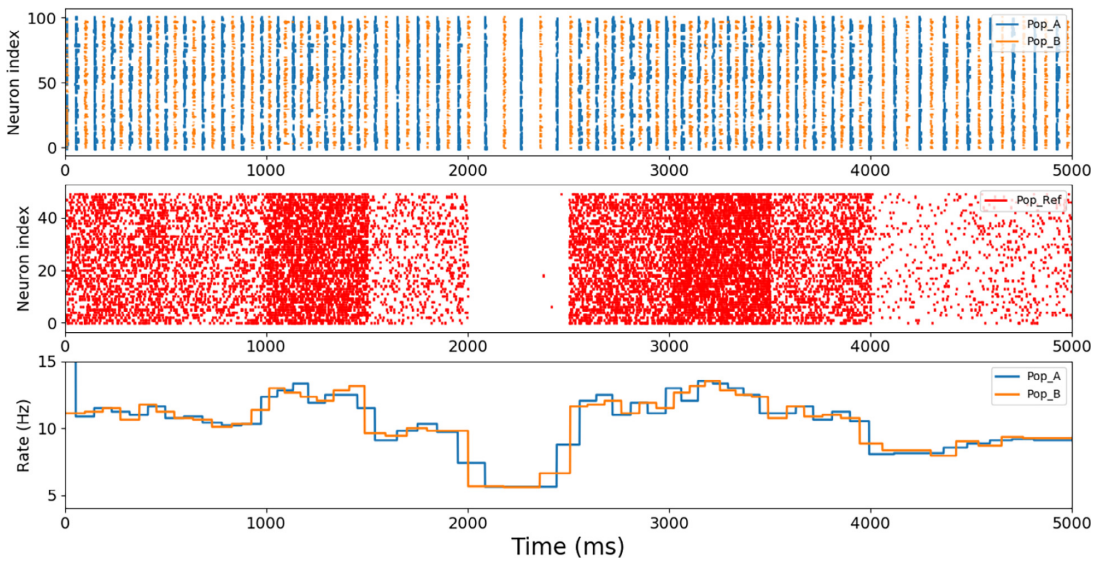


Fig. 11. SpiNNaker implementation of the CPG for ten random values of the stimulus rate (between 0 Hz and 200 Hz). The third row shows the firing rate of each population of the CPG in Hz. Thus, the global output rate of the CPG will be the combination of both.

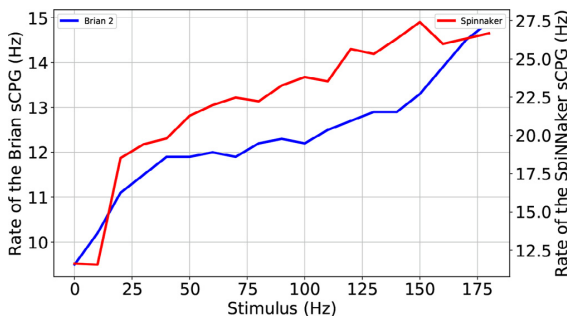


Fig. 12. Comparison of the results obtained in both Brian 2 simulation and SpiNNaker implementation. The plot shows the rate generated by the sCPG when the input stimulus (population *Ref*) is changed from 0 Hz to 180 Hz. The blue trace shows Brian 2 results (left y-axis) and the red trace the SpiNNaker results (right y-axis).

CPG_{AB}. At 1 s, *Ref* started firing at a frequency of approximately 171 Hz, exciting population 2 and inhibiting population 1. During this period, the former excited *CPG_{AB}*, increasing its oscillation frequency. After 2 s, population 1 started dominating population 2 again, slightly inhibiting *CPG_{AB}* again. Exactly the same behavior can be seen again starting at 3 s. In this figure it can be observed how the frequency of *CPG_{AB}* increased considerably between 1 s and 2 s and between 3 s and 4 s, obtaining minimum frequencies of 8 Hz and maximum frequencies of 15 Hz.

On the other hand, different simulations of more realistic cases were performed. Initially, 10 random voltage values were used in order to simulate different readings from the FSR sensor. These values were updated every 500 ms. Specifically, the frequency values for Poisson distribution in *Ref* were (171, 40, 80, 30, 5, 130, 50, 76, 20, 150) Hz. In Fig. 6 it can be seen how *CPG_{AB}* adapts its oscillation frequency based on the inputs stimuli, obtaining maximum and minimum frequency peaks of 15 Hz and 8 Hz, respectively.

After this, a constant increase in the values of the readings from the sensor was simulated. In particular, increases in steps of 20 Hz per 500 ms were introduced in the frequency of *Ref*. To check the

performance limits of the *CPG_{AB}*, the last injected frequency value exceeded up to 17% the maximum theoretical value of 171 Hz. Fig. 7 shows the results of this experiment.

As can be seen in the figure, although there is a significant increase in the amount of noise, the oscillation frequency of *CPG_{AB}* does not increase, achieving a maximum frequency of 15 Hz.

3.2. Running the model on SpiNNaker

The proposed sCPG model was tested in the SpiNNaker neuro-morphic hardware. The neuronal model is defined in sPyNNaker [37], a PyNN-based software interface that allows a quick prototyping and implementation of spiking neural networks in the SpiNNaker platform.

The spiking neuron model is the LIF with fixed threshold and decaying-exponential post-synaptic current, whose parameters are given in Table 1. These parameters were chosen so as to emulate the neuron dynamics of the simulations in Brian 2.

In order to show that the model achieves a good performance in SpiNNaker, we performed three tests: first, by verifying that the constant oscillations of the *CPG_{AB}* were observed and matched the rates presented in Brian 2. Then, by implementing the whole sCPG with increasing rates of the input sensor represented by the reference population. Finally, by testing the sCPG under random stimuli.

3.2.1. *CPG_{AB}* implementation

For the implementation of the CPG in Spinnaker (see Fig. 8) 100 neurons were used, each with a constant current I_{St} equal to 2.2 nA, like in the equivalent Brian 2 experiment. The measured rate of the oscillatory pattern of populations A and B was 11.62 Hz, which matches the rate measured in the Brian 2 simulation. Although some spikes were lost in the raster plot compared to Brian 2, which can be attributed to limited support of floating-point calculations in SpiNNaker, the pattern appears consistent and with little noise.

3.2.2. CPG under different stimuli

The SpiNNaker implementation of the full CPG, including the feedback network was tested under different conditions of the

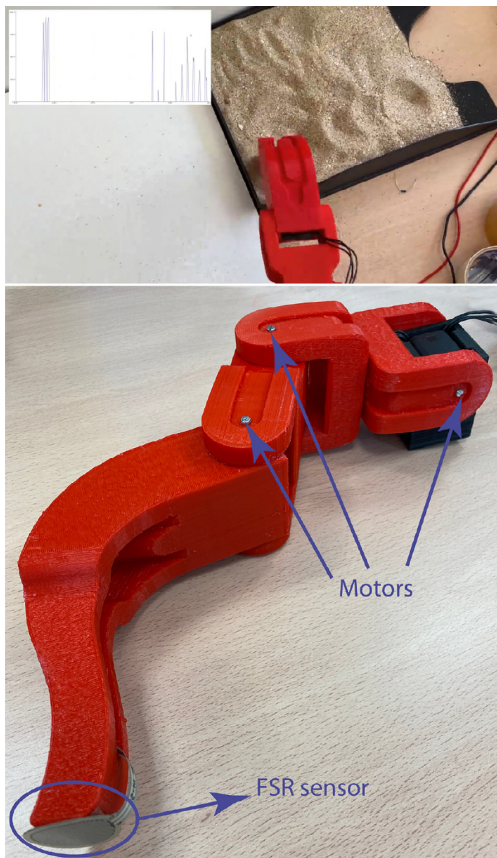


Fig. 13. Top: snapshot of the setup used to perform the experiments with the FSR sensor for two different terrains is shown. A video of the experiments can be seen in: <https://youtu.be/3s89p3qYnCU>. In the video, the leg was moving by hand. Bottom: picture of one of the legs to show where the FSR sensor and motors are located.

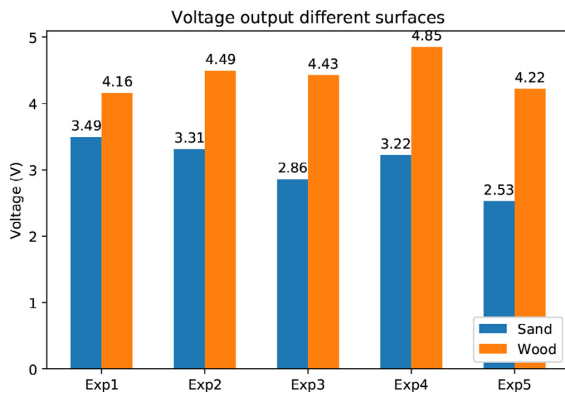


Fig. 14. The mean output voltage of the FSR sensor for two different terrains is shown. A single leg was moved up and down in two different surfaces five times. The blue bars correspond to the experiments performed in a sand surface, while the orange bars correspond to the experiments performed in a woody terrain. At the top of the bars, the mean value is shown. The standard deviation is 0.34 for the sand surface and 0.24 for the wood surface. As it can be seen, the values obtained in a softer terrain (sand) are lower than in a stiffer (wood) terrain.

stimulus. First, we simulated a sudden change of the value of the sensor, represented by the rate of the Poisson generator in the *Ref* population (see Fig. 9). This rate was set to 0.1 Hz for the first 2 s of the simulation and oscillated between 171 Hz and 0.1 Hz for the next four seconds, in order to appreciate both regimes of the CPG. It can be seen how, with a low *Pref* frequency, the first feedback population dominates and the rate of the output oscillatory is low, at around 12.5 Hz. With a high *Pref* frequency, it is the second feedback population which dominates and the measured oscillatory pattern displays a peak frequency of 27.5 Hz.

Figs. 10 and 11 show the spiking response of the SpiNNaker CPG to increasing rates of *Ref* and to random values of *Ref*, respectively. The two regimes can be clearly observed in the spiking response of populations 1 and 2 and in the measured oscillatory rates of Populations A and B, proving that the feedback mechanism works correctly.

3.3. Comparison between the results obtained in Brian 2 and SpiNNaker

Fig. 12 shows the comparison made between both approaches: the simulations on Brian 2 and the implementation on the SpiNNaker platform. In this experiment, the same input stimulus from the sensor was used for both approaches, which went from 0 Hz to 180 Hz. The rates generated by both sCPGs were very similar. In the case of the Brian 2, the operation frequency of the sCPG ranged from 9.5 Hz to 14.9 Hz, and from 11.62 Hz to 26.66 Hz in the case of the SpiNNaker platform. The calculated Pearson correlation coefficient between both is 0.905.

3.4. Testing the sCPG models with real stimuli from the FSR sensor.

The values read from the FSR are used as feedback, and they determine the kind of surface on which the robot is stepping on. As shown in Fig. 14, the feedback values determine if the surface is softer or stiffer. Since the frequency of the sCPG is correlated with the output value from the FSR, the speed of the robot will be also directly correlated with the kind of surface that the robot is stepping onto. The amplitude of the leg during the swing movement will remain the same while the speed of the swing will be decreased if the surface is soft and increased if the surface is stiff. This behaviour can be checked in Fig. 19.

Tests have been performed with real stimuli received from an FSR sensor implanted in one of the robot's legs. Specifically, wood and sand terrains have been used to verify that the hypotheses proposed in this paper are correct. Fig. 13 shows the setup of the experiments performed. The mean values obtained can be seen in Fig. 14. Five experiments were performed to compare a firmer ground (wood) and a more irregular one (sand). The bars on the plot represent the mean voltage whenever the leg is completely touching the surface. As it can be seen, the values obtained on the wooden surface exceed 4 volts, while in sand they are below 3.5 volts.

Thanks to the relationship established between the value of the voltage obtained by the FSR and the *Pref* population firing rate, it is possible to stimulate to a greater or lesser extent the oscillation frequency of the CPG_{AB} , which directly translates into a substantial variation of the walking speed of a robot. These experiments have been simulated in Brian 2 and implemented in SpiNNaker in order to verify the performance of the whole system. The results from these experiments are presented in Figs. 15 and 17 for the Brian 2 simulations, and in Figs. 16 and 18 for the SpiNNaker deployments.

As shown in Fig. 19, the difference between the frequency generated by *Pref* (which maps the FSR output) in sand and wood is, approximately, 40%, while the results show that the CPG_{AB} oscilla-

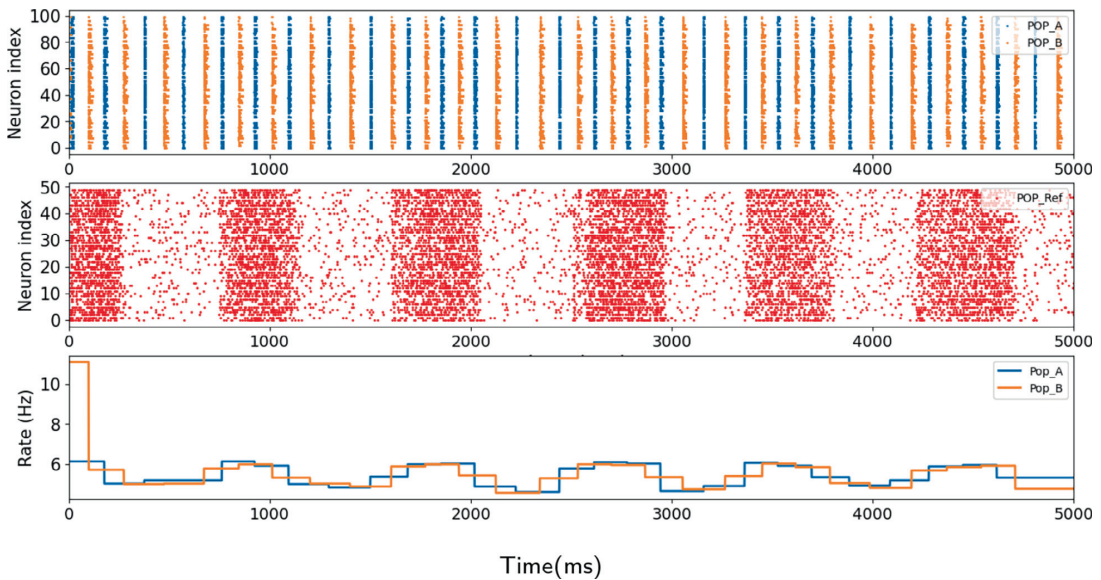


Fig. 15. Brian 2 simulation of the CPG for values of the FSR on one leg when walking on a sandy terrain. The first row shows the output spikes of the CPG_{AB} , the middle row shows the spikes fired by the population mapping the FSR sensor output and the third row shows the firing rate of each population of the CPG in Hz. Thus, the global output rate of the CPG will be the combination of both.

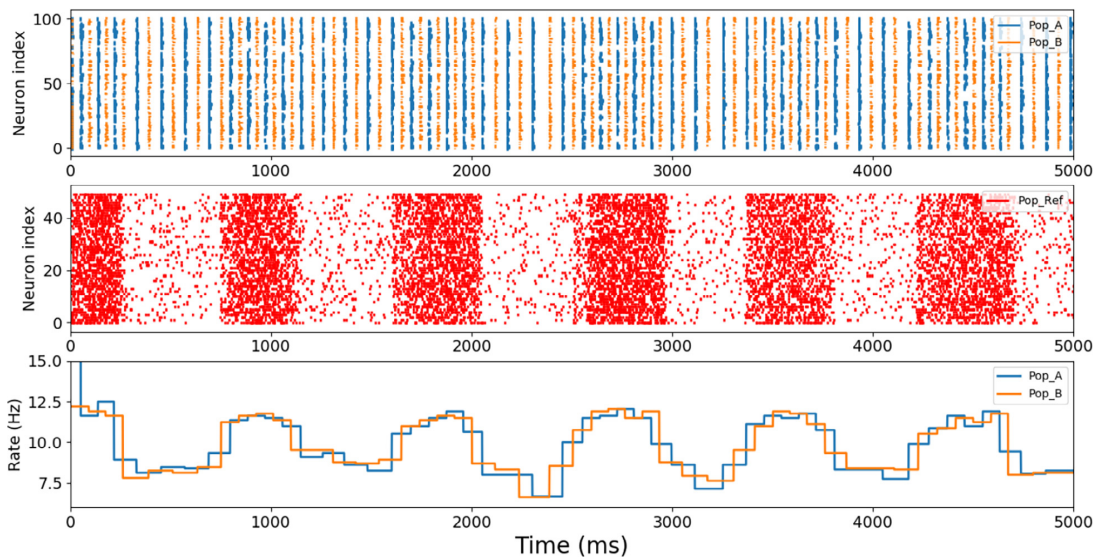


Fig. 16. CPG implemented on SpiNNaker when the leg is touching a sandy terrain. The first row shows the output spikes of the CPG_{AB} , the middle row shows the spikes fired by the population mapping the FSR sensor output and the third row shows the firing rate of each population of the CPG in Hz. Thus, the global output rate of the CPG will be the combination of both.

tion is 17% lower when the leg is in the sandy terrain than when it is in wooden terrain. The frequency difference shows that it is possible to adapt the sCPG frequency to the external conditions given by the FSR output.

A new experiment involving a transition between sandy and wooden terrains was performed to demonstrate the variability of the frequency of the spiking CPG. For this purpose, only the maxi-

um values obtained from the *Pref* population were taken, which correspond to the values obtained when the leg was fully supported on the ground, either in sand or wood. The same experiment was performed in Brian 2 (Fig. 20) and SpiNNaker (Fig. 21): the first 2500 ms correspond to the robot walking on sand, while the last 2500 ms correspond to the robot walking on wood.

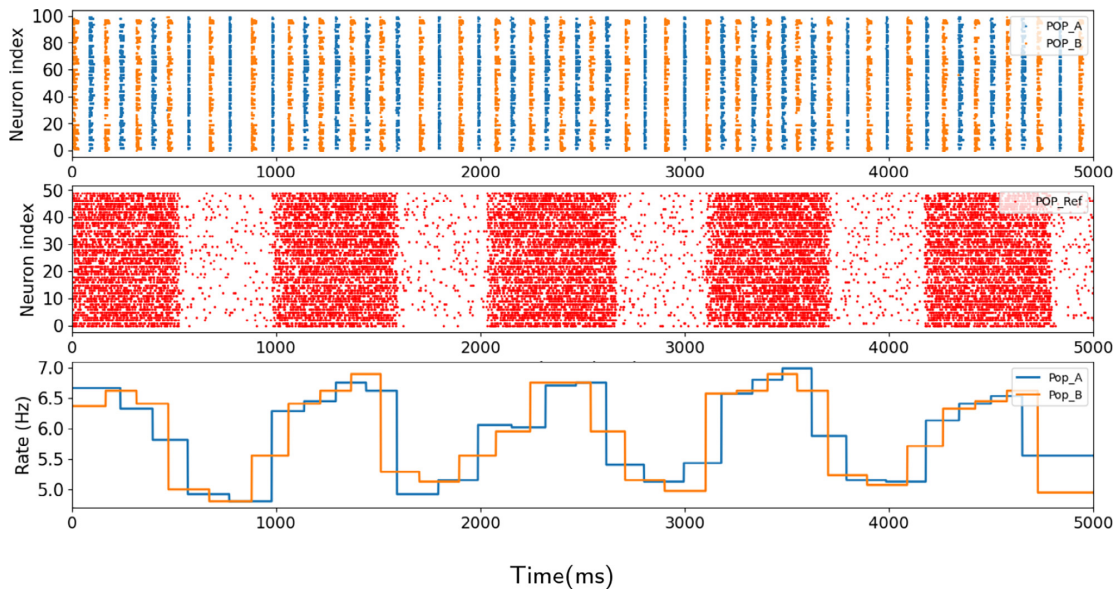


Fig. 17. Brian 2 simulation of the CPG for values of the FSR on one leg when walking on a wooden terrain. The first row shows the output spikes of the CPG_{AB} , the middle row shows the spikes fired by the population mapping the FSR sensor output and the third row shows the firing rate of each population of the CPG in Hz. Thus, the global output rate of the CPG will be the combination of both.

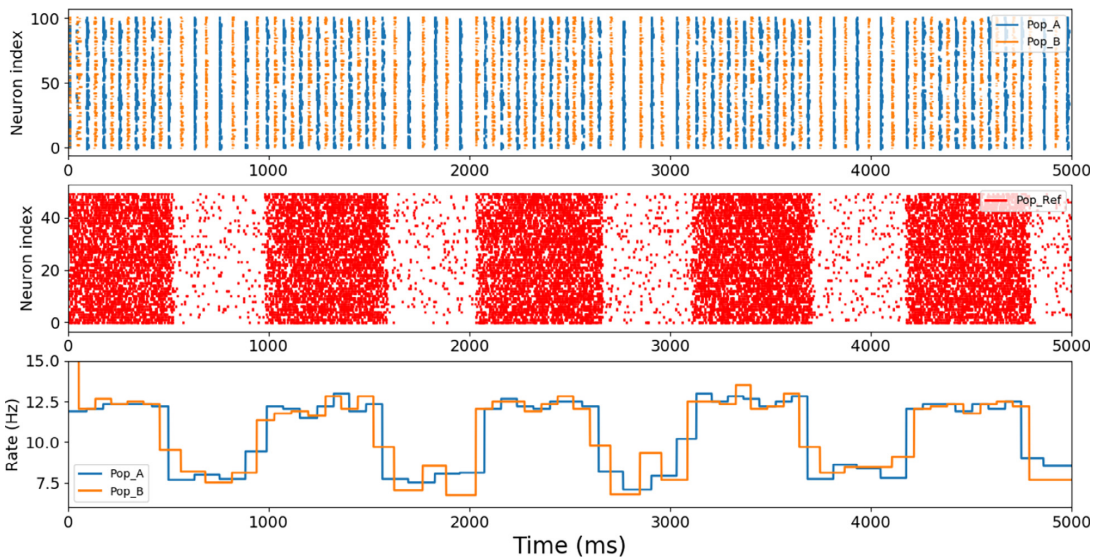


Fig. 18. CPG implemented on SpiNNaker when the leg is touching a wooden terrain. The first row shows the output spikes of the CPG_{AB} , the middle row shows the spikes fired by the population mapping the FSR sensor output and the third row shows the firing rate of each population of the CPG in Hz. Thus, the global output rate of the CPG will be the combination of both.

As can be seen in Fig. 20, the frequency of the CPG in Brian 2 for the sandy terrain oscillates around 6 Hz, while for the wooden terrain it increases up to 7 Hz.

On the other hand, in SpiNNaker (Fig. 21), the frequency of the CPG obtained when the robot was walking on a sandy terrain oscillates around 12 Hz, increasing up to 13 Hz for the wooden terrain.

4. Conclusions

In this paper, we have presented what, to the best of our knowledge is, the first sCPG that incorporates a feedback in the loop to modify the locomotion frequencies when applied to any legged robot through an adaptation mechanism. The feedback was pro-

vided by a FSR (but any sensor might be used) to simulate the force exerted on each limb of a future robot. With the use of this sensor, we injected different stimuli to the sCPG. Firstly, we performed some tests to find the optimal (minimum) input value, where the feedback effect could be easily observed. Then, some tests were performed to observe the effect in the oscillation frequencies under different stimuli. Finally, we carried out some experiments where the values read from the sensor were alternating between maximum and minimum voltage peaks in order to study the robustness of the sCPG against sudden changes in the oscillation frequency. From these experiments, we conclude that our sCPG presents a robust behavior in the adaptability of the oscillation frequencies. These frequencies can be used further in the generation of locomotion gaits for legged robots with the advantage that they can be modified depending on the terrain conditions.

