

UNIVERSIDAD DE SANTIAGO DE CHILE
FACULTAD DE INGENIERÍA
Departamento de Ingeniería Informática



**Optimización de parámetros del modelo MRG de fibra nerviosa para su uso
en simulación de fibras sensoriales**

Arturo Antonio Salinas González

Profesor guía: Leonel Medina Daza

Trabajo de titulación en conformidad a los
requisitos para obtener el título de Ingeniero
Civil en Informática

Santiago – Chile
2020

© **Arturo Antonio Salinas González** , 2020



• Algunos derechos reservados. Esta obra está bajo una Licencia Creative Commons Atribución-NoComercial-CompartirIgual 3.0. Sus condiciones de uso pueden ser revisadas en: <http://creativecommons.org/licenses/by/3.0/cl/>.

RESUMEN

La estimulación de la médula espinal, abreviado como *SCS* representa un método para el tratamiento del dolor neuropático siendo utilizado desde hace más de 40 años, pero todavía siendo investigado, estudiando nuevas formas de aplicarlo como puede ser mediante el uso de frecuencias sobre el orden de los kilohertz.

En el campo de la neurociencia, el uso de modelos es crucial para el entendimiento del sistema nervioso y lo observado en resultados experimentales. El modelo *MRG* de fibra nerviosa realizado por McIntyre, Richardson y Grill el 2002 es uno de tantos utilizado en el área. No obstante, al momento realizar simulaciones de experimentos donde se aplica *SCS* sobre roedores, los resultados obtenidos no se ajustan a lo visto experimentalmente. Dado esto, el enfoque de este trabajo es la aplicación de un método de optimización con base a una metahurística, particularmente una estrategia evolutiva mediante adaptación de matrices de covarianza o *CMA-ES*, con tal de encontrar un conjunto de parámetros del modelo *MRG* que permitan obtener simulaciones que se acerquen en mayor medida a lo visto experimentalmente.

Los resultados obtenidos para ambos experimentos se analizan y se concluye que si bien, el método de optimización *CMA-ES* logra minimizar la función objetivo dada, al aplicar los parámetros obtenidos sobre el modelo no se logran obtener simulaciones del experimento que se acerquen de mejor manera a lo esperado, por lo que se plantea utilizar una función objetivo diferente. No obstante, observando los resultados de las simulaciones para los conjuntos de parámetros obtenidos se logra observar el efecto de variar particularmente ciertos parámetros, lo cual puede ser un indicador de sobre cuales se debe trabajar para lograr obtener un mejor ajuste del modelo.

Palabras Claves: Estimulación de la médula espinal, Neurociencia computacional, Optimización de parámetros, Estrategias evolutivas, *CMA-ES*

*Take it back
Within your hands
The name you used to have*

AGRADECIMIENTOS

Agradezco en primer lugar a mi familia, por soportarme tantos años de pasar por tantas universidades antes de lograr llegar acá. Agradezco también a mi pareja, por todos los años que ha estado presente. A mi perra Candy, por todo lo que me ha acompañado. A mis amigos, por hacer que el tiempo en la universidad fuese mucho más tolerable, y por las pruebas que me recordaron tenía que rendir, e informes que entregar.

Quiero agradecer a varias personas del departamento de informática, a sus profesores en particular por la mayoría ser mucho mejores de lo que esperaba. A mi profesor guía por su apoyo en este trabajo. A la gente de soporte por aguantarme luego de que colapsara la memoria del clúster, y también a este último, por no apagarse.

Por último quiero agradecer igualmente a mi tía Gilda quien falleció este año, la cual siempre me apoyó desde el primer día del jardín, durante el colegio y desde que entré a la universidad.

Este trabajo fue financiado parcialmente por ANID - Fondecyt 11190822 y ANID - Iniciativa Científica Milenio, código NCN19_161

TABLA DE CONTENIDO

1	Introducción	1
1.1	Antecedentes y Motivación	1
1.2	Descripción del problema	3
1.2.1	Brecha en el conocimiento	3
1.3	Solución propuesta	5
1.3.1	Hipótesis de trabajo	5
1.4	Objetivos y alcances del proyecto	5
1.4.1	Objetivo general	5
1.4.2	Objetivos específicos	6
1.4.3	Alcances	6
1.5	Metodologías y herramientas utilizadas	7
1.5.1	Ambiente experimental	7
1.5.2	Procedimiento experimental	7
1.5.3	Descripción de los datos o corpus	10
1.5.4	Software y hardware requerido	12
1.6	Organización del documento	12
2	Marco Teórico	14
2.1	Sistema nervioso y modelado computacional	14
2.2	Estimulación de la médula espinal (SCS)	19
2.3	Métodos de optimización	20
2.3.1	Estrategia evolutiva mediante adaptación de matrices de covarianza (CMA-ES)	22
2.4	Herramientas de análisis	24
2.4.1	Histogramas y funciones de densidad de probabilidad	24
3	Estado del arte	28
3.1	Estimulación de la médula espinal	28
3.2	Modelo de fibra nerviosa de McIntyre, Richardson y Grill (MRG)	29
3.3	Uso del modelo de fibra nerviosa MRG en literatura	32
3.4	Optimización de parámetros en modelos de fibra nerviosa	33
4	Desarrollo de la investigación	36
4.1	Estructura del software experimental desarrollado	36
4.1.1	Modelo y simulador	37
4.1.2	Implementación del optimizador	40
4.1.3	Pruebas del optimizador con funciones conocidas	43
4.1.4	Ejecución del algoritmo en clúster	45
4.2	Resumen	45
5	Resultados experimentales	47
5.1	Pruebas del software usando funciones conocidas	47
5.1.1	Función Rosenbrock	47
5.1.2	Función Rastrigin	50
5.2	Proceso de optimización	53
5.3	Análisis de soluciones obtenidas	57
5.4	Análisis de sensibilidad	69
5.4.1	Parámetros electrofisiológicos	69
5.4.2	Parámetros morfológicos	75
6	Conclusiones	78