The implementation of the sCPG was firstly done by using the Brian 2 simulator and then, considering the same parameters, it

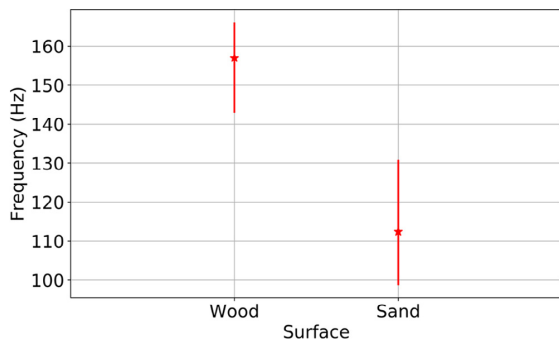


Fig. 19. Frequency generated by the *Pref* for two surfaces: wood and sand. The bars shows the mean, maximum and minimum frequency. The mean frequency for the wood surface is 157 Hz and for the sand terrain is 112.4 Hz.

was migrated to SpiNNaker. The results in both cases are highly correlated, as it is shown in Fig. 12; this fact demonstrates the reproducibility of our architecture on any platform.

Compared to the biological locomotion mechanism, we considered that our approach is highly plausible in two senses. The first one is that the proposed network model is based on spiking neurons, which are considered the neuron models that most closely mimic the behavior of biological neurons. The second aspect is that the implementation performed in SpiNNaker allowed us to improve it in terms of the power consumption, and the hardware by itself attempts to be an artificial representation of the brain.

Different experiments with real values from the output of the FSR sensors after attaching it to a leg have been performed. The proposed sCPG was stimulated with values obtained from the FSR when walking on two different terrains: sand and wood. The results confirm that the sCPG is able to adapt its oscillation rate depending on the feedback obtained from the surface that the robot is stepping onto.

As shown in Fig. 22, this network can be applied to any configuration of joints per leg, where one joint represents the vertical movement of the leg, and the other joint represents the horizontal movement. although a third joint is necessary in certain robotic models (e.g. hexapod models), it only acts as a support for the robot, so it would only be necessary to energize the motor to keep the joint in a fixed position.

As a future work, we propose to embed the SpiNNaker system into different legged robots (e.g. biped, quadruped and hexapod) to validate our approach on a real robotic platform by using the architecture shown in Fig. 22. In this architecture, the output of the Spiking Central Pattern Generator CPG_{AB} would drive four populations to execute the movement of each leg. Populations Forward Internal Motor (FIMX) and Reverse Internal Motor (RIMX) would control the clockwise and counterclockwise movement of the internal joint of the leg, respectively, while the populations FEMX and REMX would produce the clockwise and counterclockwise movement of the external joint. Furthermore, our proposal could

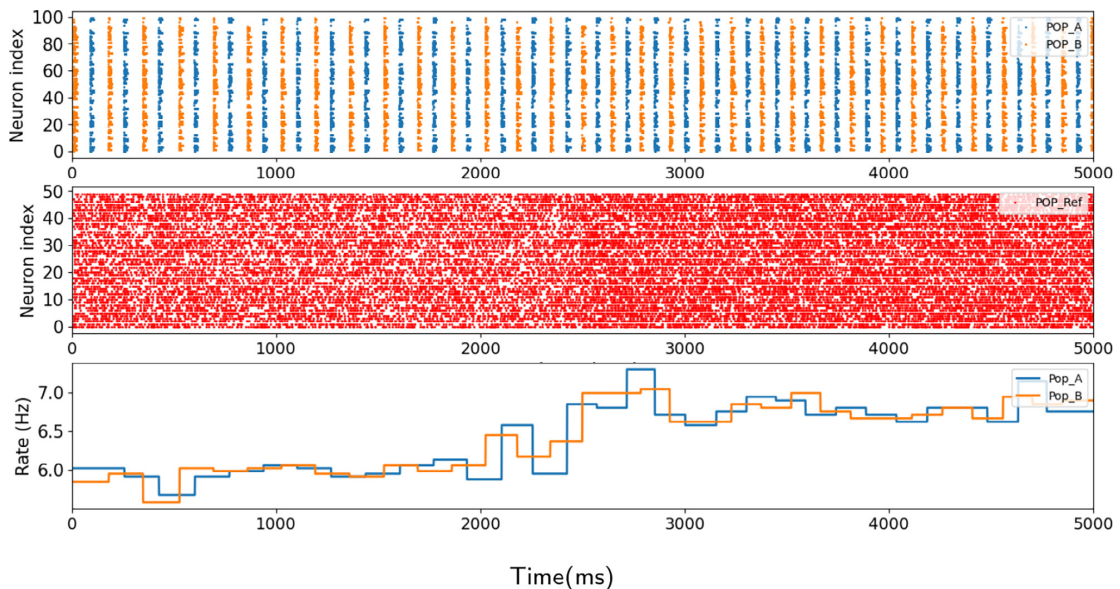


Fig. 20. Performance of the CPG simulated with Brian 2 along a move from a sandy terrain to a wooden one. The first 2500 ms corresponds to the robot walking on a sandy terrain, where the average frequency is 6 Hz, while in the last 2500 ms the frequency increases up to 7 Hz in a wooden terrain. The third row shows the firing rate of each population of the CPG in Hz. Thus, the global output rate of the CPG will be the combination of both.

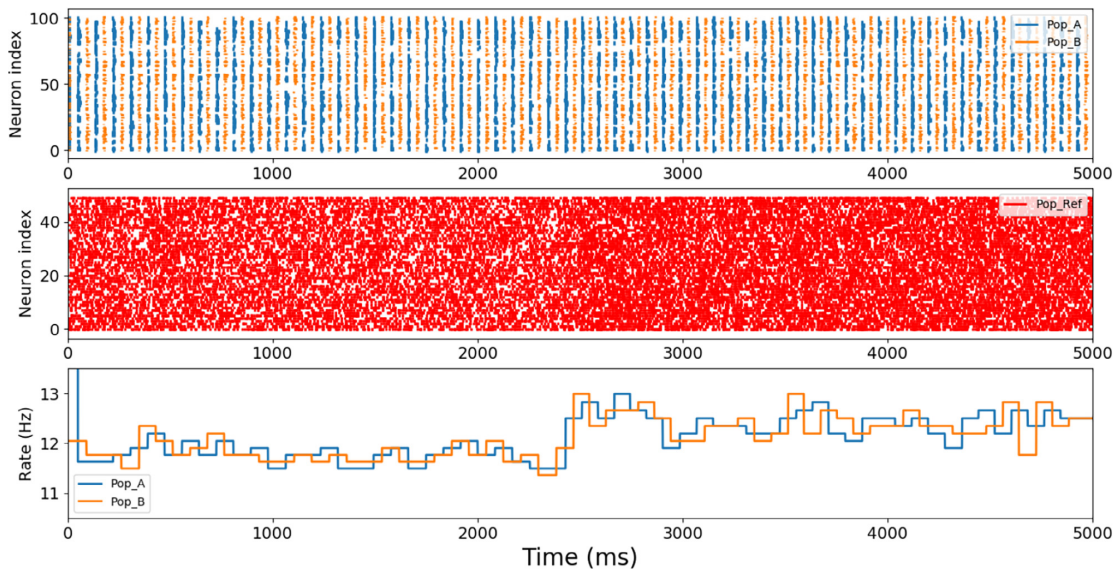


Fig. 21. Performance of the CPG implemented on SpiNNaker along a move from a sandy terrain to a wooden one. The first 2500 ms corresponds to the robot walking on a sandy terrain, where the average frequency is 12 Hz, while in the last 2500 ms the frequency increases up to 13 Hz in a wooden terrain, and the third row shows the firing rate of each population of the CPG in Hz. Thus, the global output rate of the CPG will be the combination of both.

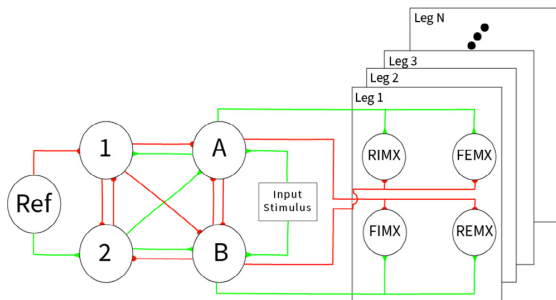


Fig. 22. Diagram of the proposed implementation architecture to drive a robotic platform with N legs. The red connections denote inhibitory connections and the green ones excitatory ones. The structure of populations that has to be replicated for each leg is composed of: FIMX, the RIMX, the FEMX and the REMX.

directly process spatio-temporal patterns with the same sCPG by incorporating a neuromorphic sensor.

CRedit authorship contribution statement

Pablo Lopez-Osorio: Methodology, Validation, Software, Conceptualization, Formal analysis, Investigation, Resources, Data curation, Writing – original draft, Writing – review & editing, Visualization. **Alberto Patiño-Saucedo:** Methodology, Validation, Software, Data curation, Writing – original draft, Writing – review & editing. **Juan P. Dominguez-Morales:** Project administration, Writing – original draft, Writing – review & editing, Resources, Conceptualization, Supervision. **Horacio Rostro-Gonzalez:** Funding acquisition, Project administration, Writing – original draft, Writing – review & editing, Resources, Supervision. **Fernando Perez-Peña:** Funding acquisition, Project administration, Writing

– original draft, Writing – review & editing, Resources, Conceptualization, Supervision.

Data availability

No data was used for the research described in the article.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was partially supported by the Interreg Atlantic Area Programme through the European Regional Development Fund (TIDE - Atlantic network for developing historical maritime tourism, EAPA_630/2018), by the Spanish grant (with support from the European Regional Development Fund) MIND-ROB (PID2019-105556 GB-C33) and by the EU H2020 project CHIST-ERA SMALL (PCI2019-111841-2).

References

- [1] A.J. Ijspeert, Central pattern generators for locomotion control in animals and robots: a review, *Neural networks* 21 (2008) 642–653.
- [2] J.L. Krichmar, H. Wagatsuma, *Neuromorphic and brain-based robots*, Cambridge University Press, 2011.
- [3] G. Indiveri, B. Linares-Barranco, T.J. Hamilton, A. Van Schaik, R. Etienne-Cummings, T. Delbruck, S.-C. Liu, P. Dudek, P. Häfliger, S. Renaud, et al., Neuromorphic silicon neuron circuits, *Front. Neurosci.* 5 (2011) 73.
- [4] R.J. Vogelstein, F.V. Tenore, L. Guevremont, R. Etienne-Cummings, V.K. Mushahwar, A silicon central pattern generator controls locomotion in vivo, *IEEE Trans. Biomed. Circuits Syst.* 2 (2008) 212–222.
- [5] T. Kano, K. Sakai, K. Yasui, D. Owaki, A. Ishiguro, Decentralized control mechanism underlying interlimb coordination of millipedes, *Bioinspiration Biomimetics* 12 (2017) 036007.
- [6] D. Owaki, T. Kano, K. Nagasawa, A. Tero, A. Ishiguro, Simple robot suggests physical interlimb communication is essential for quadruped walking, *J. R. Soc. Interface* 10 (2013) 20120669.

- [7] E. Donati, F. Corradi, C. Stefanini, G. Indiveri, A spiking implementation of the lamprey's Central Pattern Generator in neuromorphic VLSI, in: IEEE 2014 Biomedical Circuits and Systems Conference, BioCAS 2014 – Proceedings, pp. 512–515.
- [8] H. Rostro-Gonzalez, P.A. Cerna-Garcia, G. Trejo-Caballero, C.H. Garcia-Capulin, M.A. Ibarra-Manzano, J.G. Avina-Cervantes, C. Torres-Huitzil, A CPG system based on spiking neurons for hexapod robot locomotion, *Neurocomputing* 170 (2015) 47–54.
- [9] B. Cuevas-Arteaga, J.P. Dominguez-Morales, H. Rostro-Gonzalez, A. Espinal, A.F. Jimenez-Fernandez, F. Gomez-Rodriguez, A. Linares-Barranco, A SpiNNaker application: design, implementation and validation of SCPGs, in: International Work-Conference on Artificial Neural Networks, Springer, pp. 548–559.
- [10] S.B. Furber, F. Galluppi, S. Temple, L.A. Plana, The spinnaker project, *Proc. IEEE* 102 (2014) 652–665.
- [11] D. Gutierrez-Galan, J.P. Dominguez-Morales, F. Perez-Peña, A. Jimenez-Fernandez, A. Linares-Barranco, NeuroPod: a real-time neuromorphic spiking CPG applied to robotics, *Neurocomputing* 381 (2020) 10–19.
- [12] I. Polykretis, G. Tang, K.P. Michmizos, An astrocyte-modulated neuromorphic central pattern generator for hexapod robot locomotion on intel's loihi, *International Conference on Neuromorphic Systems* (2020) 1–9.
- [13] M. Davies, N. Srinivasa, T.-H. Lin, G. Chinya, Y. Cao, S.H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain, et al., Loihi: A neuromorphic manycore processor with on-chip learning, *IEEE Micro* 38 (2018) 82–99.
- [14] B. Strohmer, P. Manoonpong, L.B. Larsen, Flexible spiking cpGs for online manipulation during hexapod walking, *Front. Neurobot.* 14 (2020) 41.
- [15] G. Sartoretti, S. Shaw, K. Lam, N. Fan, M. Travers, H. Choset, Central pattern generator with inertial feedback for stable locomotion and climbing in unstructured terrain, in: 2018 IEEE International Conference on Robotics and Automation (ICRA), IEEE, pp. 1–5.
- [16] A. Spaeth, M. Tebyani, D. Haussler, M. Teodoroescu, Neuromorphic closed-loop control of a flexible modular robot by a simulated spiking central pattern generator, in: 2020 3rd IEEE International Conference on Soft Robotics (RoboSoft), IEEE, pp. 46–51.
- [17] E.M. Izhikevich, redWhich model to use for cortical spiking neurons?, *IEEE Trans Neural Networks* 15 (2004) 1063–1070.
- [18] D. Gutierrez-Galan, J.P. Dominguez-Morales, F. Perez-Peña, A. Jimenez-Fernandez, A. Linares-Barranco, Live Demonstration: Neuromorphic Robotics, from Audio to Locomotion Through Spiking CPG on SpiNNaker, in: 2019 IEEE International Symposium on Circuits and Systems (ISCAS), IEEE, pp. 1–1.
- [19] B. Strohmer, R.K. Stagsted, P. Manoonpong, L.B. Larsen, Integrating non-spiking interneurons in spiking neural networks, *Front. Neurosci.* 15 (2021) 184.
- [20] R.M. Harris-Warrick, Neuromodulation and flexibility in central pattern generator networks, *Curr. Opin. Neurobiol.* 21 (2011) 685–692.
- [21] A.S. Lele, Y. Fang, J. Ting, A. Raychowdhury, Learning to walk: Spike based reinforcement learning for hexapod robot central pattern generation, in: 2020 2nd IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS), IEEE, pp. 208–212.
- [22] C. Li, R. Lowe, T. Ziemke, Humanoids learning to walk: a natural CPG-actor-critic architecture, *Front. Neurobot.* 7 (2013) 5.
- [23] J. Ting, Y. Fang, A.S. Lele, A. Raychowdhury, Bio-inspired gait imitation of hexapod robot using event-based vision sensor and spiking neural network, arXiv preprint arXiv:2004.05450 (2020).
- [24] M. Stimberg, R. Brette, D.F. Goodman, Brian 2, an intuitive and efficient neural simulator, *Elife* 8 (2019) e47314.
- [25] N.T. Carnevale, M.L. Hines, *The NEURON book*, Cambridge University Press, 2006.
- [26] A.P. Davison, D. Brüderle, J.M. Eppler, J. Kremkow, E. Müller, D. Pecevski, L. Perrinet, P. Yger, PyNN: a common interface for neuronal network simulators, *Front. Neuroinformatics* 2 (2009) 11.
- [27] S.B. Furber, D.R. Lester, L.A. Plana, J.D. Garside, E. Painkras, S. Temple, A.D. Brown, Overview of the spinnaker system architecture, *IEEE Trans. Comput.* 62 (2012) 2454–2467.
- [28] S. Furber, P. Bogdan, SpiNNaker: A Spiking Neural Network Architecture, Boston-Delft: now publishers (2020).
- [29] H. Markram, The human brain project, *Sci. Am.* 306 (2012) 50–55.
- [30] E. Painkras, L.A. Plana, J. Garside, S. Temple, F. Galluppi, C. Patterson, D.R. Lester, A.D. Brown, S.B. Furber, SpiNNaker: A 1-W 18-core system-on-chip for massively-parallel neural network simulation, *IEEE J. Solid-State Circuits* 48 (2013) 1943–1953.
- [31] L.A. Plana, S.B. Furber, S. Temple, M. Khan, Y. Shi, J. Wu, S. Yang, A GALS infrastructure for a massively parallel multiprocessor, *IEEE Design Test Comput.* 24 (2007).
- [32] M. Mahowald, VLSI analogs of neuronal visual processing: a synthesis of form and function, Ph.D. thesis, California Institute of Technology Pasadena, 1992.
- [33] L.A. Plana, J. Garside, J. Heathcote, J. Pepper, S. Temple, S. Davidson, M. Luján, S. Furber, spiNNlink: FPGA-Based Interconnect for the Million-Core SpiNNaker System, *IEEE Access* 8 (2020) 84918–84928.
- [34] A. Yousefzadeh, M. Jabłoński, T. Iakymchuk, A. Linares-Barranco, A. Rosado, L.A. Plana, S. Temple, T. Serrano-Gotarredona, S.B. Furber, B. Linares-Barranco, On multiple AER handshaking channels over high-speed bit-serial bidirectional LVDS links with flow-control and clock-correction on commercial FPGAs for scalable neuromorphic systems, *IEEE Trans. Biomed. Circuits Syst.* 11 (2017) 1133–1147.
- [35] J.P. Dominguez-Morales, A. Jimenez-Fernandez, A. Rios-Navarro, E. Cerezuela-Escudero, D. Gutierrez-Galan, M.J. Dominguez-Morales, G. Jimenez-Moreno, Multilayer spiking neural network for audio samples classification using spinnaker, in: International conference on artificial neural networks, Springer, pp. 45–53.
- [36] T. Schoepe, D. Gutierrez-Galan, J.P. Dominguez-Morales, A. Jimenez-Fernandez, A. Linares-Barranco, E. Chicca, Neuromorphic sensory integration for combining sound source localization and collision avoidance, in: 2019 IEEE Biomedical Circuits and Systems Conference (BioCAS), IEEE, pp. 1–4.
- [37] O. Rhodes, P.A. Bogdan, C. Breninkmeijer, S. Davidson, D. Fellows, A. Gait, D.R. Lester, M. Mikaitis, L.A. Plana, A.G. Rowley, et al., sPyNNaker: a software package for running PyNN simulations on SpiNNaker, *Front. Neurosci.* 12 (2018) 816.
- [38] W. Gerstner, W. Kistler, *Spiking Neuron Models. Single Neurons, Populations, Plasticity*, Cambridge University Press, 2002.
- [39] S.J. Tripathy, J. Savitskaya, S.D. Burton, N.N. Urban, R.C. Gerkin, *NeuroElectro: a window to the world's neuron electrophysiology data*, *Front. Neuroinformatics* 8 (2014) 40.
- [40] F.J. Pontes, G. Amorim, P.P. Balestrassi, A. Paiva, J.R. Ferreira, redDesign of experiments and focused grid search for neural network parameter optimization, *Neurocomputing* 186 (2016) 22–34.

Pablo Lopez-Osorio is a PhD student in Bioengineering, Automation and Robotics. He received the Engineering degree in Electronics from the University of Cádiz (Spain) and his M.S. degree (Research in Systems and Computation Engineering) from the University of Cadiz (Spain) in 2019 and 2020 respectively. He is an Associate researcher in the University of Cadiz since 2020. His research interests include neuromorphic engineering, spiking neural networks, motor control, telecommunications, neurorobotics and neuromorphic sensors.

Alberto Patiño-Saucedo is a PhD student in Electrical Engineering at the University of Guanajuato (Mexico). He received the Engineering degree in Electronics from the Industrial University of Santander (Colombia) and the M.E. degree in Electrical Engineering from the University of Guanajuato in 2015 and 2017 respectively. His research interests include neuromorphic engineering, spiking neural networks, computational neuroscience and machine learning.

Juan P. Dominguez-Morales received the B.S. degree in computer engineering, the M.S. degree in computer engineering and networks, and the Ph.D. degree in computer engineering (specializing in neuromorphic audio processing and spiking neural networks) from the University of Seville (Sevilla, Spain), in 2014, 2015 and 2018, respectively.

From October 2015 to December 2018, he was a PhD student in the Architecture and Technology of Computers Department of the University of Seville with a research grant from the Spanish Ministry of Education and Science. Since January 2019, he has been working as Assistant Professor in the same department. His research interests include neuromorphic engineering, spiking neural networks, neuromorphic sensors, audio processing, medical image analysis, convolutional neural networks and computer-aided diagnosis systems.

Horacio Rostro-Gonzalez received his B.Eng. in Electronics from the ITC in 2003 and a M.E. degree with honors in Electrical Engineering from the University of Guanajuato in 2006, both in Mexico and the D.Sc. in Computational Neuroscience from the University of Nice-Sophia Antipolis, France in 2011. From 2011 to 2012, he also was a research fellow at the University of Cyprus, where he mainly worked on the FP7 EU SCANDLE Project. Between 2017 and 2018 he was visiting professor within the NST research group at the Technical University of Munich. Since May 2012, he is a full professor within the Department of Electronics at the University of Guanajuato in Mexico. He has authored around 40 journals and conferences proceeding papers. His research interests include reconfigurable electronics (FPGAs), neuromorphic engineering, computational neuroscience, bio-inspired algorithms, robotics, parallel computing and signal processing.

Fernando Perez-Peña received the Engineering degree in Telecommunications from the University of Seville (Spain) and his Ph.D. degree (specialized in neuromorphic motor control) from the University of Cadiz (Cadiz, Spain) in 2009 and 2014 respectively. In 2015 he was a postdoc at CITEC (Bielefeld University, Germany). He is an Assistant Professor in the Architecture and Technology of Computers Department of the University of Cadiz since 2014. His research interests include neuromorphic engineering, CPG, motor control and neurorobotics.

B

A feedback sensor based on spiking neural networks for real-time robot adaption

A feedback sensor based on spiking neural networks for real-time robot adaption

Pablo López-Osorio,¹[0000-0002-6152-6562] Fernando
Pérez-Peña¹[0000-0003-3586-2930] and Juan P.
Domínguez-Morales²[0000-0002-5474-107X]

¹ School of Engineering, Universidad de Cádiz, Spain,
pablo.osorio@uca.es,

² Robotics and Tech. of Computers Lab, ETSII-EPS
Universidad de Sevilla, Spain

Abstract. In recent years, it has been observed how the locomotion systems of vertebrate animals serve as inspiration to enhance the performance of robotic systems. These animal systems are characterized by their ability to adapt to environmental changes detected by their biological sensors. With this model in mind, our objective is to replicate this adaptability in robotics through the use of a Central Pattern Generator (CPG). We present an advanced robotic system based on Spiking Neural Networks (SNNs), implemented in both Spinnaker and Field Programmable Gate Arrays (FPGAs). This system is capable of modifying its locomotion pattern based on information provided by Force Sensitive Resistors (FSRs). Our experiments demonstrate that this platform can adapt in real-time to the terrain in which it operates.

Keywords: Spiking Neural Network, Neuromorphic Hardware, SpiN-Naker, Central Pattern Generator

1 Introduction

The synthesis of rhythmic locomotion in robotics, inspired by biological CPGs embedded in the spinal cord, has paved the way for an interdisciplinary convergence of neuromorphic engineering and robotics [1].