6.1	Desarrollo experimental	78
6.2	Resultados	79
6.2.1	Objetivos e hipótesis	81
6.3	Trabajo a futuro	82
6.4	Observaciones finales	84
Glosario		85
Referencias bibliográficas		86
Apéndices		89
A	Gráficos de respuesta eléctrica adicionales	89
A.1	Parámetros base	89
A.2	Parámetros obtenidos mediante CMA-ES aplicado solo a parámetros electrofisiológicos	91
A.3	Parámetros obtenidos mediante CMA-ES aplicado a parámetros electrofisiológicos y morfológicos	93
B	Gráficos de análisis de sensibilidad de respuesta eléctrica adicionales	95

ÍNDICE DE TABLAS

Tabla 1.1	Resumen de variables electrofisiológicos a optimizar con restricciones. . . .	10
Tabla 1.2	Funciones de extrapolación de parámetros morfológicos con constantes iniciales.	11
Tabla 1.3	Rangos posibles a tomar por los valores A y B para cada función correspondiente a un parámetro del modelo MRG.	11
Tabla 5.1	Valores obtenidos utilizando CMA-ES aplicados solo sobre parámetros electrofisiológicos comparados con los valores base del modelo.	62
Tabla 5.2	Valores obtenidos utilizando CMA-ES aplicados a parámetros electrofisi- ológicos y morfológicos del modelo	65

ÍNDICE DE FIGURAS

Figura 1.1	Porcentaje de sujetos categorizados por mejora del dolor luego de 12 meses de tratamiento con SCS tradicional y SCS a 10kHz. Evaluación por paciente (A) y por médico tratante (B).	2
Figura 1.2	Ejemplo de señales de pulsos eléctricos utilizados en SCS, mostrando distintos parámetros usados en su definición.	2
Figura 1.3	Comparación de patrones de activación en respuesta a SCS bifásico de 10-kHz (sin bloqueo de conducción) y tasa promedio de activación general, obtenidos de forma experimental y mediante simulaciones utilizando el modelo MRG.	4
Figura 2.1	Diagrama de las regiones en médula espinal y nervios principales en sus respectivas salidas entre las vértebras.	15
Figura 2.2	Diagrama de ejemplificación de un trozo de membrana de fibra nerviosa modelado como un circuito eléctrico.	16
Figura 2.3	Diagrama de ejemplo del modelado compartimental de una fibra nerviosa.	17
Figura 2.4	Diagrama del potencial de acción (a) y cambios en la permeabilidad de la membrana (b).	17
Figura 2.5	Diagrama de propagación del potencial eléctrico en una fibra nerviosa.	18
Figura 2.6	Diagrama de la inserción de electrodos en el espacio epidural de la columna.	19
Figura 2.7	Ejemplo de histograma describiendo frecuencia según peso.	25
Figura 2.8	Principios para la determinación de un gráfico PSTH en base a la medición repetida de un estímulo.	26
Figura 3.1	Propiedades de axones de la columna dorsal registrados durante pruebas de SCS.	28
Figura 3.2	Patrones de disparo mostrados en respuesta a SCS bifásico a 10 kHz, para el caso sin efectos sobre el potencial antidrómico (A) y con respuesta de activación inicial y bloqueo de conducción (B).	29
Figura 3.3	Diagrama del modelo de fibra nerviosa MRG.	30
Figura 3.4	Parámetros morfológicos del modelo MRG. Valores expresados en μm .	31
Figura 3.5	Parámetros eléctricos del modelo MRG.	31
Figura 3.6	Diagrama del uso de PSO y TRR en forma de 2 etapas (A) e híbrida (B).	34
Figura 4.1	Diagrama de la estructura general del software desarrollado.	36
Figura 5.1	Gráfico de la mediana de valores objetivos obtenidos en cada ejecución optimizando la función de rosenbrock, respecto a la última generación realizada antes del estancamiento.	48
Figura 5.2	Gráficos de caja de la mediana de valores objetivos obtenidos en cada ejecución optimizando la función de rosenbrock, respecto a determinadas generaciones.	49
Figura 5.3	Gráficos de caja del estancamiento de la mediana de valores objetivos obtenidos en cada ejecución optimizando la función de rosenbrock, respecto a determinadas generaciones.	49
Figura 5.4	Gráfico de la mediana de valores objetivos obtenidos en cada ejecución optimizando la función de rastrigin para dos dimensiones, respecto a la última generación realizada antes del estancamiento.	50