The intrinsic ability of the CPG to generate rhythmic output in the absence of external sensory stimuli is replicated in robotic frameworks to foster locomotion through various implementations, such as coupled oscillators, Artificial Neural Networks (ANNs), or SNNs [2]. The latter are more biologically plausible, recognized for their temporal encoding of information, and closely mimic biological neural structures, offering an advantage of resilience against noise over ANNs [3]. The transition from merely borrowing the biological concept of CPGs to adopting the neuromorphic approach underscores the evolution of neuromorphic robotics. This fusion not only encompasses the expertise of both communities, neuromorphic engineers and robotics experts, but also indicates a new era of low-energy, resource-efficient robotic systems [4, 5].

The spirit of neuromorphic robotics transcends traditional paradigms of robotic locomotion by incorporating the biological essence of locomotion through spiking neurons and synaptic connections, thereby simplifying the algorithmic framework. The inspiration robotics takes from biology, facilitated by neuromorphic engineering, lies at the heart of advancing locomotion strategies in robotics, paving the way for a new spectrum of applications that span both the analog and digital domains.

As previously discussed, there is a growing community of researchers looking to implement SNNs for a multitude of applications that mimic biology. Given the various ways to implement a neuro-inspired CPG [6], we will provide a brief review of the research presented to date that, in one way or another, is related to our work.

First, there have been several research efforts that have used coupled oscillators to implement a CPG. In [7], the design of a CPG using coupled Vanadium Dioxide nano-oscillators is addressed. This system mimics biological locomotion by generating rhythmic patterns that enhance energy efficiency and adaptability for applications in micro-robots and exoskeletons for spinal cord injury patients. In another recent work, a CPG based on sixteen coupled Hopf oscillators is utilized, optimized through a reinforcement learning algorithm, enabling a robotic platform to perform natural movements in swimming patterns [8]. In [9], the design and analysis of a bio-inspired CPG is described, which uses a network of mutually coupled nonlinear oscillators to generate a rhythmic walking pattern for bipedal robots.

Finally, in [10], a CPG model consisting of a set of mutually coupled Van Der Pol oscillators is discussed, intended to generate rhythmic movement patterns for robots with multiple joints. Although the functionality of oscillators is well established, they present certain disadvantages when compared to SNNs, such as stability and controllability, the effects of external noise on the network, or non-linear and non-ideal effects. Although it is possible to reduce the impact of these drawbacks in the network, it leads to an increase in system complexity, and could also reduce efficiency in terms of size and computational resources.

In other studies, different configurations of CPGs with coupled oscillators have been presented, such as in [11], to generate movement patterns in quadruped robots; however, these patterns could not be dynamically altered. Works such as [12] and [5] have employed arrays of CPGs to create different movement sequences in hexapod robots, but they lacked the capability for autonomous modification, as also reported in [11].

Some previous studies have implemented a Spiking Central Pattern Generator (sCPG) similar to the one proposed in our work but integrated an open-loop control network that bypassed the use of sensory inputs for real-time adjustments. For instance, [5, 12] detail three sCPG designs using the neuromorphic platform SpiNNaker [13] and an FPGA to manage the signals from the neuromorphic chip. The authors in [14] suggest another sCPG architecture that adjusts amplitude, phase, and frequency without the need for sensory information. These previous studies indicate that sCPGs offer significant energy savings

compared to traditional actuation and control methods. Furthermore, they highlight the interest in integrating external sensory feedback to modify the behavior of the neural network and, consequently, of the robotic platform.

Finally, [15] introduce the use of a Neuromorphic Auditory Sensor (NAS) as input to the sCPG in SpiNNaker to select movement patterns based on audio signals processed by an FPGA and interpreted by the SNN. In [16], a closed-loop system is presented that includes a Time Difference Encoder (TDE) integrated with SpiNNaker and an FPGA for processing intermediate signals. This system processes inputs from a NAS and a Sound Source Localization (SSL) to adjust the behavior of a robotic arm. Our work follows a similar path by using FSRs to regulate the movement pattern of the robotic platform.

The primary goal of this research is to develop and apply a SNN that consists of a sCPG which adjusts its operation based on external variables in the environment of a hexapod robot. The FSR modulate the activity of the sCPG according to the stiffness of the terrain, resulting in a variation in the speed of the hexapod, either increasing or decreasing it. Moreover, the SNN proposed in this study is capable of adapting to input stimuli and modifying the behavior of the neural network solely with the external sensory information provided. Although FSR are used, the adaptation process presented in this work can be applied with any type of sensor to integrate the external stimulus.

2 Materials

This section describes both the software and hardware used to perform the experiments proposed in this work.

2.1 Spiking neural network architecture (SpiNNaker)

The SpiNNaker project [17], an acronym for "Spiking Neural Network Architecture," is a research initiative and hardware framework devised for the emulation of large-scale biological neural networks. Originating from the University of Manchester in the United Kingdom and embedded within the Human Brain Project (HBP), it is predominantly engaged in the domains of computational neuroscience and biologically-inspired artificial intelligence.

SpiNNaker's conception leverages a specialized hardware configuration to mimic the operational dynamics of neural networks within the human brain. By opting for a multitude of efficient, cost-effective processors arranged in a particular configuration, it forgoes traditional CPU-based models [18]. These processors are designed to execute neuronal-level computations and are interconnected through a network characterized by low latency and high throughput, facilitating an asynchronous communication infrastructure adept at transmitting succinct packets indicative of individual neuronal spikes.

The Advanced Processor Technologies group at the School of Computer Science, University of Manchester, has recently assembled a million-core machine, which is accessible via the HBP Portal. In this work, a local SpiNN-3 machine was used for the instantiation of the robotic platform.

2.2 Robotic platform

A hexapod, designed and manufactured using 3D printing, has been developed to integrate all the necessary electronics on its surface. This device features six legs, each equipped with three Dynamixel AX-12A motors [19]. These motors communicate with the FPGA through an 8-bit half-duplex serial communication, with 1 Stop bit and no parity. For data transmission, the Dynamixel 1.0 communication protocol is used [20]. The 3D model of the robotic platform is presented in Figure 1.

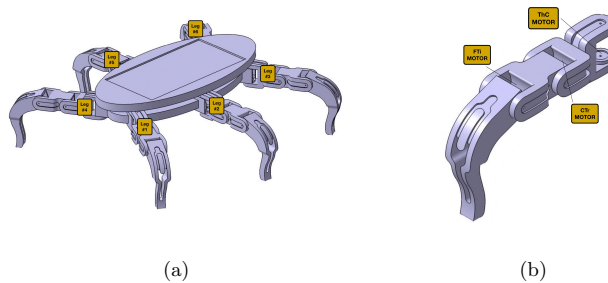


Fig. 1: A) 3D model of the complete robotic platform. B) 3D model of a single leg.

2.3 FPGA

For the interface between sensors and the neuromorphic platform, a Nexys 4 DDR development board featuring an FPGA Xilinx Artix-7 XC7A100T was used [21]. This board offers a programmable logic composed of 100,000 cells and includes 240 Digital Signal Processors (DSPs) slices, providing sufficient computational capability for the proposed task.

2.4 Force Sensitive Resistors (FSR)

To modify the walking speed of the robotic platform, analog FSRs have been used. These sensors will generate a voltage output based on the pressure exerted on them. Since the weight of the robotic platform remains stable and is distributed equally across each leg, the only variable that will modify the value provided by these sensors will be the terrain on which they are placed.

3 Methodology

3.1 Digital system

As mentioned in Section 2.3, a digital system is required to process the value obtained from the FSRs and translate this digital information into spikes. These spikes modify their transmission frequency through Pulse Frequency Modulation (PFM) and are subsequently sent through SpiNNLinks [22] to SpiNNaker. Once processed by the neural network, it will send the spikes produced by the sCPG back to the FPGA, with a frequency directly related to the pressure exerted by the surface on the robotic platform. Finally, the FPGA will modify the movement speed of the robotic platform based on the oscillation frequency of the sCPG.

To carry out this task, the previously mentioned FPGA will be utilized, which will implement an Hardware Description Language (HDL) code. This allows defining any digital circuit model by describing the connections of its components. Figure 2 shows a general and simplified diagram of the top module implemented in the FPGA. The process to be conducted is as follows:

Initially, the '*A/D Converter*' module converts the analog signal (0-5 V) into an 8-bit packet (0-255) with the instantaneous value emitted by the FSR. Subsequently, the '*PFM Generator*' module generates a variable clock signal (FSR Clock) that oscillates between 10 Hz (for a value of 0 V) and 171 Hz (for 5 V). Finally, the '*Address Event Representation (AER) in*' module will generate continuous spikes at variable frequencies, depending on the frequency of the '*FSR Clock*' signal, and these will be transmitted via SpiNNLinks [22] to SpiNNaker.

Once the incoming spikes have been processed by the neural network (see Section 3.3), the inverse process is carried out. The '*AER out*' module receives the spikes from the sCPG and processes them. This module then sends the '*Req_data*' signal to the '*PFM Receiver*' module, which links the '*req*' and '*ack*' signals of the SpiNNaker-FPGA communication. '*Req_data*' will only activate its rising edge when a '*req*' signal has been received and an '*ack*' signal has been sent, ensuring that a spike has been correctly received. Thanks to this behavior, it is possible to externally monitor the frequency of the sCPG outside of SpiNNaker.

To complete the process, the '*Locomotion Control Module*' and '*Half-Duplex Arbiter*' modules are responsible for generating the entire communication protocol between the actuators and the FPGA. They increase or decrease the speed of the actuators following the entire process that has just been explained.

3.2 SpiNNaker platform

In SpiNNaker, a total of 450 neurons are implemented, distributed as shown in table 1. The number of neurons per population and their parameters have been set prioritizing the reduction of the total number of neurons in the network: using fewer neurons per population causes erratic network performance, such as loss of rhythmicity or network outputs that do not correspond to the expected result.

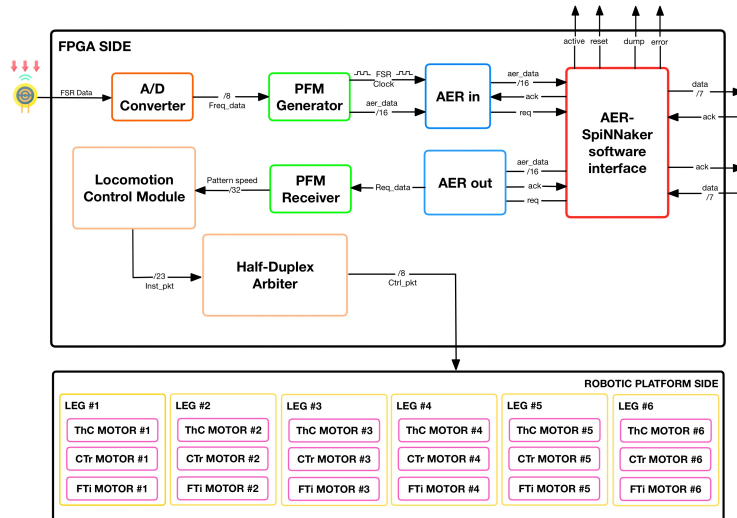


Fig. 2: Robotic platform FPGA top module overview.

Population	A	B	Exc	Inh	Ref
N ^o of neurons	100	100	100	100	50

Table 1: Number of neurons per population in the complete network.

3.3 Neural Network

The CPG is the main component of the neural network, composed of two populations of neurons, named A and B, each with 100 neurons. These form the CPG_{AB} , as illustrated in Figure 3. The number of neurons per population and their parameters have been optimized to reduce the total number in the network. Using fewer neurons per population may result in malfunctioning of the network, such as loss of rhythm or unexpected results.

Within these populations, there is self-excitation and self-inhibition with probabilities of 25% and 75%, and synaptic weights of 5 nA and 1.5 nA, respectively. This probabilistic configuration is due to the fact that populations receive more excitatory synapses from the FPGA-SpiNNaker input populations than internal inhibitory connections. The excitatory synaptic weights are significantly higher than the inhibitory ones, maintaining a balance between the probability of connection and its strength.

The probability of inhibition between one population and another is also 75%, with a weight of 0.25 nA. This setup ensures that when one population is active, the other is inhibited, generating oscillation in the network. These populations are designed to function as tonic neurons, receiving a low but constant external current to maintain regular activity. Therefore, in these two populations, the neuronal model would be as follows:

$$\frac{dV}{dt} = \frac{V_r - V + R(I_{exc} - I_{inh} + I_{st})}{\tau_m} \quad (1)$$

In this context, I_{st} represents the current injected into the neurons in populations A and B . This value has been set at 0.1 nA, which is sufficient to maintain the desired minimum oscillation in these populations. The specific neuronal parameters dictating the behavior of these populations are detailed in Table 2:

Parameter	C_m	τ_m	τ_r	u_{reset}	u_{rest}	u_{th}	τ_{syn_e}	τ_{syn_i}	I_{bias}
Value	0.25	12.5	0.001	-70.0	-65.0	-50.0	7.0	17.1	0.1
Unit	nF	ms	ms	mV	mV	mV	ms	ms	mA

Table 2: Neuron parameters for the proposed CPG in the SpiNNaker hardware platform.

By connecting the CPG_{AB} with the adaptation network through excitatory and inhibitory synaptic connections, it is possible to modify the activity of the CPG based on the input received from the FSRs. The complete network is shown in Figure 3, and its operation can be summarized in two possible cases:

- In the first case, if the activity of POP_{REF} is greater than that of the CPG, POP_{Exc} will receive strong excitation from POP_{REF} and weak inhibition from POP_A . This will increase the activity of POP_{Exc} , which in turn will increase the activity of the CPG. Additionally, POP_{Exc} will significantly inhibit POP_{Inh} . By doing this, it prevents POP_{Inh} from inhibiting the CPG. This mechanism ensures that when the activity of POP_{REF} is high, the CPG receives signals that enhance its activity, increasing the overall response of the network.
- In the second scenario, when the activity of POP_{REF} is lower than that of the CPG, an inverse process to the first case occurs. Here, POP_{Inh} will be strongly excited by the CPG and will receive weak inhibition from POP_{REF} . This will result in an increase in the activity of POP_{Inh} , leading to a decrease in the activity of the CPG. Moreover, in this situation, POP_{Inh} will exert strong inhibition on POP_{Exc} , preventing the latter from exciting the CPG. This control mechanism ensures that when the activity of POP_{REF} is low, the network responds by decreasing the activity of the CPG, demonstrating a dynamic adaptation of the network to changes in the activity levels of its components.

To optimize the weights of this neural network, a weight search algorithm based on the grid search technique was used. This methodology was chosen to ensure that the CPG maintains stable oscillations across the entire range of inputs from the FSRs. The input from these sensors varies between 10 Hz and 171 Hz, corresponding to the minimum and maximum pressure values detected, respectively.

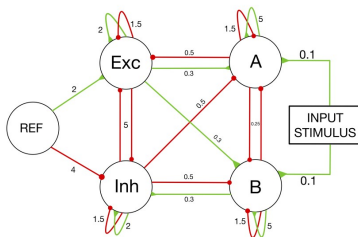


Fig. 3: FSR adaptive network topology.

The connection of CPG_{AB} with the adaptation network through excitatory and inhibitory synapses allows modifying the activity of CPG_{AB} based on the signals received from the FSRs. The complete structure of this network is represented in Figure 3.

4 Results

Initially, Figure 4 illustrates how the neural network responds to the spikes associated with the FSR. This figure highlights the variations in the network's response during activities such as running, trotting, and walking.

Another test conducted is presented in Figure 5, which shows the response of the network under increasing pressure applied to the FSR. The goal is to demonstrate minimal instability when transitioning between different walking patterns.

Figure 5 shows that a variation in the pressure of the FSR results in an immediate change in the oscillation frequency of the sCPG. Although delays may occur in the hardware logic, since the robotic platform does not directly use the output of the sCPG, but rather quantifies the number of spikes received from the neural network over a defined period of time (which can be adjusted for greater precision), a smooth transition is achieved between the three movement patterns without the need for the hexapod robot to pause its movement. The delay introduced in any of the patterns is approximately 2 ms.

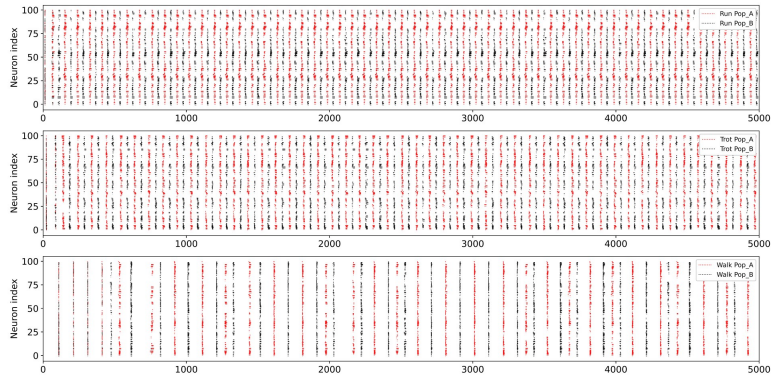


Fig. 4: Response of the CPG to incoming stimuli for generating different movement patterns.

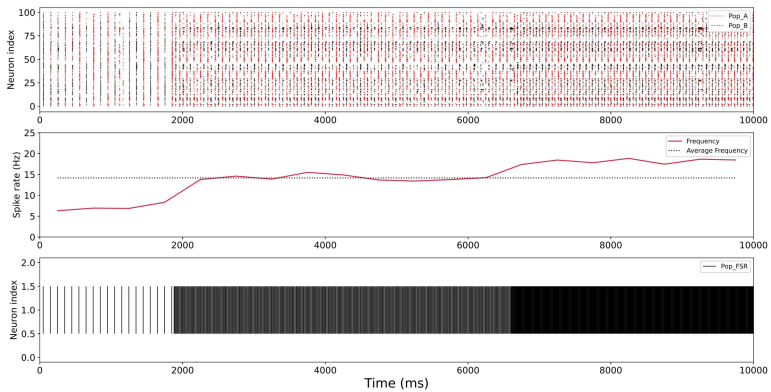


Fig. 5: Output of the sCPG and its frequency values (upper and middle graphs, respectively) obtained when subjecting the neural network to an increasing input from the FSR (lower graph).

5 Discussion and conclusions

In this study, a sCPG implemented in neuromorphic hardware has been introduced, capable of autonomously adjusting its locomotion pattern based on the terrain it encounters. As mentioned in the introduction, this system has been specifically designed to generate locomotion in a hexapod robot. In particular, on the robotic platform, post-processing is performed using PFM of the spikes

generated by the sCPG. This eliminates the need to introduce artificial delays in neuronal synapses or modify the time scaling factor to adjust the network's output to appropriate levels for actuator control.

In contrast, previous works, such as the one presented in [5], required a significant increase in the `time_scale_factor` parameter of up to 100 in order to use the output of the neural network as an activation signal for the actuators. Furthermore, in studies such as [23], a convergence time of up to 5 seconds was observed, or delays in the propagation of movement of up to 50 ms, as mentioned in [24].

In future work, it is suggested to reduce the number of neurons in each population, which would result in a significant decrease in resource consumption, both in terms of energy and computation. Additionally, it would be very interesting to explore the implementation of other neuromorphic systems, such as the use of Dynamic Vision Sensor (DVS) or NAS, to adjust other parameters such as the orientation of the platform. This would enhance the adaptability of the robotic platform by incorporating pattern or sound recognition capabilities.

Acknowledgements

This research was partially supported by project PID2019-105556GB-C33 funded by MCIN/ AEI /10.13039/501100011033.

References

1. S. Grillner and P. Wallen, "Central pattern generators for locomotion, with special reference to vertebrates," *Annual review of neuroscience*, vol. 8, no. 1, pp. 233–261, 1985.
2. S. L. Hooper, "Central pattern generators," *Current Biology*, vol. 10, no. 5, pp. R176–R179, 2000.
3. Y. Kim, Y. Li, H. Park, Y. Venkatesha, A. Hambitzer, and P. Panda, "Exploring temporal information dynamics in spiking neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 7, 2023, pp. 8308–8316.
4. A. Linares-Barranco, F. Perez-Peña, A. Jimenez-Fernandez, and E. Chicca, "Ed-biorob: a neuromorphic robotic arm with fpga-based infrastructure for bio-inspired spiking motor controllers," *Frontiers in neurobotics*, vol. 14, p. 590163, 2020.
5. D. Gutierrez-Galan, J. P. Dominguez-Morales, F. Perez-Peña, A. Jimenez-Fernandez, and A. Linares-Barranco, "Neuropod: a real-time neuromorphic spiking cpg applied to robotics," *Neurocomputing*, vol. 381, pp. 10–19, 2020.
6. A. J. Ijspeert, "Central pattern generators for locomotion control in animals and robots: a review," *Neural networks*, vol. 21, no. 4, pp. 642–653, 2008.
7. S. Dutta, A. Parihar, A. Khanna, J. Gomez, W. Chakraborty, M. Jerry, B. Grisafe, A. Raychowdhury, and S. Datta, "Programmable coupled oscillators for synchronized locomotion," *Nature communications*, vol. 10, no. 1, p. 3299, 2019.
8. V. Nguyen, D. Q. Vo, V. Duong, H. H. Nguyen, and T. T. Nguyen, "Reinforcement learning-based optimization of locomotion controller using multiple coupled cpg oscillators for elongated undulating fin propulsion." *Mathematical Biosciences and Engineering: MBE*, vol. 19, no. 1, pp. 738–758, 2021.

9. M. K. Habib, G. L. Liu, K. Watanabe, and K. Izumi, "Bipedal locomotion control via cpgs with coupled nonlinear oscillators," in *2007 IEEE International Conference on Mechatronics*. IEEE, 2007, pp. 1–6.
10. C. Liu, Q. Chen, and J. Zhang, "Coupled van der pol oscillators utilised as central pattern generators for quadruped locomotion," in *2009 Chinese control and decision conference*. IEEE, 2009, pp. 3677–3682.
11. L. Righetti and A. J. Ijspeert, "Pattern generators with sensory feedback for the control of quadruped locomotion," in *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 819–824.
12. H. Rostro-Gonzalez, P. A. Cerna-Garcia, G. Trejo-Caballero, C. H. Garcia-Capulin, M. A. Ibarra-Manzano, J. G. Avina-Cervantes, and C. Torres-Huitzil, "A cpg system based on spiking neurons for hexapod robot locomotion," *Neurocomputing*, vol. 170, pp. 47–54, 2015.
13. S. B. Furber, F. Galluppi, S. Temple, and L. A. Plana, "The spinnaker project," *Proceedings of the IEEE*, vol. 102, no. 5, pp. 652–665, 2014.
14. B. Strohmmer, P. Manoonpong, and L. B. Larsen, "Flexible spiking cpgs for online manipulation during hexapod walking," *Frontiers in neurobotics*, vol. 14, p. 41, 2020.
15. D. Gutierrez-Galan, J. P. Dominguez-Morales, F. Perez-Pena, A. Jimenez-Fernandez, and A. Linares-Barranco, "Live demonstration: neuromorphic robotics, from audio to locomotion through spiking cpg on spinnaker," in *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2019, pp. 1–1.
16. T. Schoepe, D. Gutierrez-Galan, J. P. Dominguez-Morales, H. Grotorex, A. F. Jiménez Fernández, A. Linares-Barranco, and E. Chicca, "Closed-loop sound source localization in neuromorphic systems," *Neuromorphic Computing and Engineering*, 2023.
17. S. B. Furber, D. R. Lester, L. A. Plana, J. D. Garside, E. Painkras, S. Temple, and A. D. Brown, "Overview of the spinnaker system architecture," *IEEE Transactions on Computers*, vol. 62, no. 12, pp. 2454–2467, 2012.
18. E. Painkras, L. A. Plana, J. Garside, S. Temple, F. Galluppi, C. Patterson, D. R. Lester, A. D. Brown, and S. B. Furber, "SpiNNaker: A 1-W 18-core system-on-chip for massively-parallel neural network simulation," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 8, pp. 1943–1953, 2013.
19. "Ax-12a - robotis e-manual," <https://emanual.robotis.com/docs/en/dxl/ax/ax-12a/>, 2023, accessed on August 2, 2023.
20. ROBOTIS, "Dynamixel protocol 1.0," <https://emanual.robotis.com/docs/en/dxl/protocol1/>, 2023, accedido el: 14 de noviembre de 2023.
21. Digilent, "Nexys 4 ddr reference manual," <https://digilent.com/reference/programmable-logic/nexys-4-ddr/reference-manual>, visitado el 8 de septiembre de 2023.
22. L. A. Plana, J. Garside, J. Heathcote, J. Pepper, S. Temple, S. Davidson, M. Luján, and S. Furber, "spiNNlink: FPGA-Based Interconnect for the Million-Core SpiN-Naker System," *IEEE Access*, vol. 8, pp. 84918–84928, 2020.
23. A. Crespi and A. J. Ijspeert, "Amphibot ii: An amphibious snake robot that crawls and swims using a central pattern generator," in *Proceedings of the 9th international conference on climbing and walking robots (CLAWAR 2006)*, no. CONF, 2006, pp. 19–27.
24. E. Donati, F. Corradi, C. Stefanini, and G. Indiveri, "A spiking implementation of the lamprey's central pattern generator in neuromorphic vlsi," in *2014 IEEE Biomedical Circuits and Systems Conference (BioCAS) Proceedings*. IEEE, 2014, pp. 512–515.