Figura 5.5	Gráficos de caja de la mediana de valores objetivos obtenidos en cada ejecución optimizando la función de rastrigin para dos dimensiones, respecto a determinadas generaciones.	51
Figura 5.6	Gráfico de la mediana de valores objetivos obtenidos en cada ejecución optimizando la función de rastrigin para 16 dimensiones, respecto a la última generación realizada antes del estancamiento.	52
Figura 5.7	Gráficos de caja de la mediana de valores objetivos obtenidos en cada ejecución optimizando la función de rastrigin para 16 dimensiones, respecto a determinadas generaciones.	52
Figura 5.8	Gráficos de caja del estancamiento de la mediana de valores objetivos obtenidos en cada ejecución optimizando la función de rastrigin para 16 dimensiones, respecto a determinadas generaciones.	53
Figura 5.9	Análisis de la convergencia del valor objetivo, aplicando algoritmo CMA-ES sobre parámetros electrofisiológicos (Experimento 1).	54
Figura 5.10	Análisis de la convergencia del valor objetivo, aplicando algoritmo CMA-ES sobre parámetros electrofisiológicos y morfológicos (Experimento 2).	55
Figura 5.11	Análisis de cumplimiento de restricciones, por soluciones infactibles en cada generación y por parámetros promedio fuera de restricción por solución en cada generación, aplicando algoritmo CMA-ES sobre parámetros electrofisiológicos.	56
Figura 5.12	Análisis de cumplimiento de restricciones, por soluciones infactibles en cada generación y por parámetros promedio fuera de restricción por solución en cada generación, aplicando algoritmo CMA-ES sobre parámetros electrofisiológicos y morfológicos.	57
Figura 5.13	Curvas de densidad de impulsos promedio por fibra para distintas amplitudes con parámetros base en el modelo utilizado.	58
Figura 5.14	Cantidad de impulsos promedio por fibra dentro de cada segundo para distintas amplitudes usando parámetros base.	59
Figura 5.15	Respuesta eléctrica para fibras de distintos tamaños para cada rango de amplitud durante 5 segundos, usando parámetros base.	61
Figura 5.16	Curvas de densidad de impulsos promedio por fibra para distintas amplitudes usando parámetros obtenidos mediante CMA-ES solo variando parámetros electrofisiológicos, comparadas con curvas obtenidas utilizando los parámetros base.	62
Figura 5.17	Cantidad de impulsos promedio por fibra dentro de cada segundo para distintas amplitudes usando parámetros obtenidos mediante CMA-ES, solo variando parámetros electrofisiológicos.	63
Figura 5.18	Respuesta eléctrica para fibras de distintos tamaños para cada rango de amplitud durante 5 segundos, usando parámetros obtenidos mediante CMA-ES, solo variando parámetros electrofisiológicos.	64
Figura 5.19	Curvas de densidad de impulsos promedio por fibra para distintas amplitudes usando parámetros obtenidos mediante CMA-ES solo variando parámetros electrofisiológicos y morfológicos, comparadas con curvas obtenidas utilizando los parámetros base.	66
Figura 5.20	Cantidad de impulsos promedio por fibra dentro de cada segundo para distintas amplitudes usando parámetros obtenidos mediante CMA-ES, variando parámetros electrofisiológicos y morfológicos.	67
Figura 5.21	Respuesta eléctrica para fibras de distintos tamaños para cada rango de amplitud durante 5 segundos, usando parámetros obtenidos mediante CMA-ES, variando parámetros electrofisiológicos y morfológicos.	68