C

A Neuromorphic Vision and Feedback Sensor Fusion Based on Spi- king Neural Networks for Real-Time Robot Adaption

A Neuromorphic Vision and Feedback Sensor Fusion Based on Spiking Neural Networks for Real-Time Robot Adaption

Pablo López-Osorio,* Juan Pedro Domínguez-Morales, and Fernando Perez-Peña

For some years now, the locomotion mechanisms used by vertebrate animals have been a major inspiration for the improvement of robotic systems. These mechanisms range from adapting their movements to move through the environment to the ability to chase prey, all thanks to senses such as sight, hearing, and touch. Neuromorphic engineering is inspired by brain problem-solving techniques with the goal of implementing models that take advantage of the characteristics of biological neural systems. While this is a well-defined and explored area in this field, there is no previous work that fuses analog and neuromorphic sensors to control and modify robotic behavior in real time. Herein, a system is presented based on spiking neural networks implemented on the SpiNNaker hardware platform that receives information from both analog (force-sensing resistor) and digital (neuromorphic retina) sensors and is able to adapt the speed and orientation of a hexapod robot depending on the stability of the terrain where it is located and the position of the target. These sensors are used to modify the behavior of different spiking central pattern generators, which in turn will adapt the speed and orientation of the robotic platform, all in real time. In particular, experiments show that the network is capable of correctly adapting to the stimuli received from the sensors, modifying the speed and heading of the robotic platform.

generating rhythmic patterns of neural activity even without requiring external sensory inputs.^[6,7] They are located in the spinal cord of vertebrates or in the nervous system of invertebrates and are responsible for controlling and coordinating repetitive movements and sequences of activity, such as walking in mammals,^[8] the rhythm of swimming in fish,^[9] the control of the heartbeat,^[10] or the basic rhythm of breathing.^[11] Usually, these neural structures are made of, at least, two neuronal populations that alternate their cycles of activity, producing oscillations in the output without the need to receive oscillating signals as stimulus.

The research field called neuromorphic engineering aims to implement these neural structures in electronic devices, emulating the way living beings have solved complex problems using biological circuits.^[12] Neuromorphic robotics combines both the neuromorphic engineering community and roboticists, leading to

potential applications that bring together expertise from both fields.^[13,14]

The robotics field borrows concepts from biological CPGs to create locomotion in robotic platforms. To implement these neural structures, there are several options available, including using coupled oscillators, artificial neural networks (ANNs), or spiking neural networks (SNNs). Among these options, SNNs stand out as the closest to biology in terms of performance and behavior. This is primarily due to its temporal information encoding, which enhances its resilience against noise when compared to ANNs.^[15] Furthermore, SNNs stand out in terms of biological synaptic plasticity and energy efficiency,^[16] as they only consume energy when neurons emit spikes (action potentials), closely resembling the functioning of the brain. Consequently, when applied in robotics, the use of SNNs results in reduced resource requirements, lower energy consumption, and simplified algorithms, in contrast to traditional approaches that rely on ANNs or coupled oscillators.


The use of SNNs for locomotion generation gave rise to spiking central pattern generator (sCPG). Using spiking neurons, a rhythmic output pattern can be generated, which can be used to generate motion on robotic platforms. In addition, one of the advantages of SNNs is that they can generate very stable patterns

1. Introduction

It has long been known that biological locomotion is produced by a neural structure called central pattern generator (CPG).^[1–5] This structure is made of neural circuits that are specialized in

P. López-Osorio, F. Perez-Peña
School of Engineering
Universidad de Cádiz
11519 Puerto Real, Spain
E-mail: pablo.osorio@uca.es

J. P. Domínguez-Morales
Robotics and Technology of Computers Lab.
ETSII-EPS
Universidad de Sevilla
Sevilla 41012, Spain

 The ORCID identification number(s) for the author(s) of this article can be found under <https://doi.org/10.1002/aisy.202300646>.

© 2024 The Authors. Advanced Intelligent Systems published by Wiley-VCH GmbH. This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

DOI: 10.1002/aisy.202300646

without the need to incorporate any sensory or brain information to regulate their operation or oscillation frequency.^[14,17,18]

Previous works that explore the implementation of CPGs as signal generators for robotic actuator control are: ref. [19] where two different topologies of CPGs with coupled oscillators are presented to provide different motion patterns to a quadruped robot; however, these patterns are not interchangeable on the fly; others like ref. [20] or ref. [14] use a set of CPGs to produce various motion patterns in hexapod robots; however, the pattern cannot be modified autonomously. The same occurs in ref. [19].

There are some previous works that use a sCPG similar to the one proposed in this article but they include an open-loop network that does not conceive the use of sensory inputs for real-time modification of the system. For example, in refs. [14,20,21], three sCPGs designs capable of implementing different gait patterns on robotic platforms are presented, using the SpiNNaker neuromorphic platform^[22] together with a field programmable gate array (FPGA) to process the signals received from the neuromorphic chip. In ref. [21], in addition, a dynamic vision sensor (DVS) is used that influences the selection of the hexapod's walking rhythm, but a constant connection to a computer is required. In ref. [23], the authors propose another sCPG architecture with the ability to modify its amplitude, phase, and frequency without requiring any sensory information. Other neuromorphic platforms such as Loihi^[24] have also been used to generate locomotion in a robotic lamprey that can select up to two different gait patterns but does not use a control loop over the generated^[25] motion. These previous works demonstrate that the use of sCPGs provides significant energy savings compared to conventional actuation and control methods. Furthermore, these works highlight the fact that it would be interesting to incorporate external sensory feedback to modify the behavior of the neural network and, therefore, the behavior of the robotic platform.

There are some previous works that include sensory feedback.^[26–28] In such cases, conventional oscillators have been used to model the behavior of the CPG, or Hopf oscillators, as in ref. [29] where it is used to generate the rhythmic pattern in the CPG. In another recent work,^[30] they used different inertial measurement unit sensors and distance sensors for a robotic fish to modify in real time a series of coupled oscillators, autonomously selecting the most optimal swimming routine depending on the surroundings.

In ref. [31], 12 spiking neurons modulated by sensory feedback are used to shape the sCPG and the modification of the motion pattern of the robotic structure is achieved by adding or subtracting the neural structures, whereas in our work, it is done by an adaptive mechanism of the SNN. Furthermore, the authors used the Izhikevich neural model,^[32] which involves a more complex computational model than the one proposed in this article: the leaky integrate-and-fire (LIF) model. Conversely, in ref. [33], a neuromorphic auditory sensor (NAS) is used as an input sensor to decide the pattern to be followed by the sCPG in SpiNNaker, depending on the audio processed by a FPGA and recognized by the SNN. In ref. [34], a closed-loop system is presented that incorporates a time difference encoder in SpiNNaker together with a FPGA to process the intermediate signals, which analyzes the information received from a NAS and a sound source localization system to modify the behavior of a robotic arm. In our work, we use a similar approach: a fusion

of the force-sensitive resistors (FSRs) and the DVS is used to modify the behavior of the neural network implemented in SpiNNaker in real time.

A recent work proposes the combination of firing and nonfiring neurons to simulate a neural network that regulates amplitude, phase, and frequency (each independently) in a closed-loop system.^[35] On the one hand, the nonfiring neurons used in Strohmer's study are LIF neurons, with a threshold high enough to avoid neuronal firing when the analog input varies. On the other hand, the firing neurons employ the adaptive exponential integrate-and-fire (AdEx) model, a considerably more complex neuronal model than that used in the work presented in this article, with a complex implementation in neuromorphic hardware. The primary focus of the study is the search for an optimal combination of spiking and nonspiking neurons to create a sensorimotor neuron network capable of adjusting the network's output (spiking) based on analog parameters at the network's input (nonspiking). However, locomotion is not the primary objective of this article, although it is briefly mentioned. As mentioned earlier, the authors propose compatibility of their network with neuromorphic hardware, although this compatibility is limited only to the nonspiking neurons and a cloud-based simulation service called CloudBrain.

The proposal made in this article differs in several aspects: compared to ref. [35], a different spiking neuron model is used and all neurons are firing neurons. The second aspect that sets us apart is the use of information obtained from an FSR and a DVS that allows us to modify the behavior of the neural network controlling the hexapod robot. This enables us to adjust its speed and direction based on terrain characteristics and activity detected within the field of view of the DVS. The third and most important aspect to emphasize is the validation of our model on neuromorphic hardware, specifically on the SpiNNaker platform, where we utilize high-level software tools that enhance the reproducibility of the network for the scientific community.

The main goal of this work is to design and implement a SNN composed of a sCPG and a Winner Take All (WTA) network that change their behavior depending on external variables in the environment of a hexapod robot. On the one hand, the FSR sensors will increase or decrease the activity of the sCPG depending on the stiffness of the terrain, resulting in a higher or lower speed of the hexapod. On the other hand, the DVS sensor will detect the predominant direction of motion, generating spikes associated with this direction and sending them to the WTA network. The WTA network will then select the winner and transmit the spikes to the FPGA to modify the behavior of the robotic platform, implementing a prey–predator model. Furthermore, the SNN network presented in this work is able to adapt to these input stimuli and modify the behavior of the neural network only with the external sensory information provided. Although FSR and DVS sensors are used, the adaptation step presented in this work can be used with any type of sensor to introduce the external stimulus.

The main contributions of this work are: 1) To the authors' knowledge, this work is the first to combine neuromorphic and analog sensors as input to the SpiNNaker platform for closed-loop control of a hexapod using sCPGs; 2) The experiments have not only been simulated but also implemented in hardware, in this case using an FPGA and a spinn-3 platform;

and 3) The implemented workflow is fully functional in real time and with closed-loop online terrain adaptation and pattern change.

The rest of the article is structured as follows: Section 2 is divided into two subsections: first, the materials (Section 2.1) used in this work are introduced, including the neuromorphic platform and the hardware used. The implemented methods are described in Section 2.2, along with the SNN models. The results obtained are presented in Section 3 and discussed in Section 4 and, finally, the conclusions of this work are briefly discussed in Section 5.

2. Experimental Section

2.1. Materials

This section describes both the software and hardware used to perform the experiments proposed in this work.

2.1.1. Spiking Neural Network Architecture (SpiNNaker)

SpiNNaker^[22,36,37] stands for “Spiking Neural Network Architecture” and is a research project and hardware platform designed to simulate large-scale biological neural networks. It was developed at the University of Manchester in the UK as part of the Human Brain Project (HBP)^[38] and is primarily used in the fields of computational neuroscience and biologically inspired artificial intelligence.

The idea behind SpiNNaker is to use specialized hardware design to simulate how neural networks work in the human brain. SpiNNaker models the communication between neurons in a neural network using a large number of low-cost and effective processors coupled in a certain way instead of relying on conventional central processing units.^[39] These processors can perform computations at the neuronal level and connect to each other in a low-latency, high-speed network. This configuration enables an asynchronous communication infrastructure for transmitting short packets, representing individual neuron firings,^[40] identified using address event representation (AER).^[41]

Various SpiNNaker machines were manufactured and made commercially available, including SpiNN-3 and SpiNN-5, which house 4 and 48 SpiNNaker chips, respectively. Additionally, they incorporate spiNNlinks,^[42] which facilitates real-time input/output interfacing with neuromorphic sensors and other neuromorphic platforms such as FPGAs.^[43–45] To design and implement SNNs on these machines, researchers can employ a PyNN-based software package^[46] named sPyNNaker.^[47]

Notably, the Advanced Processor Technologies group from the School of Computer Science at the University of Manchester recently constructed a million-core machine, accessible through the HBP portal. In this study, a local SpiNN-3 machine was utilized to implement the robotic platform.

2.1.2. FPGA

An FPGA Nexys 4 DDR^[48] development board was used for communication between the sensors and the neuromorphic

platform. This board is equipped with an FPGA Xilinx Artix-7 XC7A100T-1CSG324C, which offers a programmable logic capacity of 100 000 cells and digital signal processors slices, providing sufficient processing power for this work.

2.1.3. DVS

The AER DVS128 retina chip (silicon retina)^[49] is an array of autonomous photoreceptors (pixels) that respond to logarithmic changes in brightness intensity in real time by placing the address of that specific pixel on an asynchronous bus. Only pixels that are stimulated by a change in illumination generate spikes, saving storage and bandwidth over conventional vision sensors. The pixel addresses use of the AER protocol and contain the X and Y coordinates of the photoreceptor that detected the change in brightness.

In this work, a retina DVS sensor^[50,51] was used, containing a 128×128 pixels matrix, whose generated output is used as input to a WTA network to decide whether the robotic platform should continue forward or whether it should turn left or right, to target the direction that generates the highest spiking activity. The addresses associated with each pixel are composed of 7 bits associated with the Y coordinate, 7 bits associated with the X coordinate, and a polarity bit representing the change in contrast, where a logic 1 means an increase in brightness and a logic 0 means a decrease. This sensor is connected to a pAER interface that allows real-time visualization of the retinal output, while simultaneously sending the events to the FPGA platform through the CAVIAR connector.^[52]

2.1.4. FSRs

To modify the running speed of the robotic platform, different analog sensors have been used. These sensors will generate a voltage output depending on the pressure that is exerted on them. Since the weight of the robotic platform is stable and equally distributed on each leg, the only variable that will modify the value provided by these sensors will be the terrain on which they are located: If the robot is on firm, flat ground (such as concrete), the forces will be distributed evenly over the surface of the sensors, generating a higher pressure and therefore a higher voltage. If, in contrast, the platform is on more uneven ground (such as sand), the forces generated by the weight of the robot will not be evenly distributed over the entire surface of the sensors, thus generating less pressure on them and therefore a lower voltage.

2.1.5. Dynamixel AX-12A Actuators

Eighteen Dynamixel AX-12A motors will be used to move the robotic platform.^[53] These actuators have the ability to control their speed, temperature, position, voltage, and load supported. Each of these motors incorporates an ARM Cortex-M3 microcontroller that manages the internal logic of the motor, allows them to be easily controlled, and enables bidirectional communication via the Dynamixel network protocol. In addition, multiple motors can be daisy-chained on a single communication bus, allowing

multiple motors to be controlled with a single cable and a single interface.

2.1.6. Robotic Platform

A hexapod has been designed and manufactured using 3D printing, which allows all the necessary electronics to be housed on its surface. The robotic platform performs the movement thanks to the 18 Dynamixel motors that it incorporates, which are controlled by the FPGA.

2.1.7. Minimum Required Configuration

Figure 1 illustrates the minimum configuration required to accurately replicate the tests conducted in this study. This setup includes a SpiNN-3 board, a DVS, an FSR, an FPGA unit, and two Dynamixel AX-12A motors. One of the Dynamixel motors is used to regulate speed in response to signals from the FSR, while the other adjusts orientation based on the information provided by the DVS.

2.2. Methodology

2.2.1. Neuron Models

The neural model used to implement the neurons of the network is the LIF model. This model is defined according to Equation (1) and (2)

$$\frac{dV}{dt} = \frac{-(V - V_r) + RI(t)}{\tau_m} \quad (1)$$

$$\text{if } V(t) = V_{th} \text{ then } \lim_{\delta \rightarrow 0, \delta > 0} V(t + \delta) = V_r \quad (2)$$

where V is the membrane potential of the neuron, R represents the resistance presented by this membrane, τ_m the time constant

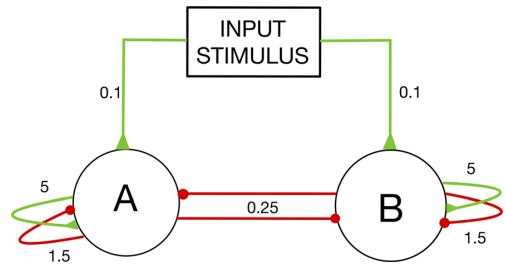


Figure 2. CPG_{AB} network topology.

of the neuron, V_r the resting potential, V_{th} the threshold required to trigger the firing of the neuron, and $I(t)$ is the stimulus (in the form of current) entering the neuron.

CPG Populations: The CPG is the main neural structure of the network and is composed of the populations A and B of 100 neurons each, which form the CPG_{AB}, represented in **Figure 2**. The number of neurons per population and their parameters have been set prioritizing the reduction of the total number of neurons in the network: using fewer neurons per population causes erratic network performance, such as loss of rhythmicity or network outputs that do not correspond to the expected result. These populations, on the one hand, are self-excited and self-inhibited with a probability of 25% and 75% and weights of 5 and 1.5 nA, respectively. These probabilistic parameters have been established since the number of excitatory synapses received by the adapting network populations from the FPGA-SpiNNaker input populations are more numerous than the inhibitory connections produced internally in the adapting network itself. Furthermore, the excitatory synaptic weight values are considerably higher than those of the inhibitory synaptic weights, maintaining a balance between the probability of connection and the strength with

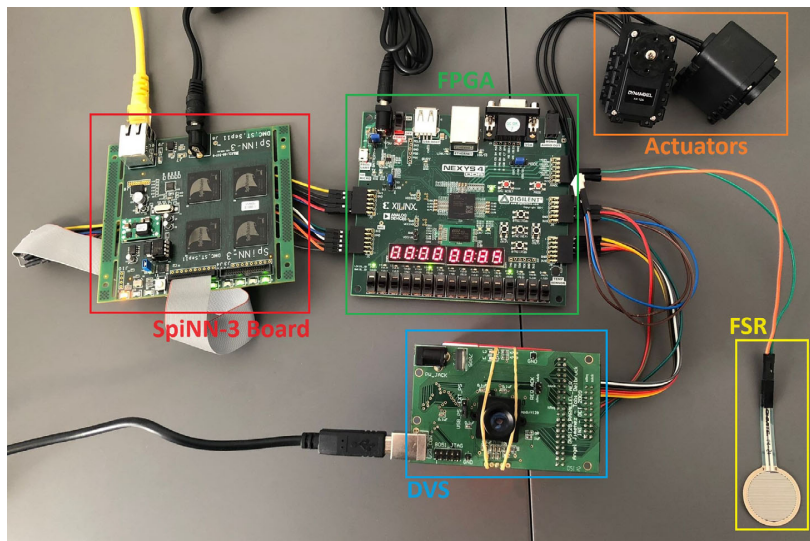


Figure 1. Minimum required configuration to replicate the experiment.

which the connection is established. The probability of inhibition between one population and the other is also 75%, with a weight of 0.25 nA. Due to this configuration, when one of the populations has activity, the other will be inhibited accordingly, causing the oscillation in the network. It is important to note that these populations are designed to work as tonic neurons, so they are injected with a very low but constant external current to ensure that they are constantly generating activity. Therefore, in these two populations the neuronal model would be as follows

$$\frac{dV}{dt} = \frac{V_r - V + R(I_{exc} - I_{inh} + I_{st})}{\tau_m} \quad (3)$$

where I_{st} is the current injected into neurons in populations A and B. This value is set to 0.1 nA, which is sufficient to maintain the desired minimum oscillation. The neuronal parameters governing the behavior of these populations are presented in Table 1.

Adaptive Populations: In this part of the network, we can distinguish three populations: POP_{REF} is composed of 50 neurons that directly receive the input stimuli linked to the FSR, and its neuronal parameters are those presented in Table 1. POP_E and POP_I are populations of 100 neurons each, and their weights are configured in such a way that, depending on the activity of POP_{REF}, the activity of these neurons increases or decreases. In turn, these populations have inhibitory connections between them, achieving an oscillatory phenomenon between these two populations similar to that of CPG_{AB}. These populations are based on the neuronal model LIF, but their neuronal parameters are adapted in such a way that the input–output ratio of these populations is 1:1, thus ensuring that their behavior is closely related to the input provided by the FSR. Their parameters are shown in Table 2, and their connection scheme is illustrated in Figure 3.

By connecting CPG_{AB} to the adaptive network via excitatory and inhibitory synaptic connections, it is possible to modify its activity depending on the input received from the FSR sensors. The complete network is shown in Figure 4 and its operation can be summarized in two possible cases: 1) If the activity of POP_{REF} is greater than the activity of CPG, POP_{Exc} will be strongly excited by POP_{REF} and weakly inhibited by POP_A, which will increase its activity, in turn causing an increase in the activity of CPG. Furthermore, POP_{Exc} will strongly inhibit POP_{Inh},

Table 1. Neuron parameters for the proposed CPG in SpiNNaker hardware platform.

Parameter	Value
c_m	0.25 nF
τ_m	12.5 ms
τ_r	0.001 ms
u_{reset}	-70.0 mV
u_{rest}	-65.0 mV
u_{th}	-50.0 mV
τ_{syn_e}	7.0 ms
τ_{syn_i}	17.1 ms
I_{bias}	0.1 mA

Table 2. Proposed neural parameters for the adaptive populations on the SpiNNaker hardware platform.

Parameter	Value
c_m	0.1 nF
τ_m	0.1 ms
τ_r	0.0 ms
u_{reset}	-65.0 mV
u_{rest}	-65.0 mV
u_{th}	-64.91 mV
τ_{syn_e}	0.1 ms
τ_{syn_i}	0.1 ms
I_{bias}	0.0 mA

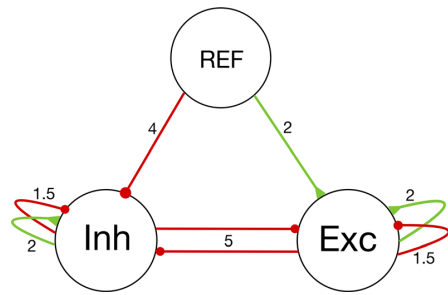


Figure 3. Topology of the CPG_{EI} network.

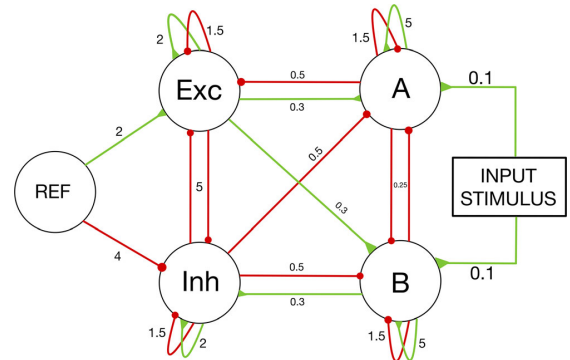


Figure 4. FSR adaptive network topology.

preventing the latter from inhibiting CPG; and 2) If the activity of POP_{REF} is lower than the activity of CPG, POP_{Inh} will be strongly excited by CPG and weakly inhibited by POP_{REF}, which will increase its activity, which in turn will cause a decrease in the activity of CPG. In addition, POP_{Inh} will strongly inhibit POP_{Exc}, preventing the latter from exciting CPG.