Figura 5.22 Respuesta eléctrica para una fibra de 3.35 micrómetros, modificando únicamente la conductancia máxima de sodio según el valor obtenido utilizando en el segundo experimento, manteniendo el resto de parámetros base.	69
Figura 5.23 Respuesta eléctrica para una fibra de 3.35 micrómetros, modificando únicamente la conductancia máxima de potasio según el valor obtenido en el segundo experimento, manteniendo el resto de parámetros base.	71
Figura 5.24 Respuesta eléctrica para una fibra de 3.35 micrómetros, modificando únicamente la conductancia máxima de sodio persistente según el valor obtenido en el segundo experimento, manteniendo el resto de parámetros base.	72
Figura 5.25 Respuesta eléctrica para una fibra de 3.35 micrómetros, modificando únicamente la conductancia MYSA según el valor obtenido utilizando CMA-ES en el segundo experimento, manteniendo el resto de parámetros base.	73
Figura 5.26 Respuesta eléctrica para una fibra de 3.35 micrómetros, modificando únicamente la conductancia MYSA según el valor obtenido utilizando CMA-ES en el primer experimento, manteniendo el resto de parámetros base.	74
Figura 5.27 Respuesta eléctrica para una fibra de 3.35 micrómetros, modificando únicamente la segunda componente del diámetro nodal según el valor obtenido utilizando CMA-ES para todos los parámetros.	75
Figura 5.28 Respuesta eléctrica para una fibra de 3.35 micrómetros, modificando únicamente la primera componente del diámetro axonal según el valor obtenido utilizando CMA-ES para todos los parámetros.	76
Figura 5.29 Respuesta eléctrica para una fibra de 3.35 micrómetros, modificando únicamente la segunda componente del diámetro axonal según el valor obtenido utilizando CMA-ES para todos los parámetros.	77
Figura A.1 Respuesta eléctrica para fibras de 3.35 micrómetros para cada rango de amplitud durante 2.8 segundos, usando parámetros base.	89
Figura A.2 Respuesta eléctrica para fibras de 3.35 micrómetros para cada rango de amplitud durante 12 segundos, usando parámetros base.	90
Figura A.3 Respuesta eléctrica para fibras de 3.35 micrómetros para cada rango de amplitud durante 2.8 segundos, usando parámetros obtenidos mediante CMA-ES aplicado únicamente a parámetros electrofisiológicos.	91
Figura A.4 Respuesta eléctrica para fibras de 3.35 micrómetros para cada rango de amplitud durante 12 segundos, usando parámetros obtenidos mediante CMA-ES aplicado únicamente a parámetros electrofisiológicos.	92
Figura A.5 Respuesta eléctrica para fibras de 3.35 micrómetros para cada rango de amplitud durante 2.8 segundos, usando parámetros obtenidos mediante CMA-ES aplicado a todo el conjunto de parámetros.	93
Figura A.6 Respuesta eléctrica para fibras de 3.35 micrómetros para cada rango de amplitud durante 12 segundos, usando parámetros obtenidos mediante CMA-ES aplicado únicamente a todo el conjunto de parámetros.	94
Figura B.1 Respuesta eléctrica para una fibra de 3.35 micrómetros, modificando únicamente la conductancia máxima de sodio según el valor obtenido utilizando CMA-ES únicamente sobre los parámetros electrofisiológicos.	95
Figura B.2 Respuesta eléctrica para una fibra de 3.35 micrómetros, modificando únicamente las dos componentes del diámetro axonal según el valor obtenido utilizando CMA-ES para todos los parámetros.	96

CAPÍTULO 1. INTRODUCCIÓN

1.1 ANTECEDENTES Y MOTIVACIÓN

La estimulación de la médula espinal (Spinal Cord Stimulation o SCS) es un método establecido para el tratamiento del dolor neuropático, donde tradicionalmente se usa un arreglo de electrodos colocados en el espacio epidural para entregar estimulación mediante series de impulsos eléctricos a baja frecuencia en la fibra nerviosa de la espina dorsal, induciendo parestesia sobreponiéndose a la distribución del dolor del paciente (Amirdelfan et al., 2018). Como terapia se utilizó por primera vez en 1967 para el tratamiento del dolor y entró en uso rutinario en la década de los 80's, siendo aprobada por la FDA (Food and Drug Administration) de los EEUU en 1989 para tratar el dolor crónico de daño nervioso a la zona del tronco, brazos o piernas. Alrededor del mundo aproximadamente unos 34,000 pacientes entran a cirugía al año para la implantación de estimuladores de la médula espinal, y la terapia actualmente corresponde al 70% tratamientos de neuromodulación (Thomson, 2019). A pesar del éxito que se ha visto en las terapias de este tipo, todavía existen amplias oportunidades de mejorar la eficacia de los tratamientos con base a estimulación de la médula espinal, donde solo un 58% de los pacientes reporta resultados exitosos, definidos como una mejora de un 50% o más del dolor reportado (Zhang et al., 2014).

En la actualidad se prueban regularmente nuevas formas de uso de SCS para distintos tipos de dolor neuropático. Amirdelfan et al. (2018) explica cómo el uso de SCS tradicional de baja frecuencia no entrega un alivio del dolor sostenido a largo plazo, y no es considerablemente efectivo para tratar dolores a la espalda baja, y muestra que el uso de SCS de alta frecuencia (10 kHz) presenta un alivio del dolor considerable en un mayor porcentaje de pacientes en comparación al SCS tradicional como se muestra en la figura 1.1, tanto en lo que percibe el mismo paciente como lo que registra el médico tratante. Por otro lado, Owusu et al. (2017) describe el uso de tratamientos de SCS con formas de onda sin parestesia, usando pulsaciones en ráfagas o de alta frecuencia en comparación a métodos tradicionales, donde dependiendo del paciente se puede preferir distintas configuraciones del tipo de onda y frecuencia.