The weights of this entire network have been tuned using a grid search-based weight search algorithm, and are configured to ensure stable oscillations of the CPG over the entire input range of the FSRs. This input will oscillate between 10 and

171 Hz (for minimum and maximum pressure values, respectively). Thanks to this configuration, the CPG will provide stable oscillations but with variable activity, which will be used to select the gait pattern to be adopted by the robotic platform.

Table 3. Neuron parameters for the proposed WTA network in SpiNNaker hardware platform.

Parameter	Value
c_m	0.25 nF
τ_m	12.5 ms
τ_r	0.001 ms
u_{reset}	-70.0 mV
u_{rest}	-65.0 mV
u_{th}	-50.0 mV
τ_{syn_e}	7.0 ms
τ_{syn_i}	17.1 ms
I_{bias}	0.0 mA

WTA Network: The neurons of the WTA network receive the activity generated by the DVS retina and their output will be used to select the direction to be followed by the robotic platform. These populations incorporate the same parameters as those of CPG_{AB}, except for the injected current (Table 3). For this reason, the activity or inactivity of these neurons is due solely to the input activity received by these populations from the FPGA. The topology of this network and its interaction with the DVS can be seen in Figure 5.

Since the retina is a 128×128 pixel matrix, to select the direction of the robotic platform, it was decided to divide the X coordinate into three windows: the first 39 pixel columns are linked to the right direction, the next 50 to the center, and the last 39 to the left. In this way, when a pixel detects changes in brightness, the direction AER associated with that pixel will be sent to SpiNNaker. Depending on its X coordinate, this address will be used to excite one of the three populations in SpiNNaker. The weights of the synaptic connections are configured so that the population that generates the most activity is the winner and strongly inhibits the previous two; thanks to this, only the AER events of a single winning population will be transmitted to the robotic platform at a time, allowing it to decide which direction to follow.

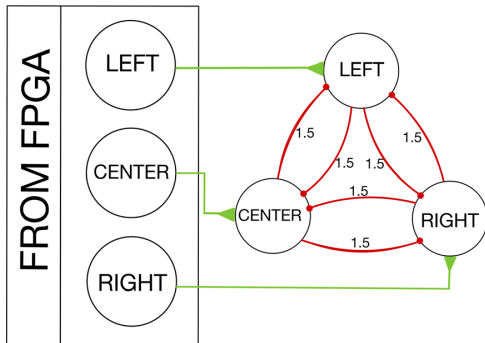


Figure 5. DVS adaptive network topology.

2.2.2. Complete SNN Topology

The complete topology of the neural network that allows controlling the speed and orientation of the robotic platform is shown in Figure 6 and its distribution of populations and neurons is shown in Table 4.

3. Experiments and Results

This section describes the experiments conducted with respect to the proposed spiking architecture. Four experiments are carried out without the retina to check the performance of the network considering the external sensor changing value. Another four

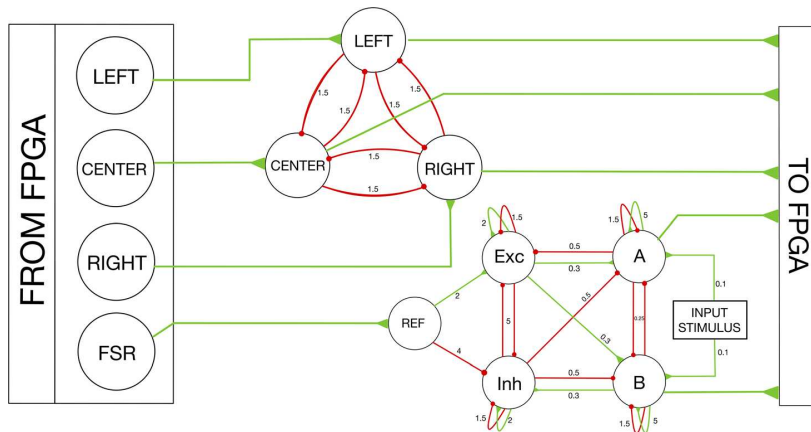


Figure 6. Complete topology of the adaptive network with a total of 9 populations and 579 neurons.

Table 4. Number of neurons per population in the complete network.

Population	Number of neurons
A	100
B	100
Exc	100
Inh	100
Ref	50
Center	50
Right	39
Left	39
FSR	1

experiments include the retina to show the network decision considering the spiking input stimulus.

The first experiment includes a 10 Hz input stimulus from the population bound to the FSR (minimum value provided by the FSRs). The results are shown in **Figure 7**. It can be seen that the activity of POP_{EXC} and POP_{FSR} is very similar. Due to the interaction of POP_{EXC} and POP_{INH} with CPG_{AB}, the average oscillation frequency of the obtained POP_{CPG} is 7 Hz. It is important to note that the frequency of the CPG does not linearly adapt to the input frequency of the FSR. It rather moves in a narrow frequency range (between 5 and 20 Hz) which is enough to be able to select and modify the gait pattern of the robotic platform.

The second experiment includes, as stimulus, intermediate frequencies of 85 Hz from the population bound to the FSR. The results are shown in **Figure 8**. In this experiment, it can be observed how the activity of the POP_{EXC} population is greatly

increased by the excitation received by POP_{FSR}. It is interesting to highlight how, although increasing the frequency of POP_{FSR} should decrease the frequency of POP_{INH}, the opposite effect occurs due to the increase in the oscillation frequency of POP_{CPG}, which excites, albeit weakly, POP_{INH}. The average oscillation frequency of the CPG is 13 Hz.

As a third experiment, the CPG network was subjected to constant inputs of 171 Hz (maximum values provided by the FSRs) from the population bound to them. The results are shown in **Figure 9**. In this case, it can be observed how POP_{EXC} is strongly excited by POP_{FSR}, while the activity of POP_{INH} is considerably reduced. In this scenario, not only does the oscillation frequency of POP_{CPG} increase, which is around 20 Hz, but also the number of shots at a given time instant in each population of POP_{CPG} increases.

Finally, a last experiment has been performed in which the input linked to the FSR is oscillatory, and it can be observed how the CPG adapts to the input by increasing or decreasing its total frequency (the frequency obtained by considering both populations of the CPG) and its individual frequency (the frequency of each neuron within each population at individual time points when its population is firing). The results are shown in **Figure 10**.

As mentioned previously, the WTA network will receive all AER addresses associated with retinal pixels that have detected changes in luminosity. It is possible to receive up to 20 000 addresses per second on SpiNNaker, each representing one of the 128 possible directions linked to the X-coordinate of the retina. Various experiments have been conducted to subject the network to different stimuli, using a pen to create motion in front of the retina and activate the pixels. It is important to note that these scenarios propose situations that will not occur in a real and

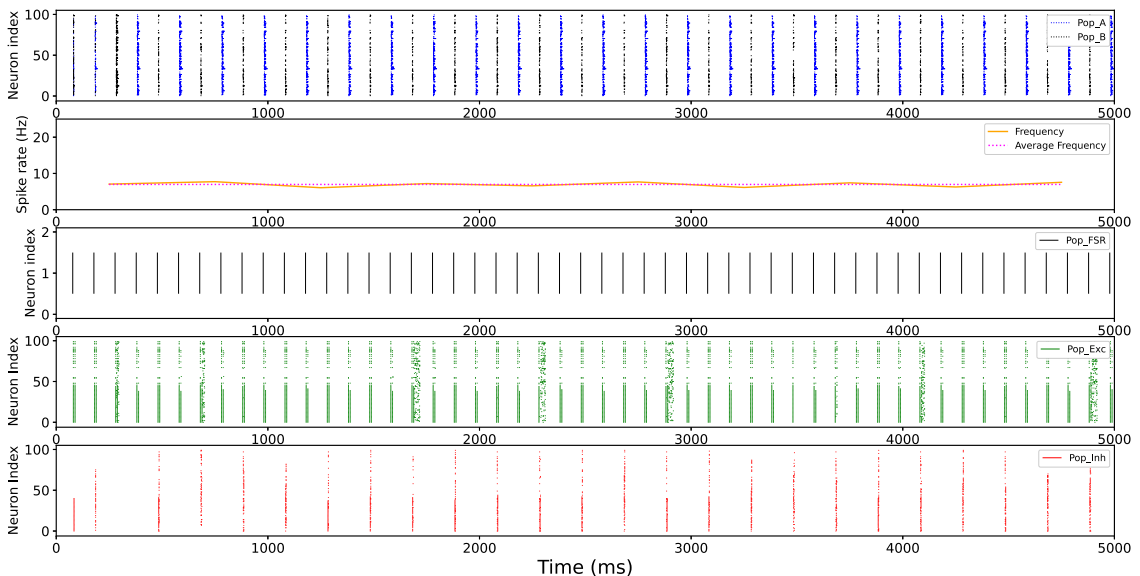


Figure 7. Behavior of the CPG network at 10 Hz inputs. Thanks to the regulating action of the POP_{EXC} and POP_{INH} populations, the output frequency of the CPG_{AB} remains stable around 7 Hz.

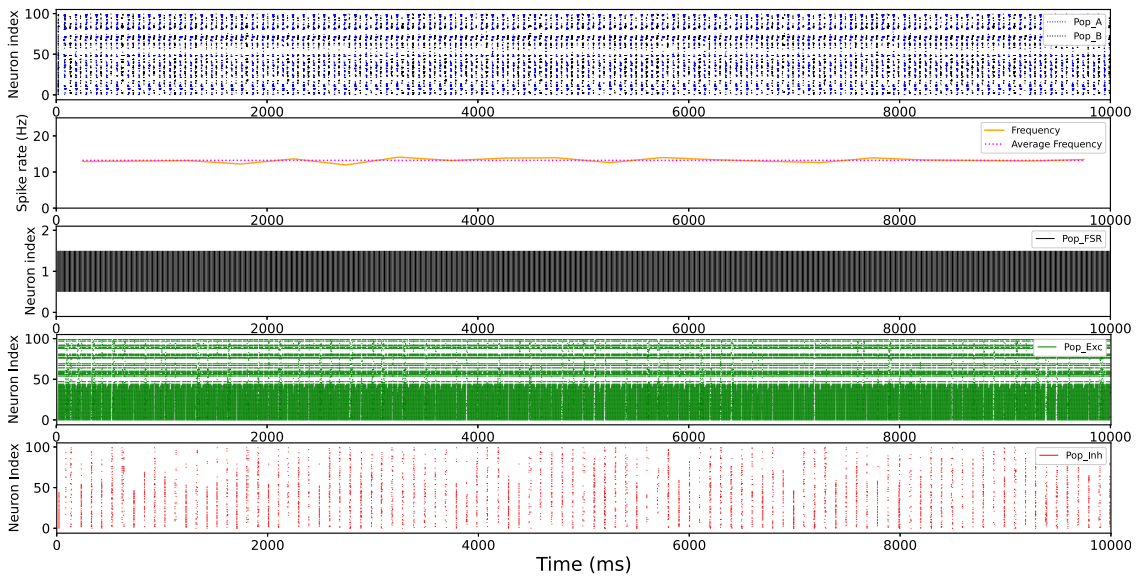


Figure 8. Behavior of the CPG network at 85 Hz inputs. Intermediate frequencies cause a generalized increase in oscillations in all populations, while POP_{INH} and POP_{EXC} keep the CPG stable throughout the duration of the experiment.

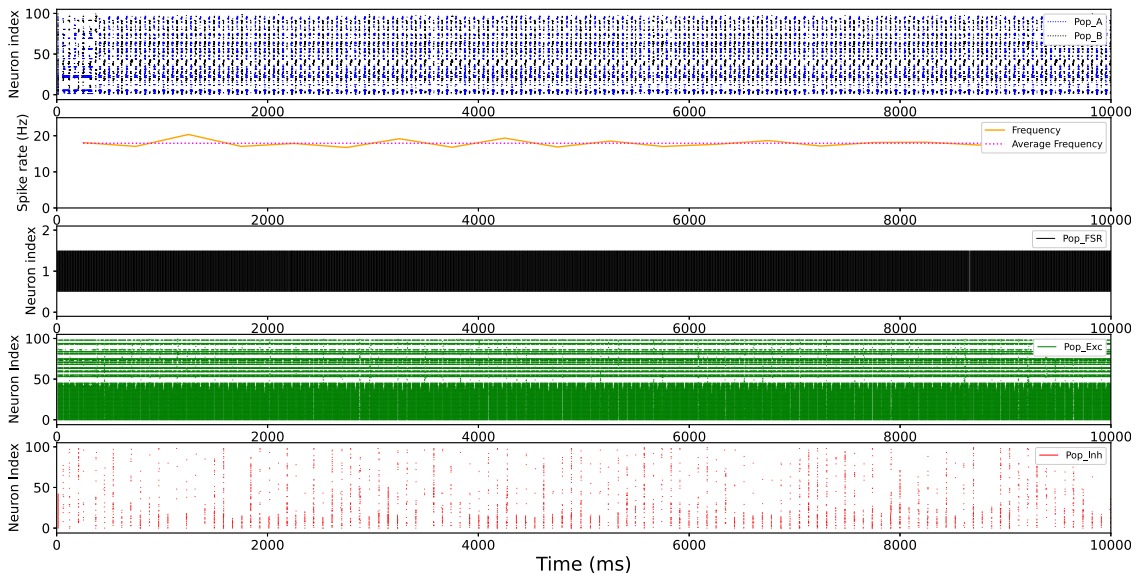


Figure 9. Behavior of the CPG network in response to 171 Hz inputs. It is interesting to observe how the significant excitation caused by POP_{EXC} on the CPG not only increases its frequency to 20 Hz but also leads to an increase in the number of neurons firing in each population of the CPG during the moments when they are active.

controlled environment of the robotic platform, such as receiving similar changes in luminosity in more than one window simultaneously or introducing noise into one of the windows to assess the network's behavior in response to these phenomena.

First, a higher activity was generated in the left region of the DVS, where it can be observed that during the entire simulation the stimulus was generated intermittently in the POP_{LEFT} region intermittently, generating peaks of minimum activity (e.g.,

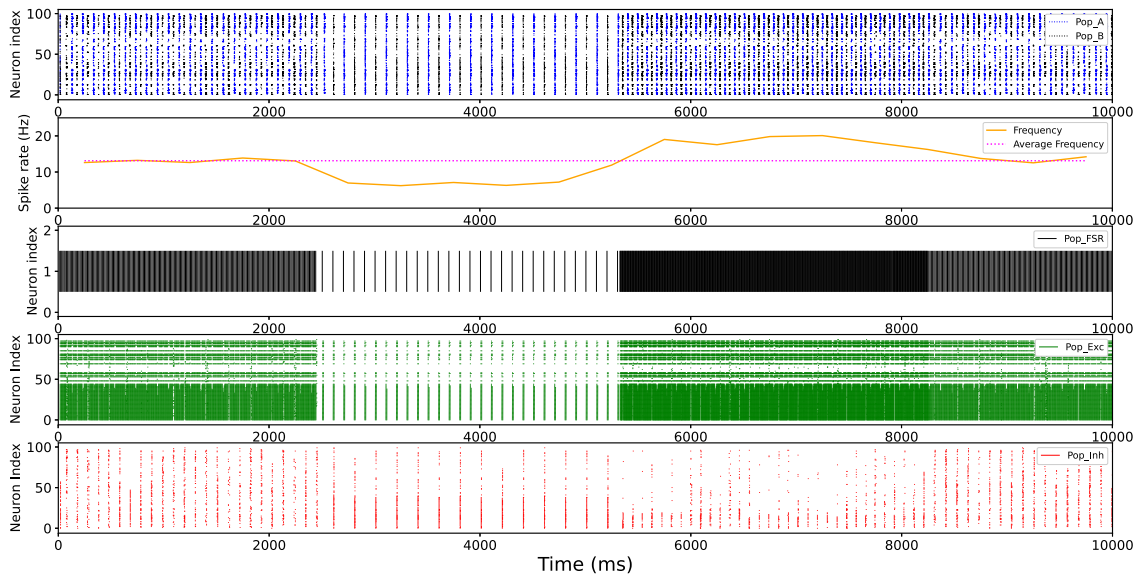


Figure 10. Behavior of the CPG network in response to oscillatory inputs. The CPG adjusts its frequency based on the input associated with the FSRs, demonstrating its adaptation to external stimuli.

$t = 1000$, $t = 2000$, and $t = 4000$ ms) in some instants due to the stimulus coming out of the visible range of the retina. The results are shown in **Figure 11**.

Another experiment was also carried out in which more activity was generated in the central region of the DVS. The results can be seen in **Figure 12**, and it is possible to check that the detection of events in all the windows does not imply activity in all of them at the output of the WTA network. The activity will be maintained only in the population that has generated the highest number of events in a given time.

The same experiment was also performed generating increased activity in the right region of the DVS. In this experiment, it is again verified that, although the central population has produced some events in the instants $t = 800$ and $t = 3000$ ms, the output of the WTA network only contemplates the events of the winning population. The results can be seen in **Figure 13**.

Finally, an experiment has also been performed in which the WTA network is subjected to random activity over the entire visual field of the DVS. The result of this experiment can be seen in **Figure 14**. It is interesting to see how the network shows tolerance to instantaneous noise. For example, although at $t = 1200$ ms, the winning population at that instant is POP_{Left}, at the output of the WTA POP_{Center} remains the winner because POP_{Left} has not generated enough activity for long enough to exceed the activity previously generated by POP_{Center}.

4. Discussion

The set of experiments conducted demonstrates: 1) the ability of the sCPG model to adapt to variable inputs from the FSRs by changing its oscillation frequency to increase or decrease the

speed of the robotic platform, respectively; and 2) the ability of the WTA network model to select the spatial region with the highest spiking activity from a DVS input, thus orienting the robotic platform toward that direction. Thanks to these two features, course and speed adaptation, it is possible to implement a fully functional “prey–predator” model where analog and neuromorphic sensor inputs are fused. This information can be used by the network implemented in SpiNNaker to make a decision about the behavior of the robotic platform.

From a bioinspired point of view, the biological features of the CPG (described in Section 1) show the functions that these neuronal structures can perform in living beings, as well as the characteristics of the neurons that form this type of neural structure. The CPG design includes biologically plausible structures.

In the biological model, the neurons that make up the CPG can be either tonic or phasic. In the former case, tonic neurons emit the neurotransmitter at a constant but low rate, whereas phasic neurons act almost instantly, emitting a rapid pulse of activity to bring postsynaptic neurons into action quickly. In this work, both types of neurons have been implemented: on the one hand, CPG_{AB} is made up entirely of tonic neurons. They include a very low (0.1 nA) but constant current injection, which ensures that even if no stimulus was received from the FSRs, the oscillation of the CPGs could be maintained over time, fulfilling the intrinsic rhythm generation that biological CPGs show. On the other hand, the POP_{Exc} and POP_{Inh} populations act as phasic neurons, since at a minimum excitation signal received, they are able to reach the threshold to produce a neuronal firing: exciting or inhibiting the CPG_{AB} almost instantly. With this feature, the centralized control to which the CPG is subjected is modeled. Although CPGs can function autonomously, they are often under

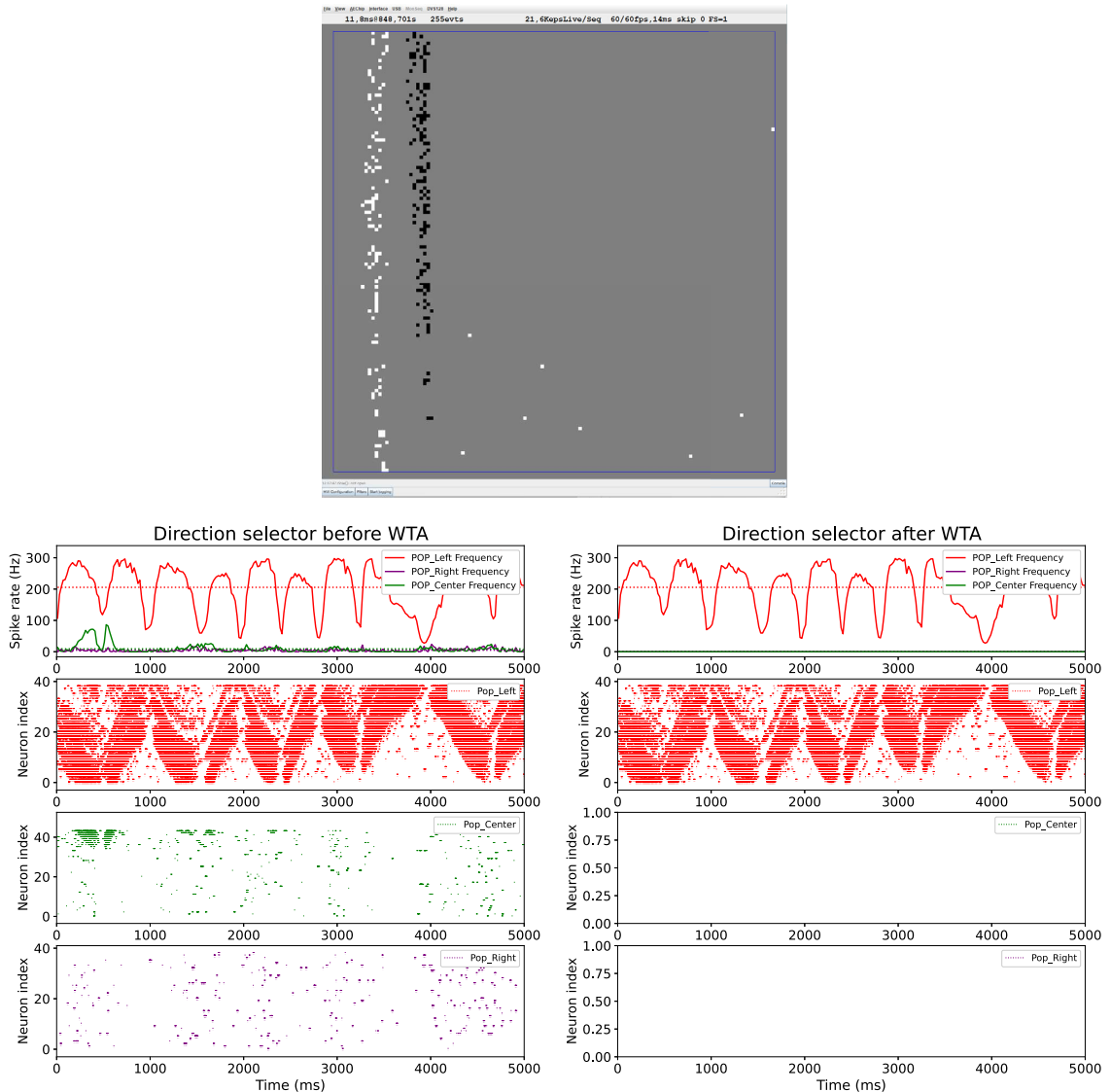


Figure 11. Behavior of the WTA network in response to events generated in the left region.

control of the central nervous system to adapt to the needs of the surrounding environment.

Another feature of biological CPGs is their plasticity, which means that they can adapt and change over time in response to different signals. In this work, the CPG implemented can adapt to the conditions of its environment just as any biological being would do. The FSRs modify the oscillation frequency of the CPG, which results in a regulation of the gait of the robotic platform depending on the terrain. This functioning is similar to that of the vestibular system of living beings. Thanks to this behavior, the condition of variability presented

by the biological CPGs, which adapt their patterns in response to different conditions, is also fulfilled. To complete the CPG model, the DVS is used in the network to select the orientation to be followed by the robotic platform, imitating the decision-making capacity that living beings have when taking a course based on what their optical system detects (prey, dangers, obstacles, etc.).