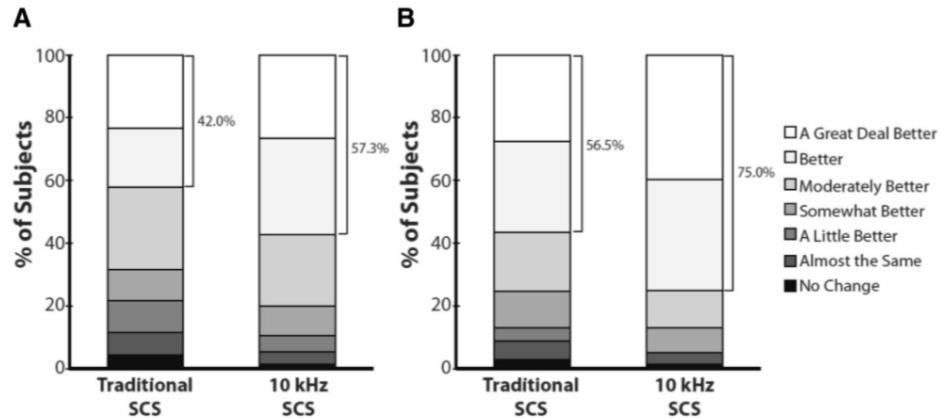


Figura 1.1: Porcentaje de sujetos categorizados por mejora del dolor luego de 12 meses de tratamiento con SCS tradicional y SCS a 10kHz. Evaluación por paciente (A) y por médico tratante (B).

Fuente: Amirdelfan et al. (2018)

Existen distintas formas de aplicar SCS con diferentes parámetros a variar en cada configuración. La figura 1.2 muestra en el primer recuadro una señal básica de pulsos definiendo la frecuencia, ancho y amplitud de cada uno, mientras que en el segundo cuadro se muestra una configuración donde se utilizan ráfagas de pulsos, considerando tanto la frecuencia entre cada pulso dentro de una ráfaga como la frecuencia entre cada una de estas (Caylor et al., 2019). Para frecuencias bajas las fibras nerviosas disparan aproximadamente un impulso nervioso o *spike*, definido por el aumento y disminución del potencial eléctrico en la fibra nerviosa, por cada pulso eléctrico dado sobre esta, mientras que para frecuencias más altas se tienen patrones de disparos irregulares o incluso en algunos casos se presenta bloqueo de conducción (Crosby et al., 2017), donde los impulsos nerviosos dejan de propagarse en la fibra.

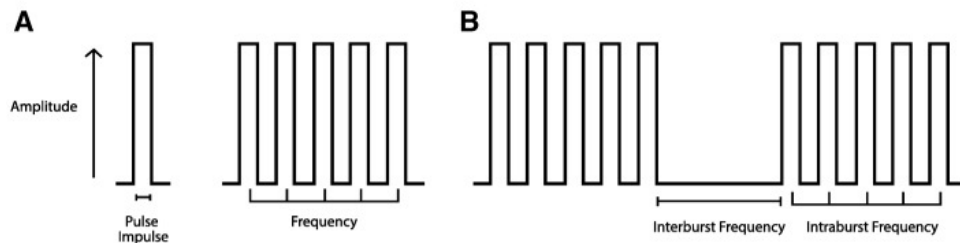


Figura 1.2: Ejemplo de señales de pulsos eléctricos utilizados en SCS, mostrando distintos parámetros usados en su definición.

Fuente: Caylor et al. (2019)

Las distintas configuraciones de SCS y los efectos que pueden tener en el alivio del dolor crónico inducían la necesidad de probar distintos parámetros al llevar a cabo la estimulación. Siendo la teoría una de las herramientas primarias de los científicos, un modelo es una estructura que nos permite representar fenómenos permitiendo mediante la interpretación de esta estructura

un mayor entendimiento del fenómeno (Levenstein et al., 2020). A modo de ejemplo, Durá et al. (2019) muestra el uso de un modelo computacional tridimensional de médula espinal desarrollado en el software de simulación de física y análisis de elementos finitos COMSOL Multiphysics, en conjunto con un modelo de fibra nerviosa mielinizada implementado en el software de cómputo numérico MATLAB basado en el modelo MRG (McIntyre et al., 2002) para probar los efectos de la polaridad de los electrodos en terapia mediante SCS. Este último modelo en el que se basó el estudio mencionado ha sido utilizado para simular distintos tipos de fibra nerviosa como se muestra en Gaines et al. (2018) para modelar axones motores y sensoriales.

No obstante, el modelo MRG y otros modelos de fibra nerviosa si bien pueden ser utilizados para representar distintos tipos de fibras, puede que no representen adecuadamente fibras particulares del cuerpo cuyas propiedades geométricas y eléctricas pueden diferir entre individuos y las mismas fibras nerviosas de este. Al simular con el modelo MRG la respuesta ante SCS de un tipo de axón sensorial de la columna dorsal los resultados obtenidos no coinciden con las mediciones experimentales obtenidas de Crosby et al. (2017), planteando la pregunta de si existe un conjunto de parámetros que permitan obtener una simulación que se ajuste mejor a los datos experimentales, y cómo obtener estos parámetros.