However, there are also features of the CPG presented in this work that may differ with respect to biological ones. The number of neurons used in this work has been estimated only with the stability of the whole system as a priority, reducing almost to zero

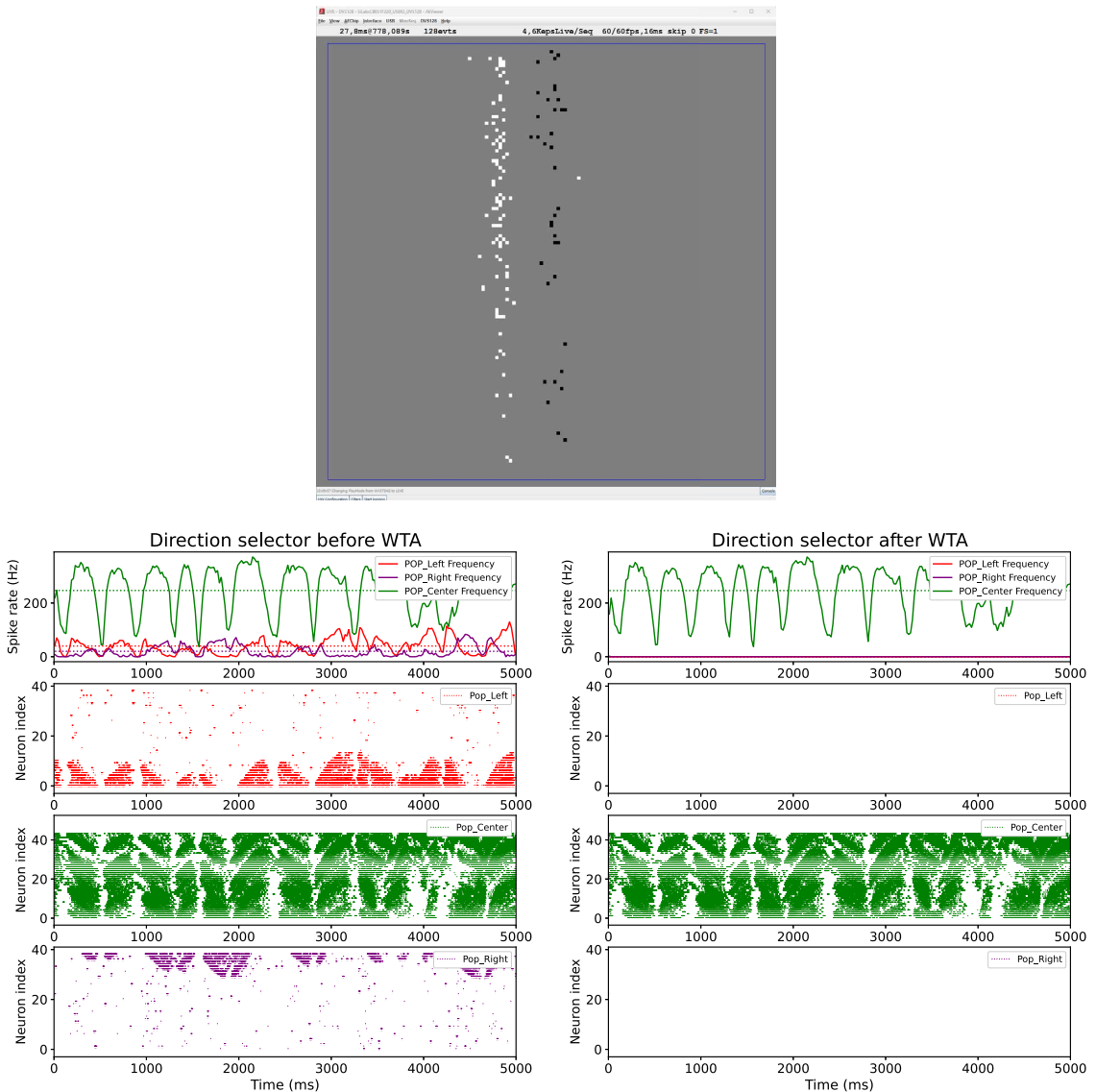


Figure 12. Behavior of the WTA network in response to events generated in the central region.

the scenarios in which the network output is unstable or noncoherent. In biological CPGs focused on locomotion, a relatively complex neural network is required (depending on factors such as species or movement pattern), but it is estimated that in mammals, such as humans, these CPGs can be composed of thousands of neurons.^[2] Furthermore, while in this work the CPG is only responsible for adjusting the speed of the robotic platform, in the biological environment, these neural structures are used simultaneously to control speed, direction, balance, and other aspects of locomotion according to the specific needs of the

organism and environmental conditions, without being the only ones involved in this task.^[1,7]

Another important aspect to mention, which is related to the previous one, is the use of a WTA network for decision making regarding the orientation of the robotic platform. While these networks may model certain aspects of competition and inhibition observed in some brain functions, they do not directly or specifically resemble a particular biological neural structure of the nervous system, leaving this function in the biological environment reserved, again, for the CPGs.^[54]

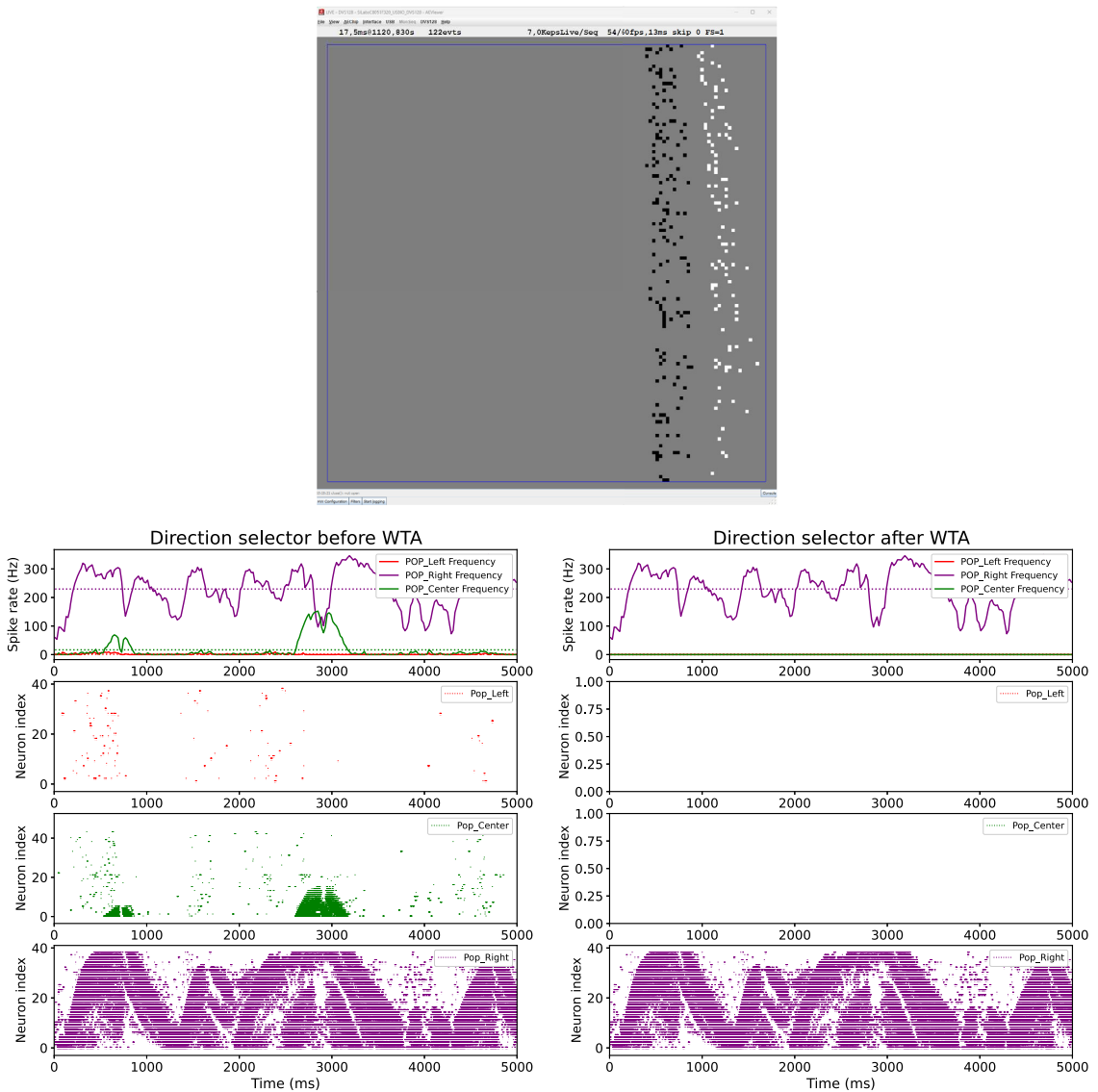


Figure 13. Behavior of the WTA network in response to events generated in the right region.

Finally, in terms of computational resources and power consumption, the network presented in this work uses a total of 579 neurons with around 200.000 synapses, while other works such as refs. [20,31] use a much smaller number of neurons to control the robotic platform (both of these works used 12 neurons to build the entire network). Due to the high number of neurons used, more power consumption is expected to train and run the network. However, this system is potentially more efficient than those based on ANNs. This greater efficiency is mainly due to the fact that its neurons only activate in response

to a specific stimulus, thereby reducing the required computational calculation. In our system, this translates into lower energy consumption when processing sensory inputs. Particularly, by using the WTA network to process signals from a DVS, we observed reduced energy consumption compared to equivalent processing by ANNs. This is primarily due to the sporadic and event-based nature of the DVS, which consumes energy only when its pixels detect changes in brightness. In contrast, conventional cameras process frames constantly, implying higher energy consumption.

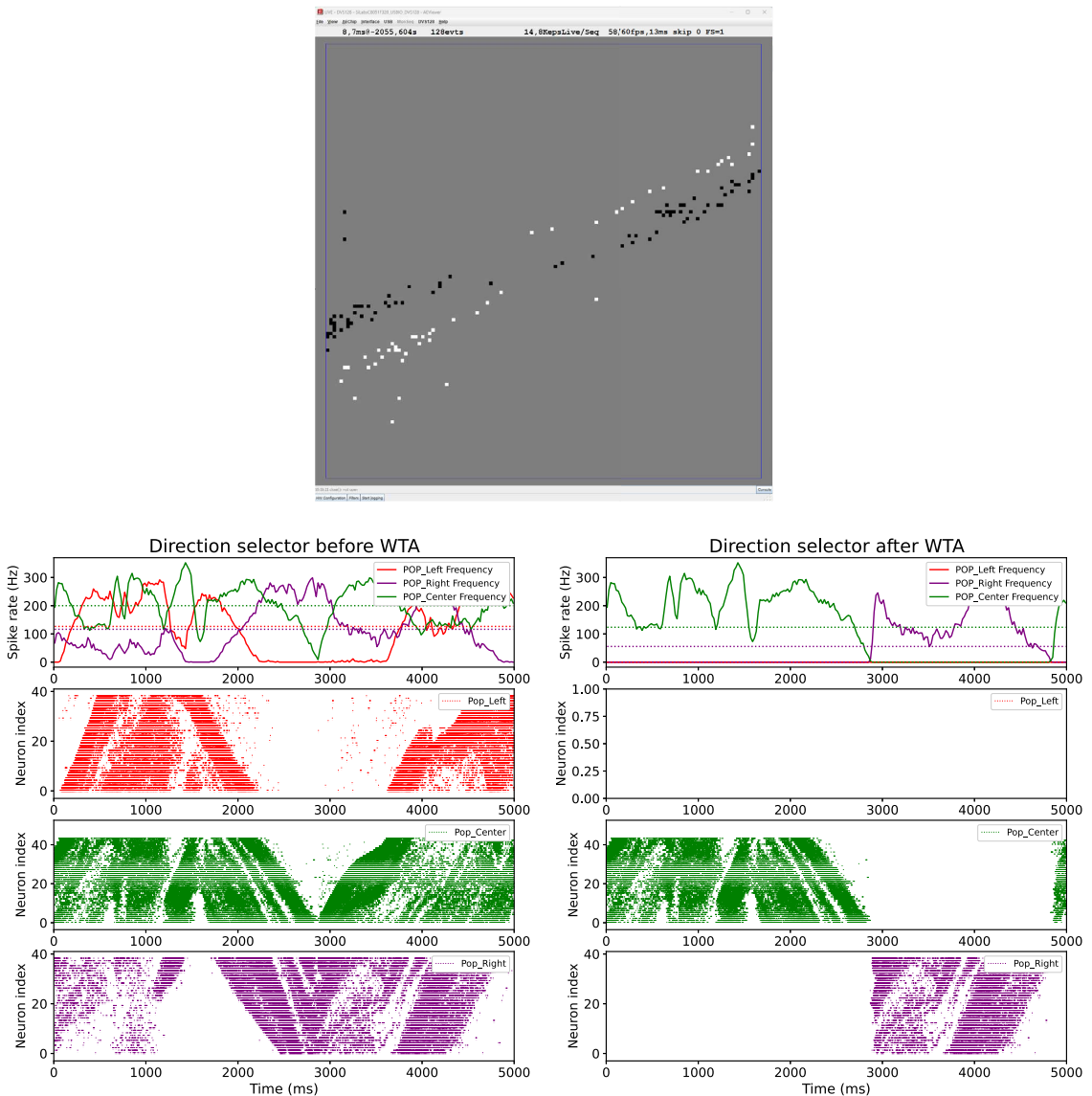


Figure 14. Behavior of the WTA network in response to randomly generated events.

5. Conclusion

In this article, a sCPG model implemented on SpiNNaker that receives information from analog and neuromorphic sensors simultaneously is proposed. The architecture presented includes a closed-loop control to adapt locomotion frequencies and orientation that can be applied to any legged robot. The information from the environment provided by the FRSs and the DVSS is processed instantaneously by SpiNNaker, which generates a frequency output that is sent to the FPGAs and eventually used to control the actuators.

The system has been validated using data from irregular surfaces. These data generate pressure differences on the FSRs, which in turn cause variations in the voltage emitted by these sensors. The obtained values are converted into spikes, which are received and processed by SpiNNaker. Subsequently, SpiNNaker adjusts the frequency of the CPG and the gait pattern of the robotic platform based on these signals.

Furthermore, the proper operation of the WTA network to determine the direction of the robotic platform should follow has been verified. The WTA network is based on activity detected

in various regions of the DVS. The results of the experiments show that the WTA network effectively selects the winning region, completely excluding the other two options. Similar to the CPG, the output of the WTA network is sent to the FPGA, which in turn adjusts the orientation of the robotic platform based on this information.

In future work, the architecture presented will be tested in robots with different leg configurations, which will allow us to demonstrate its reproducibility on other robotic platforms, regardless of the number of legs they have. Furthermore, the DVS will be used not only to follow the direction where the highest activity is, but also to incorporate into the network a learning routine that allows it to recognize specific shapes or patterns, which would induce the robotic platform to pursue a specific target by recognizing its shape and not only the generated activity. In addition, other neuromorphic sensors such as the NAS will be included in the loop to bring robotic behavior closer to biology.

Acknowledgements

This research was partially supported by project PID2019-105556GB-C33 funded by MCIN/AEI/10.13039/501100011033.

Conflict of Interest

The authors declare no conflict of interest.

Data Availability Statement

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

Keywords

adaptive learning, central pattern generators, neuromorphic hardware, neurorobotics, spiking neural networks, SpiNNaker

Received: October 9, 2023

Revised: December 19, 2023

Published online:

[1] S. Grillner, P. Wallen, *Annu. Rev. Neurosci.* **1985**, *8*, 233.
 [2] M. MacKay-Lyons, *Phys. Ther.* **2002**, *82*, 69.
 [3] G. M. Hughes, C. A. Wiersma, *J. Exp. Biol.* **1960**, *37*, 657.
 [4] R. M. Harris-Warrick, *Prog. Brain Res.* **2010**, *187*, 213.
 [5] P. A. Guertin, *Front. Neurol.* **2013**, *3*, 183.
 [6] S. L. Hooper, *Curr. Biol.* **2000**, *10*, R176.
 [7] E. Marder, D. Bucher, *Curr. Biol.* **2001**, *11*, R986.
 [8] A. J. Ijspeert, *Neural networks* **2008**, *21*, 642.
 [9] J. Kahn, A. Roberts, *Philos. Trans. R. Soc. London, Ser. B* **1982**, *296*, 229.
 [10] E. A. Arbas, R. L. Calabrese, *J. Comp. Physiol. A* **1984**, *155*, 783.
 [11] C. Von Euler, *J. Appl. Physiol.* **1983**, *55*, 1647.
 [12] C. Mead, *Nat. Electron.* **2020**, *3*, 434.
 [13] A. Linares-Barranco, F. Perez-Peña, A. Jimenez-Fernandez, E. Chicca, *Front. Neurobot.* **2020**, *14*, 590163.

[14] D. Gutierrez-Galan, J. P. Dominguez-Morales, F. Perez-Peña, A. Jimenez-Fernandez, A. Linares-Barranco, *Neurocomputing* **2020**, *381*, 10.
 [15] Y. Kim, Y. Li, H. Park, Y. Venkatesha, A. Hambitzer, P. Panda, in *Proc. of the AAAI Conf. on Artificial Intelligence*, Vol. 37, Washington D.C., February **2023**, pp. 8308–8316.
 [16] G.-I. Uleru, M. Hulea, A. Barleanu, *Biomimetics* **2023**, *8*, 28.
 [17] P. Lopez-Osorio, A. Patiño-Saucedo, J. P. Dominguez-Morales, H. Rostro-Gonzalez, F. Perez-Peña, *Neurocomputing* **2022**, *502*, 57.
 [18] G. Indiveri, B. Linares-Barranco, T. J. Hamilton, A. V. Schaik, R. Etienne-Cummings, T. Delbruck, S.-C. Liu, P. Dudek, P. Häfliger, S. Renaud, J. Schemmel, G. Cauwenberghs, J. Arthur, K. Hynna, F. Fowolosele, S. Saighi, T. Serrano-Gotarredona, J. Wijekoon, Y. Wang, K. Boahen, *Front. Neurosci.* **2011**, *5*, 73.
 [19] L. Righetti, A. J. Ijspeert, in *2008 IEEE Int. Conf. on Robotics and Automation*, IEEE, Pasadena, CA, May **2008**, pp. 819–824.
 [20] H. Rostro-Gonzalez, P. A. Cerna-Garcia, G. Trejo-Caballero, C. H. Garcia-Capulin, M. A. Ibarra-Manzano, J. G. Avina-Cervantes, C. Torres-Huitzil, *Neurocomputing* **2015**, *170*, 47.
 [21] B. Cuevas-Arteaga, J. P. Dominguez-Morales, H. Rostro-Gonzalez, A. Espinal, A. F. Jimenez-Fernandez, F. Gomez-Rodriguez, A. Linares-Barranco, in *Advances in Computational Intelligence: 14th Int. Work-Confer. on Artificial Neural Networks, IWANN 2017, Cadiz, Spain, June 14–16, 2017, Proceedings, Part 1*, Springer, Cádiz, Spain, June **2017**, pp. 548–559.
 [22] S. B. Furber, F. Galluppi, S. Temple, L. A. Plana, *Proc. IEEE* **2014**, *102*, 652.
 [23] B. Strohmmer, P. Manoonpong, L. B. Larsen, *Front. Neurobot.* **2020**, *14*, 41.
 [24] M. Davies, N. Srinivasa, T.-H. Lin, G. Chinya, Y. Cao, S. H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain, Y. Liao, C.-K. Lin, A. Lines, R. Liu, D. Mathaikutty, S. McCoy, A. Paul, J. Tse, G. Venkataramanan, Y.-H. Weng, A. Wild, Y. Yang, H. Wang, *IEEE Micro* **2018**, *38*, 82.
 [25] E. Angelidis, E. Buchholz, J. Arreguit, A. Rougé, T. Stewart, A. von Arnim, A. Knoll, A. Ijspeert, *Neuromorph. Comput. Eng.* **2021**, *1*, 014005.
 [26] G. Endo, J. Morimoto, T. Matsubara, J. Nakanishi, G. Cheng, *Int. J. Rob. Res.* **2008**, *27*, 213.
 [27] G. Li, H. Zhang, J. Zhang, R. T. Bye, *Adv. Rob.* **2014**, *28*, 389.
 [28] S. A. Kurkin, D. D. Kulminskiy, V. I. Ponomarenko, M. D. Prokhorov, S. V. Astakhov, A. E. Hramov, *Chaos* **2022**, *3232*, 033117.
 [29] M. Wang, J. Yu, M. Tan, *Int. J. Adv. Rob. Syst.* **2014**, *11*, 170.
 [30] D. Korkmaz, G. Ozmen Koca, G. Li, C. Bal, M. Ay, Z. H. Akpolat, *J. Mar. Eng. Technol.* **2021**, *20*, 125.
 [31] A. Spaeth, M. Tebyani, D. Haussler, M. Teodorescu, in *2020 3rd IEEE Int. Conf. on Soft Robotics (RoboSoft)*, IEEE, New Haven, CT, May **2020**, pp. 46–51.
 [32] E. M. Izhikevich, *IEEE Trans. Neural Networks* **2004**, *15*, 1063.
 [33] D. Gutierrez-Galan, J. P. Dominguez-Morales, F. Perez-Peña, A. Jimenez-Fernandez, A. Linares-Barranco, in *2019 IEEE Int. Symp. on Circuits and Systems (ISCAS)*, IEEE, Sapporo, Hokkaido, Japan, May **2019**, pp. 1–1.
 [34] T. Schoepe, D. Gutierrez-Galan, J. P. Dominguez-Morales, H. Grotorex, A. F. Jiménez Fernández, A. Linares-Barranco, E. Chicca, *Neuromorph. Comput. Eng.* **2023**, *3*, 024009.
 [35] B. Strohmmer, R. K. Stagsted, P. Manoonpong, L. B. Larsen, *Front. Neurosci.* **2021**, *15*, 633945.
 [36] S. B. Furber, D. R. Lester, L. A. Plana, J. D. Garside, E. Painkras, S. Temple, A. D. Brown, *IEEE Trans. Comput.* **2012**, *62*, 2454.
 [37] S. Furber, P. Bogdan, *SpiNNaker: A Spiking Neural Network Architecture*, now publishers, Boston-Delft **2020**.
 [38] H. Markram, *Sci. Am.* **2012**, *306*, 50.