1.2 DESCRIPCIÓN DEL PROBLEMA

1.2.1 Brecha en el conocimiento

Crosby et al. (2017) muestran un estudio de la actividad y conductividad de fibras nerviosas singulares de la columna dorsal ante SCS con frecuencia del orden de los kilohertz, basado en experimentos sobre ratas anestesiadas. A modo de ejemplo, en la parte izquierda de la figura 1.3 se muestra en la parte superior un patrón de activación (disparo) en respuesta a estimulación bifásica a 10 kHz , particularmente un caso donde el SCS (durante los primeros segundos) no produce un bloqueo de conducción y pasado este las puntas antidrómicas continúan activándose. En la parte inferior se muestra la densidad de impulsos registrados por segundo, o tasa de disparo promedio para distintos ejemplares de fibras nerviosas durante SCS a lo largo del tiempo, indicando un incremento y rápida disminución de esta tasa durante el comienzo, la cual se pronuncia en mayor medida según la amplitud se acerca al umbral motor (MT) promedio, correspondiente a la intensidad mínima necesaria para inducir movimiento motor (McConnell et al., 2001).

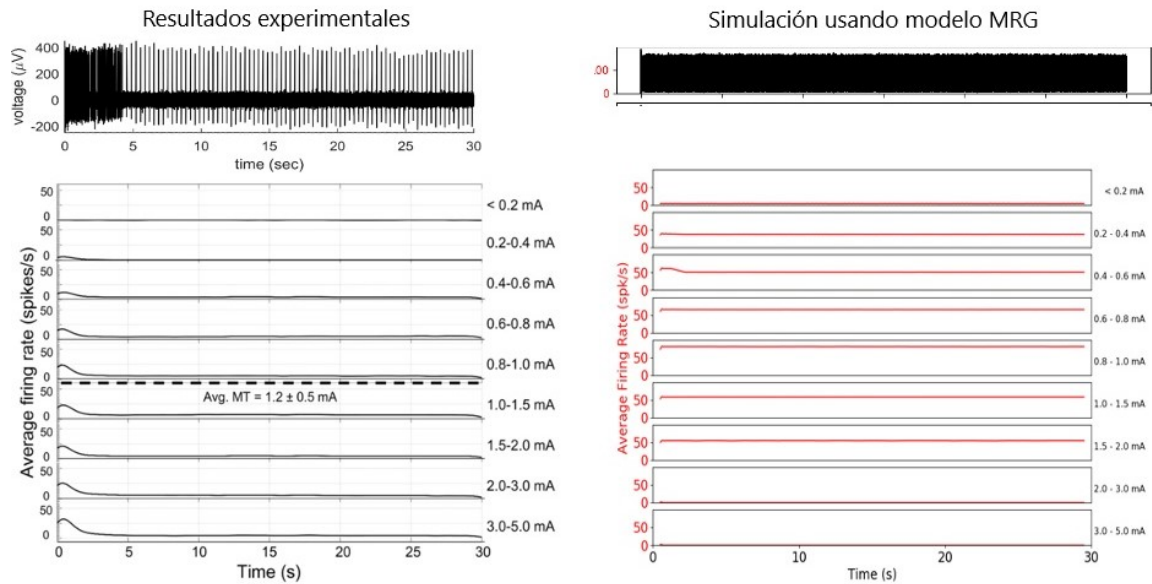


Figura 1.3: Comparación de patrones de activación en respuesta a SCS bifásico de 10-kHz (sin bloqueo de conducción) y tasa promedio de activación general, obtenidos de forma experimental y mediante simulaciones utilizando el modelo MRG.

Fuente: Crosby et al. (2017), Medina (2020)

En una simulación de estos experimentos realizada por Medina (2020), utilizando el modelo MRG de fibra nerviosa, algunos de los resultados obtenidos presentes en la parte derecha de la figura 1.3 muestran patrones de activación diferentes a los obtenidos experimentalmente, en la parte superior mostrando un caso donde no hubo ni bloqueo de conducción ni detención del SCS, mostrando una saturación constante de puntas, y en la parte inferior se muestra como las curvas del promedio de tasa de disparo son distintas a las esperadas. En particular, se observa que para un amplio rango de amplitudes de estimulación, el modelo MRG muestra tasas constantes de disparo, sin capturar la tasa decreciente observada experimentalmente.

En la literatura actual lo más cercano que se tiene a encontrar nuevos parámetros para este modelo es lo visto en Durá et al. (2019) donde se modificaron algunos parámetros eléctricos, pero el tipo de fibra nerviosa que se estudia en aquel caso corresponde al de fibras nerviosas sensoriales grandes de la raíz y columna dorsal que superan los $12.8 \mu m$ de diámetro, mientras que en este caso se tienen axones singulares de la columna dorsal de en promedio $3.3 \mu m$ de diámetro. Por otro lado, en Durá et al. (2019) el modelo MRG se usa únicamente para probar el potencial de membrana, y no se estudian patrones de activación u otros experimentos realizados en Crosby et al. (2017). Siguiendo esto, a la fecha no existe literatura reciente donde se estudie la aplicación de algoritmos de optimización para encontrar parámetros para el modelo MRG en este tipo de fibra nerviosa, pero si aplicado en otros modelos de distintas fibras como se muestra en Xu et al. (2019) utilizando algoritmos evolutivos.