- [39] E. Painkras, L. A. Plana, J. Garside, S. Temple, F. Galluppi, C. Patterson, D. R. Lester, A. D. Brown, S. B. Furber, *IEEE J. Solid-State Circuits* **2013**, *48*, 1943.
- [40] L. A. Plana, S. B. Furber, S. Temple, M. Khan, Y. Shi, J. Wu, S. Yang, *IEEE Des. Test Comput.* **2007**, *24*, 454.
- [41] M. Mahowald, Ph.D. Dissertation, California Institute of Technology Pasadena **1992**.
- [42] L. A. Plana, J. Garside, J. Heathcote, J. Pepper, S. Temple, S. Davidson, M. Luján, S. Furber, *IEEE Access* **2020**, *8*, 84918.
- [43] A. Yousefzadeh, M. Jabłoński, T. Iakymchuk, A. Linares-Barranco, A. Rosado, L. A. Plana, S. Temple, T. Serrano-Gotarredona, S. B. Furber, B. Linares-Barranco, *IEEE Trans. Biomed. Circuits and Syst.* **2017**, *11*, 1133.
- [44] J. P. Dominguez-Morales, A. Jimenez-Fernandez, A. Rios-Navarro, E. Cerezuela-Escudero, D. Gutierrez-Galan, M. J. Dominguez-Morales, G. Jimenez-Moreno, in *Int. Conf. on Artificial Neural Networks*, Springer, Barcelona, Spain, September **2016**, pp. 45–53.
- [45] T. Schoepe, D. Gutierrez-Galan, J. P. Dominguez-Morales, A. Jimenez-Fernandez, A. Linares-Barranco, E. Chicca, in *2019 IEEE Biomedical Circuits and Systems Conf. (BioCAS)*. IEEE, Nara, Japan, October **2019**, pp. 1–4.
- [46] A. P. Davison, D. Brüderle, J. M. Eppler, J. Kremkow, E. Muller, D. Pecevski, L. Perrinet, P. Yger, *Front. Neuroinf.* **2009**, *2*, 11.
- [47] O. Rhodes, P. A. Bogdan, C. Breninkmeijer, S. Davidson, D. Fellows, A. Gait, D. R. Lester, M. Mikaitis, L. A. Plana, A. G. Rowley, A. B. Stokes, S. B. Furber, *Front. Neurosci.* **2018**, *12*, 816.
- [48] Diligent, <https://diligent.com/reference/programmable-logic/nexys-4-ddr/reference-manual>, visitado el 8 de septiembre de **2023**.
- [49] P. Lichtsteiner, C. Posch, T. Delbruck, *IEEE J. Solid-State Circuits* **2008**, *43*, 566.
- [50] F. Perez-Peña, A. Morgado-Estevez, A. Linares-Barranco, A. Jimenez-Fernandez, F. Gomez-Rodriguez, G. Jimenez-Moreno, J. Lopez-Coronado, *Sensors* **2013**, *13*, 15805.
- [51] A. Linares-Barranco, F. Gomez-Rodriguez, A. Jimenez-Fernandez, T. Delbruck, P. Lichtensteiner, in *2007 IEEE Int. Symp. on Circuits and Systems*, New Orleans, LA, May **2007**, pp. 1192–1195.
- [52] R. Serrano-Gotarredona, M. Oster, P. Lichtsteiner, A. Linares-Barranco, R. Paz-Vicente, F. Gómez-Rodríguez, L. Camuñas-Mesa, R. Berner, M. Rivas-Pérez, T. Delbruck, R. Douglas, P. Häfliger, G. Jimenez-Moreno, A. C. Ballcels, T. Serrano-Gotarredona, A. J. Acosta-Jimenez, B. Linares-Barranco, *IEEE Trans. Neural networks* **2009**, *20*, 1417.
- [53] Ax-12a - Robotis e-manual, <https://manual.robotis.com/docs/en/dxl/ax/ax-12a/> (accessed: August 2023).
- [54] R. Coultrip, R. Granger, G. Lynch, *Neural Networks* **1992**, *5*, 47.

Glosario de siglas

AdEx *Adaptive Exponential Integrate and Fire*. 66

AER *Address Event Representation*. 24, 79, 83–88, 106, 108–110, 127–129, 133, 134, 136, 137

ANC *Analog Network Core*. 15

ANN *Artificial Neural Network*. 2, 4, 56, 57, 70

API *Application Programming Interface*. 81

BMP *Board Management Processor*. 100, 101

bps *Bits por segundo*. 91

Caltech *Instituto de Tecnología de California*. 14

CAM *Content-Addressable Memory*. 23

CAVIAR *Convolution AER Vision Architecture for Real-Time*. 86, 88

CMOS *complementary metal-oxide-semiconductor*. 15, 18, 21, 24, 106

CPG *Central Pattern Generator*. 2, 4, 5, 38, 60, 69–72, 114, 117, 118, 123, 131, 143–147, 150, 154, 156–160, 164, 165, 168–171, 180–182

dB *Decibelios*. 107

DNN *Deep Neural Network*. 56, 57

DSP *Digital Signal Processor*. 124

DVS *Dynamic Vision Sensor*. 79, 80, 105–110, 133, 135–137, 139, 144, 168, 172–176, 179, 181

DYNAP-SE *Dynamic Neuromorphic Asynchronous Processor*. 22, 23

E/S *Entrada/Salida*. 103

FPGA *Field Programmable Gate Array*. 6, 92–95, 100, 101, 114, 122, 124, 127–129, 133, 137, 139, 168, 171

FSR *Force-Sensing Resistor*. 114, 119, 120, 124, 128, 132, 137, 139, 144, 150, 152, 156, 160–164, 166–172, 179–181

GABA ácido gamma-aminobutírico. 50

GALS *Global Asynchronous Locally Synchronous*. 98

GCP *Generador Central de Patrones*. 4

GPU *Graphic Processing Unit*. 56

HDL *Hardware Description Language*. 128, 136

HH *Hodgkin-Huxley*. 63, 65

HiCANN *High input Count Analog Neural Network*. 15

ICTEAM *Tecnologías de la Información y la Comunicación, Electrónica y Matemáticas Aplicadas*. 24

IF *Integrate-and-Fire*. 67

ISI *Inter-Spike Interval*. 78

JTAG *Joint Test Action Group*. 103

LED *Light-Emitting Diode*. 101, 103, 126

LIF *Leaky Integrate-and-Fire*. 17, 23, 24, 64–68, 115, 117, 118

- MAC** *Media Access Control*. 102
- Mbps** *Megabits por segundo*. 91, 100, 102
- MLP** *Multi-Layer Perceptron*. 56
- NAS** *Neuromorphic Auditory Sensor*. 79, 80
- NAVIS** *Neuromorphic Auditory VISualizer Tool*. 79
- NoC** *Network-on-Chip*. 99
- NPU** *Neural Processing Unit*. 24, 25
- ODIN** *Online-learning Digital Spiking Neuromorphic processor*. 24
- PCB** *Printed Circuit Board*. 114, 130
- PEPS** *potencial excitatorio postsináptico*. 53–55, 61
- PFM** *Pulse Frequency Modulation*. 124, 125, 127–129
- PIPS** *potencial inhibitorio postsináptico*. 53, 54
- PSO** *Particle Swarm Optimizations*. 12
- PWM** *Pulse Width Modulation*. 124–128
- PyNN** *Python Neural Network*. 17
- RKUH** *Ruprecht-Karls-Universität Heidelberg*. 15
- RNP** *Red Neuronal Pulsante*. 4
- ROM** *Read Only Memory*. 102
- S1** *área somatosensorial primaria*. 31, 32
- sCPG** *spiking Central Pattern Generator*. 73, 113, 124, 129, 160, 161, 163, 165–167, 179
- SDRAM** *Synchronous Dynamic Random-Access Memory*. 98

- SII** *Corteza somatosensorial secundaria*. 32
- SNN** *Spiking Neural Network*. 2, 4, 5, 24, 57, 59–61, 68, 70, 73, 78, 82, 126
- SONAR** *Sound Navigation And Ranging*. 12
- SpiNNaker** *Spiking Neural Network Architecture*. 6, 16–18, 97
- SRAM** *Static Random Access Memory*. 24
- STDP** *Spike Timing Dependent Plasticity*. 17, 74
- TUD** *Technische Universität Dresden*. 15
- UART** *Universal Asynchronous Receiver/Transmitter*. 93
- UCLouvain** *Universidad Católica de Louvain*. 24
- USB** *Universal Serial Bus*. 79
- VLSI** *Very Large-Scale Integration*. 14, 83, 84
- WTA** *Winner Take All*. 60, 73, 74, 114, 130, 133, 134, 137, 144, 172–176, 179, 181

Referencias

- [1] Oludare Isaac Abiodun et al. «State-of-the-art in artificial neural network applications: A survey». En: Heliyon 4.11 (2018).
- [2] Mary Ann C Maher et al. «Implementing neural architectures using analog VLSI circuits». En: IEEE transactions on circuits and systems 36.5 (1989), págs. 643-652.
- [3] Carver A Mead y Misha A Mahowald. «A silicon model of early visual processing». En: Neural networks 1.1 (1988), págs. 91-97.
- [4] Patrick Lichtsteiner, Christoph Posch y Tobi Delbruck. «A 128x128 120 dB 15 microseconds latency asynchronous temporal contrast vision sensor». En: IEEE journal of solid-state circuits 43.2 (2008), págs. 566-576.
- [5] André van Schaik y Shih-Chii Liu. «AER EAR: A matched silicon cochlea pair with address event representation interface». En: 2005 IEEE International Symposium on Circuits and Systems (ISCAS). IEEE. 2005, págs. 4213-4216.
- [6] Angel Jiménez-Fernández et al. «A binaural neuromorphic auditory sensor for FPGA: a spike signal processing approach». En: IEEE transactions on neural networks and learning systems 28.4 (2016), págs. 804-818.
- [7] Angel Jimenez-Fernandez et al. «A neuro-inspired spike-based PID motor controller for multi-motor robots with low cost FPGAs». En: Sensors 12.4 (2012), págs. 3831-3856.
- [8] Fernando Perez-Peña et al. «Neuro-inspired spike-based motion: From dynamic vision sensor to robot motor open-loop control through spike-VITE». En: Sensors 13.11 (2013), págs. 15805-15832.
- [9] Francisco Gómez-Rodríguez et al. «ED-Scorbot: A Robotic test-bed Framework for FPGA-based Neuromorphic systems». En: 2016 6th IEEE International Conference on Biomedical Robotics and Biomechanics (BioRob). IEEE. 2016, págs. 237-242.

-
- [10] Shih-Chii Liu y Tobi Delbruck. «Neuromorphic sensory systems». En: *Current opinion in neurobiology* 20.3 (2010), págs. 288-295.
 - [11] Daniel Gutierrez-Galan et al. «OpenNAS: Open source neuromorphic auditory sensor HDL code generator for FPGA implementations». En: *Neurocomputing* 436 (2021), págs. 35-38.
 - [12] Charles Darwin's. «On the origin of species». En: published on 24 (1859), pág. 1.
 - [13] Janine Benyus et al. «Innovation inspired by nature». En: *William Morrow, New York* (1997).
 - [14] Stephen J O'Leary, Rachael R Richardson y Hugh J McDermott. «Principles of design and biological approaches for improving the selectivity of cochlear implant electrodes». En: *Journal of neural engineering* 6.5 (2009), pág. 055002.
 - [15] Zhi-Zhi Sheng et al. «Bioinspired approaches for medical devices». En: *Chinese Chemical Letters* 28.6 (2017), págs. 1131-1134.
 - [16] Rohan Kapoor et al. «A bio-inspired acoustic sensor system for UAS navigation and tracking». En: *2017 IEEE/AIAA 36th digital avionics systems conference (DASC)*. IEEE. 2017, págs. 1-7.
 - [17] James Kennedy y Russell Eberhart. «Particle swarm optimization». En: *Proceedings of ICNN'95-international conference on neural networks*. Vol. 4. IEEE. 1995, págs. 1942-1948.
 - [18] Giacomo Indiveri et al. «Neuromorphic silicon neuron circuits». En: *Frontiers in neuroscience* 5 (2011), pág. 73.
 - [19] Kwabena Boahen. «Neuromorphic microchips». En: *Scientific American* 292.5 (2005), págs. 56-63.
 - [20] Carver Mead y Mohammed Ismail. *Analog VLSI implementation of neural systems*. Vol. 80. Springer Science & Business Media, 1989.
 - [21] Shih-Chii Liu. *Analog VLSI: circuits and principles*. MIT press, 2002.
 - [22] Johannes Schemmel et al. «A wafer-scale neuromorphic hardware system for large-scale neural modeling». En: *2010 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE. 2010, págs. 1947-1950.
 - [23] Stefan Scholze et al. «VLSI implementation of a 2.8 Gevent/s packet-based AER interface with routing and event sorting functionality». En: *Frontiers in neuroscience* 5 (2011), pág. 117.
 - [24] Karlheinz Meier. «A mixed-signal universal neuromorphic computing system». En: *2015 IEEE International Electron Devices Meeting (IEDM)*. IEEE. 2015, págs. 4-6.

Referencias

- [25] Mihai A Petrovici et al. «Characterization and compensation of network-level anomalies in mixed-signal neuromorphic modeling platforms». En: PloS one 9.10 (2014), e108590.
- [26] Simon Friedmann et al. «Demonstrating hybrid learning in a flexible neuromorphic hardware system». En: IEEE transactions on biomedical circuits and systems 11.1 (2016), págs. 128-142.
- [27] Andrew P Davison et al. «PyNN: a common interface for neuronal network simulators». En: Frontiers in neuroinformatics 2 (2009), págs. 388.
- [28] Steve B Furber et al. «The SpiNNaker project». En: Proceedings of the IEEE 102.5 (2014), págs. 652-665.
- [29] Simon Segars. «The ARM9 family-high performance microprocessors for embedded applications». En: Proceedings International Conference on Computer Design. VLSI in Computers and Processors (Cat. No. 98CB36273). IEEE. 1998, págs. 230-235.
- [30] Eugene M Izhikevich. «Simple model of spiking neurons». En: IEEE Transactions on neural networks 14.6 (2003), págs. 1569-1572.
- [31] Jeffrey C Magee y Daniel Johnston. «A synaptically controlled, associative signal for Hebbian plasticity in hippocampal neurons». En: Science 275.5297 (1997), págs. 209-213.
- [32] Christian Mayr, Sebastian Hoepfner y Steve Furber. «Spinnaker 2: A 10 million core processor system for brain simulation and machine learning». En: arXiv preprint arXiv:1911.02385 (2019).
- [33] Ben Varkey Benjamin et al. «Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations». En: Proceedings of the IEEE 102.5 (2014), págs. 699-716.
- [34] Alexander Neckar et al. «Braindrop: A mixed-signal neuromorphic architecture with a dynamical systems-based programming model». En: Proceedings of the IEEE 107.1 (2018), págs. 144-164.
- [35] K Boahen. «Neurogrid: Simulating a million neurons and a billion synapses with sixteen neuromorphic chips». En: Frontiers in Computational Neuroscience 40 (2012).
- [36] Antara Ganguly, Rajeev Muralidhar y Virendra Singh. «Towards energy efficient non-von neumann architectures for deep learning». En: 20th international symposium on quality electronic design (ISQED). IEEE. 2019, págs. 335-342.
- [37] Filipp Akopyan et al. «Truenorth: Design and tool flow of a 65 mw 1 million neuron programmable neurosynaptic chip». En: IEEE transactions on

- computer-aided design of integrated circuits and systems 34.10 (2015), págs. 1537-1557.
- [38] Wei Wen et al. «A new learning method for inference accuracy, core occupation, and performance co-optimization on TrueNorth chip». En: Proceedings of the 53rd Annual Design Automation Conference. 2016, págs. 1-6.
- [39] Lucas Antón Pastur-Romay et al. «Deep artificial neural networks and neuromorphic chips for big data analysis: pharmaceutical and bioinformatics applications». En: International journal of molecular sciences 17.8 (2016), pág. 1313.
- [40] Mark Barnell et al. «Demonstrating Advanced Machine Learning and Neuromorphic Computing Using IBM's NS16e». En: Intelligent Computing: Proceedings of the 2020 Computing Conference, Volume 1. Springer. 2020, págs. 1-11.
- [41] Mike Davies et al. «Loihi: A neuromorphic manycore processor with on-chip learning». En: IEEE Micro 38.1 (2018), págs. 82-99.
- [42] Carlo Michaelis et al. «Brian2Loihi: An emulator for the neuromorphic chip Loihi using the spiking neural network simulator Brian». En: Frontiers in Neuroinformatics 16 (2022), pág. 1015624.
- [43] Garrick Orchard et al. «Efficient neuromorphic signal processing with loihi 2». En: 2021 IEEE Workshop on Signal Processing Systems (SiPS). IEEE. 2021, págs. 254-259.
- [44] Yoon Seok Yang y Yongtae Kim. «Recent trend of neuromorphic computing hardware: Intel's neuromorphic system perspective». En: 2020 International SoC Design Conference (ISOCC). IEEE. 2020, págs. 218-219.
- [45] Alberto Marchisio y Muhammad Shafique. «Embedded Neuromorphic Using Intel's Loihi Processor». En: Embedded Machine Learning for Cyber-Physical, IoT, and Edge Computing: Software Optimizations and Hardware/Software Codesign. Springer, 2023, págs. 137-172.
- [46] Dmitry Ivanov et al. «Neuromorphic artificial intelligence systems». En: Frontiers in Neuroscience 16 (2022), pág. 1513.
- [47] Fredrik Sandin y Mattias Nilsson. «Synaptic delays for insect-inspired temporal feature detection in dynamic neuromorphic processors». En: Frontiers in Neuroscience 14 (2020), pág. 150.
- [48] Natacha Vanattou-Saïbouline et al. «A robust model of Stimulus-Specific Adaptation validated on neuromorphic hardware». En: Scientific Reports 11.1 (2021), pág. 17904.

Referencias

- [49] Giacomo Indiveri. The Dynap-se chip | Institute of Neuroinformatics | UZH. 2023. URL: <https://www.ini.uzh.ch/en/research/groups/ncs/chips/Dynap-se.html>.
- [50] SynSense. Neuromorphic Chip DYNAP™-SE2 | SynSense. 2023. URL: <https://www.synsense.ai/products/neuromorphic-chip-dynap-se2/>.
- [51] Catherine D Schuman et al. «Opportunities for neuromorphic computing algorithms and applications». En: Nature Computational Science 2.1 (2022), págs. 10-19.
- [52] C.H. Frenkel. The ODIN 256: A Study on Reproducing the 20 Izhikevich Models. Accedido el 12/10/2023. Año. URL: <https://chfrenkel.github.io/#:~:text=The%20ODIN%20256,reproduce%20the%2020%20Izhikevich.>
- [53] Charlotte Frenkel et al. «A 0.086-mm² 12.7-pJ/SOP 64k-synapse 256-neuron online-learning digital spiking neuromorphic processor in 28-nm CMOS». En: IEEE transactions on biomedical circuits and systems 13.1 (2018), págs. 145-158.
- [54] Author Name. Neuromorphic Chip Research Around the World. Accedido el 12/10/2023. Sep. de 2019. URL: <https://www10.edacafe.com/blogs/editorial/2019/09/27/neuromorphic-chip-research-around-the-world/>.
- [55] De Ma et al. «Darwin: A neuromorphic hardware co-processor based on spiking neural networks». En: Journal of systems architecture 77 (2017), págs. 43-51.
- [56] AI researchers develop 'Darwin,' a neuromorphic chip based on spiking ... Accessed on October 12, 2023. URL: <https://phys.org/news/2015-12-ai-darwin-neuromorphic-chip-based.html#:~:text=Recently%2C%20researchers%20from%20Zhejiang%20University,NPU%29%2C%20a%20neuromorphic%20hardware.>
- [57] Brodmann's Areas. Consultado el 19 de octubre de 2023. 2023. URL: https://www.operativeneurosurgery.com/doku.php?id=brodmann_s_areas.
- [58] Stefan Geyer et al. «Functional neuroanatomy of the primate isocortical motor system». En: Anatomy and embryology 202 (2000), págs. 443-474.
- [59] Brodmann Areas Of The Brain: Anatomy And Functions. Consultado el 19 de octubre de 2023. 2023. URL: <https://www.simplypsychology.org/brodmann-areas.html>.

- [60] Human brain function evaluated with rCBF-SPECT Memory and pain related changes and new diagnostic possibilities in Alzheimer's disease. Consultado el 19 de octubre de 2023. 2023. URL: https://www.researchgate.net/publication/267227556_Human_brain_function_evaluated_with_rCBF-SPECT_Memory_and_pain_related_changes_and_new_diagnostic_possibilities_in_Alzheimer's_disease.
- [61] Karl Zilles. «Brodmann: a pioneer of human brain mapping—his impact on concepts of cortical organization». En: *Brain* 141.11 (2018), págs. 3262-3278.
- [62] The Premotor Cortex. NCBI.
- [63] M. Fabbri-Destro G. Rizzolatti. «Premotor Cortex». En: *Encyclopedia of Neuroscience* (2009).
- [64] «The Human Premotor Cortex Is 'Mirror' Only for Biological Actions». En: *ScienceDirect* (2013).
- [65] «Dorsal Premotor Cortex Neurons Signal the Level of Choice Difficulty». En: *ScienceDirect* (2016).
- [66] «Rhythm and time in the premotor cortex». En: *PLOS Biology* (2017).
- [67] Science Direct. «Primary motor cortex contains populations of neurons that respond to the activation of specific muscles». En: *Science Direct Topics* (2023). URL: <https://www.sciencedirect.com/topics/immunology-and-microbiology/primary-motor-cortex>.
- [68] TandF Online. «The Role of Primary Motor Cortex: More Than Movement Execution». En: *TandF Online* (2023). URL: <https://www.tandfonline.com/doi/full/10.1080/17588928.2018.1506749>.
- [69] Annual Reviews. «Plasticity and Primary Motor Cortex». En: *Annual Review of Neuroscience* (2023). URL: <https://www.annualreviews.org/doi/10.1146/annurev.neuro.23.1.393>.
- [70] MR Borich et al. «Understanding the role of the primary somatosensory cortex: Opportunities for rehabilitation». En: *Neuropsychologia* 79 (2015), págs. 246-255.
- [71] Yinghua Yu et al. «Layer-specific activation in human primary somatosensory cortex during tactile temporal prediction error processing». En: *NeuroImage* 248 (2022), pág. 118867.
- [72] Nicholas Paul Holmes et al. «Locating primary somatosensory cortex in human brain stimulation studies: experimental evidence». En: *Journal of Neurophysiology* 121.1 (2019), págs. 336-344.

Referencias

- [73] Wilder Penfield y Edwin Boldrey. «Somatic motor and sensory representation in the cerebral cortex of man as studied by electrical stimulation». En: *Brain* 60.4 (1937), págs. 389-443.
- [74] Fengqiao Sun et al. «Functional characteristics of the human primary somatosensory cortex: An electrostimulation study». En: *Epilepsy & Behavior* 118 (2021), pág. 107920.
- [75] J Ruben et al. «Somatotopic organization of human secondary somatosensory cortex». En: *Cerebral cortex* 11.5 (2001), págs. 463-473.
- [76] Antje Peters et al. «Visual perceptual load and processing of somatosensory stimuli in primary and secondary somatosensory cortices». En: *Scientific Reports* 13.1 (2023), pág. 7005.
- [77] Daniel G Taub et al. «The Secondary Somatosensory Cortex Gates Mechanical and Thermal Sensitivity». En: *bioRxiv* (2023), págs. 2023-05.
- [78] Rafael V Bretas et al. «Secondary somatosensory cortex of primates: beyond body maps, toward conscious self-in-the-world maps». En: *Experimental Brain Research* 238.2 (2020), págs. 259-272.
- [79] François Mauguière. «The role of secondary somatosensory cortex and insula in pain». En: *Supplements to Clinical Neurophysiology* 57 (2004), págs. 62-71.
- [80] Henk J Groenewegen et al. «The basal ganglia and motor control». En: *Neural plasticity* 10.1-2 (2003), págs. 107-120.
- [81] Kristina Simonyan. «Recent advances in understanding the role of the basal ganglia». En: *F1000Research* 8 (2019).
- [82] Ashesh K Dhawale et al. «The basal ganglia control the detailed kinematics of learned motor skills». En: *Nature neuroscience* 24.9 (2021), págs. 1256-1269.
- [83] Okihide Hikosaka. «Basal ganglia—possible role in motor coordination and learning». En: *Current opinion in neurobiology* 1.4 (1991), págs. 638-643.
- [84] Sopiko Jimsheleishvili y Marine Dididze. «Neuroanatomy, cerebellum». En: (2019).
- [85] Mario Manto et al. «Consensus paper: roles of the cerebellum in motor control—the diversity of ideas on cerebellar involvement in movement». En: *The Cerebellum* 11 (2012), págs. 457-487.
- [86] EGIDIO D'ANGELO. «Neural circuits of the cerebellum: hypothesis for function». En: *Journal of Integrative Neuroscience* 10.03 (2011), págs. 317-352.

- [87] Daniel M Wolpert, R Chris Miall y Mitsuo Kawato. «Internal models in the cerebellum». En: Trends in cognitive sciences 2.9 (1998), págs. 338-347.
- [88] Jonathan Graff-Radford. «Cerebellum | Mayo Clinic Neurology Board Review | Oxford Academic». En: Oxford Academic (). URL: <https://academic.oup.com/mcn/article-abstract/doi/10.1093/med/9780190244897.003.0015/2552744>.
- [89] Marc A Sommer. «The role of the thalamus in motor control». En: Current opinion in neurobiology 13.6 (2003), págs. 663-670.
- [90] J Massion y L Rispal-Padel. «The thalamus: motor functions». En: Revue Neurologique 142.4 (1986), págs. 327-336.
- [91] Clement Hamani, Jonathan O Dostrovsky y Andres M Lozano. «The motor thalamus in neurosurgery». En: Neurosurgery 58.1 (2006), págs. 146-158.
- [92] RW Guillery y SM Sherman. «The thalamus as a monitor of motor outputs». En: Philosophical Transactions of the Royal Society of London, Series B. 357.1428 (2002), págs. 1809-1821.
- [93] Robin A Hurley et al. «The brainstem: anatomy, assessment, and clinical syndromes». En: The Journal of neuropsychiatry and clinical neurosciences 22.1 (2010), págs. iv-7.
- [94] Dale Purves et al. «Motor Control Centers in the Brainstem: Upper Motor Neurons That Maintain Balance and Posture». En: Sinauer Associates: Sunderland, MA, USA (2001).
- [95] J.L Wilkinson. «Brainstem - an overview | ScienceDirect Topics». En: ScienceDirect (2023). URL: <https://www.sciencedirect.com/science/article/abs/pii/B9780750614474500098>.
- [96] Trevor Drew, Stephen Prentice y Bénédicte Schepens. «Cortical and brainstem control of locomotion». En: Progress in brain research 143 (2004), págs. 251-261.
- [97] Daniel Dautan et al. «Modulation of motor behavior by the mesencephalic locomotor region». En: Cell reports 36.8 (2021).
- [98] Dimitri Ryczko. «The Mesencephalic Locomotor Region: Multiple Cell Types, Multiple Behavioral Roles, and Multiple Implications for Disease». En: The Neuroscientist (2022), pág. 10738584221139136.
- [99] ZL Mel'Nikova. «Connections of the subthalamic and mesencephalic "locomotor regions" in rats». En: Neurophysiology 9.3 (1977), págs. 214-218.

Referencias

- [100] Sengul y Watson. «Extrapyramidal System - an overview». En: Science-Direct Topics (2014). URL: <https://www.sciencedirect.com/topics/neuroscience/extrapyramidal-system>.
- [101] Ricardo de Oliveira-Souza. «The human extrapyramidal system». En: Medical hypotheses 79.6 (2012), págs. 843-852.
- [102] Michaela Harrow-Mortelliti, Vamsi Reddy y George Jimshelishvili. «Physiology, spinal cord». En: StatPearls [Internet]. StatPearls Publishing, 2023.
- [103] Lingyan Xing et al. Spinal cord development and neural regeneration. Frontiers Media SA, 2023.
- [104] Emilio Bizzi et al. «New perspectives on spinal motor systems». En: Nature Reviews Neuroscience 1.2 (2000), págs. 101-108.
- [105] Rachel E Noto, Logan Leavitt y Mary Ann Edens. «Physiology, Muscle». En: (2018).
- [106] Kavitha Mukund y Shankar Subramaniam. «Skeletal muscle: A review of molecular structure and function, in health and disease». En: Wiley Interdisciplinary Reviews: Systems Biology and Medicine 12.1 (2020), e1462.
- [107] Andrew D Nordin et al. «Biomechanics and neural control of movement, 20 years later: what have we learned and what has changed?» En: Journal of neuroengineering and rehabilitation 14 (2017), págs. 1-11.
- [108] Jennifer N Kerkman et al. «Network structure of the human musculoskeletal system shapes neural interactions on multiple time scales». En: Science advances 4.6 (2018), eaat0497.
- [109] Kyeong Min Moon et al. «Proprioception, the regulator of motor function». En: BMB reports 54.8 (2021), pág. 393.
- [110] Lorenzo Vannucci, Egidio Falotico y Cecilia Laschi. «Proprioceptive feedback through a neuromorphic muscle spindle model». En: Frontiers in Neuroscience 11 (2017), pág. 341.
- [111] Kian Jalaleddini et al. «Neuromorphic meets neuromechanics, part II: the role of fusimotor drive». En: Journal of neural engineering 14.2 (2017), pág. 025002.
- [112] Neurona. Wikipedia. Consultado el 12 de febrero de 2024. URL: <https://ast.wikipedia.org/wiki/Neurona>.
- [113] Carsten Duch y Stefanie Ryglewski. «Structure and function of neuronal dendrites». En: e-Neuroforum 22.4 (2016), págs. 71-81.
- [114] Greg J Stuart y Nelson Spruston. «Dendritic integration: 60 years of progress». En: Nature neuroscience 18.12 (2015), págs. 1713-1721.

- [115] Fabian A Mikulasch, Lucas Rudelt y Viola Priesemann. «Local dendritic balance enables learning of efficient representations in networks of spiking neurons». En: *Proceedings of the National Academy of Sciences* 118.50 (2021), e2021925118.
- [116] Lukas Fischer et al. «Dendritic Mechanisms for In Vivo Neural Computations and Behavior». En: *Journal of Neuroscience* 42.45 (2022), págs. 8460-8467.
- [117] Michael London y Michael Häusser. «Dendritic computation». En: *Annu. Rev. Neurosci.* 28 (2005), págs. 503-532.
- [118] Jyotibha Acharya et al. «Dendritic computing: branching deeper into machine learning». En: *Neuroscience* 489 (2022), págs. 275-289.
- [119] Anna L Vlasits et al. «A role for synaptic input distribution in a dendritic computation of motion direction in the retina». En: *Neuron* 89.6 (2016), págs. 1317-1330.
- [120] Parker E Ludwig, Vamsi Reddy y Matthew Varacallo. «Neuroanatomy, neurons». En: (2017).
- [121] Shira Sardi et al. «New types of experiments reveal that a neuron functions as multiple independent threshold units». En: *Scientific reports* 7.1 (2017), pág. 18036.
- [122] Jennifer Bourne y Kristen M Harris. «Do thin spines learn to be mushroom spines that remember?» En: *Current opinion in neurobiology* 17.3 (2007), págs. 381-386.
- [123] Alexander J Lingley et al. «Mapping of neuron soma size as an effective approach to delineate differences between neural populations». En: *Journal of Neuroscience Methods* 304 (2018), págs. 126-135.
- [124] Gaynor Smith et al. «How neurons maintain their axons long-term: an integrated view of axon biology and pathology». En: *Frontiers in Neuroscience* 17 (2023).
- [125] Max Hofmann et al. «Cytoskeletal assembly in axonal outgrowth and regeneration analyzed on the nanoscale». En: *Scientific Reports* 12.1 (2022), pág. 14387.
- [126] Franziska Fröb y Michael Wegner. «Coordination of Schwann cell myelination and node formation at the transcriptional level». En: *Neural Regeneration Research* 17.6 (2022), pág. 1269.
- [127] Elise LV Malavasi et al. «Dynamic early clusters of nodal proteins contribute to node of Ranvier assembly during myelination of peripheral neurons». En: *Elife* 10 (2021), e68089.

Referencias

- [128] Aakash Patel et al. «Myelination and node of Ranvier formation in a human motoneuron--Schwann cell serum-free coculture». En: *ACS Chemical Neuroscience* 11.17 (2020), págs. 2615-2623.
- [129] Khan Academy. *Función y estructura de la neurona* (artículo). Accessed: 26/10/2023. 2023. URL: <https://es.khanacademy.org/science/biology/human-biology/neuron-nervous-system/a/overview-of-neuron-structure-and-function>.
- [130] Michael J Caire, Vamsi Reddy y Matthew Varacallo. «Physiology, synapse». En: (2018).
- [131] Alberto E Pereda. «Electrical synapses and their functional interactions with chemical synapses». En: *Nature Reviews Neuroscience* 15.4 (2014), págs. 250-263.
- [132] Ko-Chung Tseng, Alice C Parker y Jonathan Joshi. «A directionally-selective neuromorphic circuit based on reciprocal synapses in Starburst Amacrine Cells». En: *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE. 2011, págs. 5674-5677.
- [133] Abdulkadir Hassen Ali. «Neuromorphic computing for GABA detection». Tesis de mtría. 2023.
- [134] Jun Wang, Gert Cauwenberghs y Frédéric D Broccard. «Neuromorphic dynamical synapses with reconfigurable voltage-gated kinetics». En: *IEEE Transactions on Biomedical Engineering* 67.7 (2019), págs. 1831-1840.
- [135] Günther Sperk et al. «GABA and its receptors in epilepsy». En: *Recent advances in epilepsy research* (2004), págs. 92-103.
- [136] Bingjie Han, Andrew Bellemer y Michael R Koelle. «An evolutionarily conserved switch in response to GABA affects development and behavior of the locomotor circuit of *Caenorhabditis elegans*». En: *Genetics* 199.4 (2015), págs. 1159-1172.
- [137] Jørn Arnt y Jørgen Scheel-Krüger. «GABA in the ventral tegmental area: differential regional effects on locomotion, aggression and food intake after microinjection of GABA agonists and antagonists». En: *Life sciences* 25.15 (1979), págs. 1351-1360.
- [138] Adel Wassef, Jeffrey Baker y Lisa D Kochan. «GABA and schizophrenia: a review of basic science and clinical studies». En: *Journal of clinical psychopharmacology* 23.6 (2003), págs. 601-640.
- [139] R Bruce Lydiard. «The role of GABA in anxiety disorders». En: *Journal of Clinical Psychiatry* 64 (2003), págs. 21-27.

- [140] Richard AE Edden et al. «Reduced GABA concentration in attention-deficit/hyperactivity disorder». En: Archives of general psychiatry 69.7 (2012), págs. 750-753.
- [141] Mitchell J Vaughn y Julie S Haas. «On the diverse functions of electrical synapses». En: Frontiers in Cellular Neuroscience 16 (2022), pág. 910015.
- [142] Dale Purves et al. «Electrical Synapses». En: Neuroscience,(Sinauer Associates) (2001).
- [143] E Anne Martin, Abagael M Lasseigne y Adam C Miller. «Understanding the molecular and cell biological mechanisms of electrical synapse formation». En: Frontiers in neuroanatomy 14 (2020), pág. 12.
- [144] Virginia M Pickel y Menahem Segal. The synapse: structure and function. Elsevier, 2013.
- [145] Tyler Junior College J. Gordon Betts. 12.5 Communication Between Neurons. Accessed: 2023-10-27. 2023. URL: <https://openstax.org/books/anatomy-and-physiology/pages/12-5-communication-between-neurons>.
- [146] David Ferster y Bharathi Jagadeesh. «EPSP-IPSP interactions in cat visual cortex studied with in vivo whole-cell patch recording». En: Journal of Neuroscience 12.4 (1992), págs. 1262-1274.
- [147] Kashu Yamazaki et al. «Spiking neural networks and their applications: A Review». En: Brain Sciences 12.7 (2022), pág. 863.
- [148] Alan L Hodgkin y Andrew F Huxley. «A quantitative description of membrane current and its application to conduction and excitation in nerve». En: The Journal of physiology 117.4 (1952), pág. 500.
- [149] Mark Nelson y John Rinzel. «The hodgkin-huxley model». En: The book of genesis (1995), págs. 29-49.
- [150] Eugene M Izhikevich. «Which model to use for cortical spiking neurons?» En: IEEE transactions on neural networks 15.5 (2004), págs. 1063-1070.
- [151] Eugene M Izhikevich. Dynamical systems in neuroscience. MIT press, 2007.
- [152] Romain Brette y Wulfram Gerstner. «Adaptive exponential integrate-and-fire model as an effective description of neuronal activity». En: Journal of neurophysiology 94.5 (2005), págs. 3637-3642.
- [153] Scott L Hooper. «Central pattern generators». En: Current Biology 10.5 (2000), R176-R179.

Referencias

- [154] Arthur D Kuo. «The relative roles of feedforward and feedback in the control of rhythmic movements». En: *Motor control* 6.2 (2002), págs. 129-145.
- [155] Rafael Yuste et al. «The cortex as a central pattern generator». En: *Nature Reviews Neuroscience* 6.6 (2005), págs. 477-483.
- [156] ANDREW D McClellan y WOOCHAN Jang. «Mechanosensory inputs to the central pattern generators for locomotion in the lamprey spinal cord: re-setting, entrainment, and computer modeling». En: *Journal of Neurophysiology* 70.6 (1993), págs. 2442-2454.
- [157] Serge Rossignol. «Locomotion and its recovery after spinal injury». En: *Current opinion in neurobiology* 10.6 (2000), págs. 708-716.
- [158] Ole Kiehn y Simon JB Butt. «Physiological, anatomical and genetic identification of CPG neurons in the developing mammalian spinal cord». En: *Progress in neurobiology* 70.4 (2003), págs. 347-361.
- [159] Julia T Choi y Amy J Bastian. «Adaptation reveals independent control networks for human walking». En: *Nature neuroscience* 10.8 (2007), pág. 1055.
- [160] T Graham Brown. «On the nature of the fundamental activity of the nervous centres; together with an analysis of the conditioning of rhythmic activity in progression, and a theory of the evolution of function in the nervous system». En: *The Journal of physiology* 48.1 (1914), págs. 18-46.
- [161] Duc Trong Tran et al. «Central pattern generator based reflexive control of quadruped walking robots using a recurrent neural network». En: *Robotics and autonomous systems* 62.10 (2014), págs. 1497-1516.
- [162] Payam Atoofi, Fred H Hamker y John Nassour. «Learning of central pattern generator coordination in robot drawing». En: *Frontiers in Neurobotics* (2018), pág. 44.
- [163] Ioannis Polykretis, Guangzhi Tang y Konstantinos P Michmizos. «An astrocyte-modulated neuromorphic central pattern generator for hexapod robot locomotion on intel's Loihi». En: *International Conference on Neuromorphic Systems 2020*. 2020, págs. 1-9.
- [164] Emmanouil Angelidis et al. «A spiking central pattern generator for the control of a simulated lamprey robot running on SpiNNaker and Loihi neuromorphic boards». En: *Neuromorphic Computing and Engineering* 1.1 (2021), pág. 014005.

- [165] Yu A Tsybina et al. «Toward biomorphic robotics: A review on swimming central pattern generators». En: *Chaos, Solitons & Fractals* 165 (2022), pág. 112864.
- [166] Daniel Gutierrez-Galan et al. «NeuroPod: a real-time neuromorphic spiking CPG applied to robotics». En: *Neurocomputing* 381 (2020), págs. 10-19.
- [167] Rodney J Douglas y Kevan AC Martin. «Recurrent neuronal circuits in the neocortex». En: *Current biology* 17.13 (2007), R496-R500.
- [168] Rodney J Douglas y Kevan AC Martin. «Neuronal circuits of the neocortex». En: *Annu. Rev. Neurosci.* 27 (2004), págs. 419-451.
- [169] Matthias Oster y Shih-Chii Liu. «Spiking inputs to a winner-take-all network». En: *Advances in Neural Information Processing Systems* 18 (2005).
- [170] Wolfgang Maass. «On the computational power of winner-take-all». En: *Neural computation* 12.11 (2000), págs. 2519-2535.
- [171] Matthias Oster, Rodney Douglas y Shih-Chii Liu. «Computation with spikes in a winner-take-all network». En: *Neural computation* 21.9 (2009), págs. 2437-2465.
- [172] Hugh G Manning et al. «Emergence of winner-takes-all connectivity paths in random nanowire networks». En: *Nature communications* 9.1 (2018), pág. 3219.
- [173] Elisabetta Chicca et al. «Neuromorphic electronic circuits for building autonomous cognitive systems». En: *Proceedings of the IEEE* 102.9 (2014), págs. 1367-1388.
- [174] Chiara Bartolozzi, Giacomo Indiveri y Elisa Donati. «Embodied neuromorphic intelligence». En: *Nature communications* 13.1 (2022), pág. 1024.
- [175] Trevor Bekolay et al. «Nengo: a Python tool for building large-scale functional brain models». En: *Frontiers in neuroinformatics* 7 (2014), pág. 48.
- [176] Marc-Oliver Gewaltig y Markus Diesmann. «Nest (neural simulation tool)». En: *Scholarpedia* 2.4 (2007), pág. 1430.
- [177] Marcel Stimberg, Romain Brette y Dan FM Goodman. «Brian 2, an intuitive and efficient neural simulator». En: *Elife* 8 (2019), e47314.
- [178] Michael L Hines y Nicholas T Carnevale. «NEURON: a tool for neuroscientists». En: *The neuroscientist* 7.2 (2001), págs. 123-135.

Referencias

- [179] Tobi Delbruck et al. «Frame-free dynamic digital vision». En: *Secure-Life Electronics, Advanced Electronics for Quality Life*. Vol. 1. Citeseer. 2008, págs. 21-26.
- [180] Juan P Dominguez-Morales et al. «Navis: neuromorphic auditory visualizer tool». En: *Neurocomputing* 237 (2017), págs. 418-422.
- [181] Neuromorphic Engineering Group, University of Seville. AER Tools. <https://www.rtc.us.es/neuromorphic-engineering/aertools/>. Accedido: 29/02/2024.
- [182] Nicholas T Carnevale y Michael L Hines. *The NEURON book*. Cambridge University Press, 2006.
- [183] Francesco Carlo Morabito, Andreas G Andreou y Elisabetta Chicca. «Neuromorphic engineering: from neural systems to brain-like engineered systems». En: *Neural Networks* 45 (2013).
- [184] Leslie S Smith. «Neuromorphic systems: Past, present and future». En: *Brain Inspired Cognitive Systems 2008* (2010), págs. 167-182.
- [185] Valentino Braitenberg et al. «Cortical architectonics». En: *Cortex: Statistics and geometry of neuronal connectivity* (1998), págs. 135-137.
- [186] Eric R Kandel et al. *Principles of neural science*. Vol. 4. McGraw-hill New York, 2000.
- [187] Mark Bear, Barry Connors y Michael A Paradiso. *Neuroscience: exploring the brain, enhanced edition: exploring the brain*. Jones & Bartlett Learning, 2020.
- [188] Massimo Antonio Sivilotti. *Wiring considerations in analog VLSI systems, with application to field-programmable networks*. California Institute of Technology, 1991.
- [189] John Lazzaro et al. «Silicon auditory processors as computer peripherals». En: *Advances in Neural Information Processing Systems* 5 (1992).
- [190] John Lazzaro y John Wawrzynek. «A multi-sender asynchronous extension to the AER protocol». En: *Proceedings Sixteenth Conference on Advanced Research in VLSI*. IEEE. 1995, págs. 158-169.
- [191] Kwabena A Boahen. «Point-to-point connectivity between neuromorphic chips using address events». En: *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing* 47.5 (2000), págs. 416-434.
- [192] Misha Mahowald. «VLSI analogs of neuronal visual processing: a synthesis of form and function». En: (1992).

- [193] Rafael Serrano-Gotarredona et al. «CAVIAR: A 45k neuron, 5M synapse, 12G connects/s AER hardware sensory--processing--learning--actuating system for high-speed visual object recognition and tracking». En: IEEE Transactions on Neural networks 20.9 (2009), págs. 1417-1438.
- [194] ROBOTIS. AX-12A. <https://emanual.robotis.com/docs/en/dxl/ax/ax-12a/>. Último acceso: enero de 2024.
- [195] R Bautista-Quintero, F Aguirre-Cerrillo, JA Ramos-Andrade et al. «Diseño, Construcción y Control de un Robot Cuadrúpedo con actuadores de alto desempeño conectados en red». En: 10º Congreso Nacional de Mecatrónica. 2011.
- [196] Alexis Sánchez et al. «Design and construction of an anthropomorphic robotic arm of seven degrees of freedom with kinematic and dynamic analysis based on genetic algorithms». En: 2016 IEEE International Conference on Automatica (ICA-ACCA). IEEE. 2016, págs. 1-8.
- [197] SN74LS241 | Texas Instruments. [Online; accessed 4-January-2024]. URL: https://www.ti.com/product/SN74LS241?utm_source=supplyframe&utm_medium=SEP&utm_campaign=not_alldatasheet&DCM=yes&dclid=CKq74aTMw4MDFSxGHQkd4zMBFg.
- [198] The SpiNNaker Project. SpiNNaker -- A Massively-Parallel Architecture for Neuromorphic Computing. <https://apt.cs.manchester.ac.uk/projects/SpiNNaker/SpiNNchip/>. Accessed: February 13, 2024.
- [199] Shreejoy J Tripathy et al. «NeuroElectro: a window to the world's neuron electrophysiology data». En: Frontiers in neuroinformatics 8 (2014), pág. 40.
- [200] RD Clarke. «An application of the Poisson distribution». En: Journal of the Institute of Actuaries 72.3 (1946), págs. 481-481.
- [201] Digilent. Nexys 4 DDR Reference Manual. <https://digilent.com/reference/programmable-logic/nexys-4-ddr/reference-manual>. Visitado el 8 de septiembre de 2023.
- [202] Luis A Plana et al. «spiNNlink: FPGA-Based Interconnect for the Million-Core SpiNNaker System». En: IEEE Access 8 (2020), págs. 84918-84928.
- [203] Elliott J Bayly. «Spectral analysis of pulse frequency modulation in the nervous systems». En: IEEE Transactions on Biomedical Engineering 4 (1968), págs. 257-265.
- [204] RW Jones et al. «Pulse modulation in physiological systems, phenomenological aspects». En: IRE Transactions on Bio-Medical Electronics 8.1 (1961), págs. 59-67.

Referencias

- [205] John E Hall y Michael E Hall. Guyton and Hall textbook of medical physiology e-Book. Elsevier Health Sciences, 2020.
- [206] William F Ganong. «Review of medical physiology». En: Dynamics of blood and lymph flow 30 (1995), págs. 525-541.
- [207] Fernando Pérez Peña. Neuromorphic adaptive spiking CPG towards bio-inspired locomotion. <https://www.youtube.com/watch?v=3s89p3qYnCU>. [Fecha de acceso: 26 01, 2024]. 2022.
- [208] Marilyn MacKay-Lyons. «Central pattern generation of locomotion: a review of the evidence». En: Physical therapy 82.1 (2002), págs. 69-83.
- [209] Sten Grillner y Peter Wallen. «Central pattern generators for locomotion, with special reference to vertebrates». En: Annual review of neuroscience 8.1 (1985), págs. 233-261.
- [210] Eve Marder y Dirk Bucher. «Central pattern generators and the control of rhythmic movements». En: Current biology 11.23 (2001), R986-R996.
- [211] Beck Strohmer, Poramate Manoonpong y Leon Bonde Larsen. «Flexible spiking cpgs for online manipulation during hexapod walking». En: Frontiers in neurorobotics 14 (2020), págs. 41.
- [212] Beck Strohmer et al. «Integrating non-spiking interneurons in spiking neural networks». En: Frontiers in neuroscience 15 (2021), págs. 633945.
- [213] Alex Spaeth et al. «Neuromorphic closed-loop control of a flexible modular robot by a simulated spiking central pattern generator». En: 2020 3rd IEEE International Conference on Soft Robotics (RoboSoft). IEEE. 2020, págs. 46-51.
- [214] Horacio Rostro-Gonzalez et al. «A CPG system based on spiking neurons for hexapod robot locomotion». En: Neurocomputing 170 (2015), págs. 47-54.