1.3 SOLUCIÓN PROPUESTA

Dado los problemas expuestos, se propone como solución la aplicación de un algoritmo de optimización para encontrar un conjunto de parámetros que adapte el modelo MRG para representar de mejor manera las fibras que se desean simular en términos del comportamiento eléctrico esperado al momento de aplicar estimulación eléctrica de alta frecuencia como la utilizada en SCS. El método de optimización en particular corresponde al de una estrategia evolutiva mediante adaptación de matrices de covarianza (*CMA-ES*), buscando minimizar el error producto de las simulaciones realizadas sobre el modelo en comparación a aquellos obtenidos de forma experimental.

1.3.1 Hipótesis de trabajo

Con base a las preguntas presentadas y lo revisado en la literatura, el problema principal que se tiene es el de la búsqueda de parámetros para el modelo MRG de fibra nerviosa, con tal de encontrar un conjunto de estos que se ajusten de mejor manera a los resultados descritos en Crosby et al. (2017) al momento de simular aquel tipo de fibra nerviosa (axón singular de la columna dorsal), esto en comparación con los parámetros originales propuestos por McIntyre et al. (2002). Por ende, la hipótesis de trabajo que se define es la siguiente:

- **La aplicación de un algoritmo de búsqueda de parámetros CMA-ES al modelo MRG de fibra nerviosa permite encontrar un conjunto de valores electrofisiológicos y/o morfológicos que ajusten de mejor manera el modelo al tipo de neurona registrada en Crosby et al. (2017) en términos de su comportamiento eléctrico al aplicar estimulación eléctrica de alta frecuencia. Esto en comparación a los parámetros originales.**

1.4 OBJETIVOS Y ALCANCES DEL PROYECTO

1.4.1 Objetivo general

Implementar un algoritmo de búsqueda de parámetros CMA-ES aplicado al modelo MRG de fibra nerviosa, comparando el ajuste del modelo utilizando los parámetros obtenidos con

los experimentos y resultados mostrados en Crosby et al. (2017), para encontrar un conjunto de parámetros del modelo que caracterice de manera más exacta estos en relación a los parámetros originales descritos por McIntyre et al. (2002).

1.4.2 Objetivos específicos

- Simular el experimento de Crosby et al. (2017) utilizando Python + NEURON, permitiendo comparar los resultados de esta simulación para distintos parámetros del modelo con los resultados experimentales mediante una métrica de ajuste.
- Implementar el algoritmo CMA-ES propuesto en Xu et al. (2019), utilizando dentro de este el software de simulación NEURON para el cálculo de la métrica de ajuste referida en el punto anterior mediante la simulación del experimento para distintos parámetros, utilizando esta métrica como la función objetivo sobre la cual se optimiza el modelo.
- Diseñar e implementar una estrategia para realizar las simulaciones requeridas y la ejecución del algoritmo de optimización de forma distribuida utilizando un clúster computacional.
- Comparar los parámetros obtenidos mediante la ejecución del algoritmo de optimización con aquellos descritos en los trabajos de McIntyre et al. (2002), y determinar si tales parámetros aplicados al modelo MRG resultan en una mejor caracterización de los resultados experimentales vistos en Crosby et al. (2017), permitiendo recrear estos de forma precisa.

1.4.3 Alcances

Este trabajo contempla el modelado del problema objetivo como uno de optimización, considerando funciones objetivo y restricciones, la adaptación del algoritmo CMA-ES implementado por un framework para poder resolver el problema, integrando este con el software de simulación NEURON para el computo de la función objetivo, además de todo el trabajo necesario para poder ejecutar este software en un clúster basándose en bibliotecas disponibles para esto. Se considera también el desarrollo de herramientas para el estudio de los datos obtenidos en las ejecuciones como es la realización de gráficos, además del análisis de los resultados obtenidos en base a estas herramientas.

1.5 METODOLOGÍAS Y HERRAMIENTAS UTILIZADAS

1.5.1 Ambiente experimental

Entre las partes que componen del ambiente experimental, está el equipamiento computacional a utilizar, el cual se describe en detalle en la sección de software y hardware requerido. Por otro lado, dado que no existe necesidad de obtención de información en terreno u otra necesidad similar, el espacio físico donde se desarrollan los procedimientos está limitado a la habitación y recinto donde se operen los equipos computacionales. Es necesario que en este exista conexión a internet hacia el equipo computacional, particularmente para el uso de otros equipos y sistemas de forma remota, como puede ser un clúster computacional.

Entrando en detalle a lo necesario propio del estudio, se necesita acceso a toda la literatura descrita en este documento para utilizarse como referencia, así como también conjuntos de datos definidos a utilizar. El software requerido se debe encontrar instalado si corresponde en los equipos a utilizar.

1.5.2 Procedimiento experimental

El procedimiento experimental se centra en la aplicación de una estrategia evolutiva de adaptación de matriz de covarianza (CMA-ES) para ajustar los parámetros del modelo MRG con tal representar de mejor manera los resultados descritos en Crosby et al. (2017) en términos del comportamiento eléctrico esperado, particularmente ante la simulación de SCS a 10 kHz. Los motivos para elegir el uso de CMA-ES sobre otros vistos en la literatura dado su buen rendimiento general y el evitar utilizar métodos que requieran de un problema de optimización diferenciable, como pueden ser métodos de gradiente, particularmente considerando que el cálculo de los potenciales y otras funciones del modelo se hacen mediante software que implementa las ecuaciones correspondientes de forma interna en vez de utilizarse estas directamente.

Para llevar el experimento a cabo, necesitamos definir primero que cuál es la representación de la solución a utilizar. Considerando que se deben variar tanto los parámetros morfológicos y electrofisiológicos que utiliza el modelo, estos se pueden representar como un arreglo donde cada posición es un parámetro y el valor en aquella es el valor que toma el parámetro.

Dada la representación de la solución a utilizar, se requiere además definir una

función objetivo mediante la cual se evalúa cada solución generada. Debido a que lo buscado en este experimento es un conjunto de parámetros del modelo para que esté logre expresar el mismo comportamiento descrito en la literatura, debemos evaluar que tanto los resultados generados por el modelo (aplicada la solución encontrada) generan un comportamiento similar al recrear el experimento de Crosby et al. (2017) para SCS de 10 kHz. La forma de evaluar se define de la siguiente manera:

1. Utilizando los parámetros de la solución generada, recrear el experimento de SCS a 10 kHz para 30 ejemplares de axón generados aleatoriamente, cada uno con un diámetro de fibra nerviosa dentro de los rangos definidos en Crosby et al. (2017) para SCS bifásico a 10 kHz. Esto debe realizarse para distintas amplitudes durante intervalos de 5 segundos y en cada una generando una curva de densidad de impulsos promedio.
2. Realizar una comparación de cada curva de densidad obtenida para la solución generada con la curva esperada para la amplitud correspondiente descrita en la literatura, utilizando las diferencias entre ambas curvas en distintos puntos como medida de error. Entre menores sean los valores obtenidos, mayor es la similitud de los resultados simulados con los esperados, y por ende una mayor calidad del modelo.

Dado esto, se establece como métrica de calidad del modelo el error cuadrático medio entre cada curva de densidad de impulsos promedio generada por el modelo dado un conjunto de parámetros, al compararla con la curva de densidad correspondiente esperada según los resultados en la literatura. Al compararse curvas para múltiples amplitudes de SCS, los errores obtenidos para cada una se ponderan de igual manera, dando como valor un error ponderado el cual representa el valor de la función objetivo a minimizar para los parámetros dados. Esta se puede definir de como se muestra en la ecuación (1.1), Siendo p el conjunto de parámetros a evaluar del modelo, E_{ij}^p el valor esperado y G_{ij}^p el valor generado utilizando el conjunto de parámetros p , ambos en para la curva j en el punto i . N la cantidad de puntos a evaluar entre ambas curvas (Alrededor de 50 dependiendo del cálculo de la curva del histograma) y S la cantidad de curvas a comparar, siendo una por cada amplitud sobre la cual se esté evaluando el modelo.

$$F(p) = \frac{1}{S} \sum_{j=1}^S \frac{1}{N} \sum_{i=1}^N (E_{ij} - G_{ij}^p)^2 \quad (1.1)$$

El procedimiento para llevar a cabo el experimento parte con la configuración del software para representar el modelo MRG, siendo utilizado en este caso un modelo ya generado mediante el software de simulación de fibras nerviosas NEURON, usando Python como interfaz para manipular este. Se deben definir métodos e interfaces para poder realizar las modificaciones de parámetros necesarias, así como también la simulación de experimentos, generación de cur-

vas de densidad promedio y comparación de la curva generada con la esperada para el cálculo de la función objetivo. El código generado en este punto debe definirse como una librería para su uso dentro del algoritmo de optimización mediante estrategia evolutiva (CMA-ES). Para esto último se utilizará DEAP (Fortin et al., 2012), el cual permite implementar el algoritmo basado en el trabajo de Hansen & Ostermeier (2001). Uno de los puntos por los que se decide utilizar DEAP es su integración con SCOOP para distribución de tareas en cluster computacionales, reduciendo la espera entre cada iteración del experimento.

A modo de referencia, el tiempo que demora en ejecutar en un solo núcleo una simulación del experimento de Crosby et al. (2017) para los primeros 5 segundos es de aproximadamente 3 minutos. Para obtener una curva de histograma se debe realizar esta simulación para 30 ejemplares de fibras, lo cual resulta en un tiempo de 90 minutos. Si se realiza la comparación para 3 histogramas para diferentes amplitudes de SCS, se tienen 270 minutos, lo cual representa una aproximación del tiempo para evaluar una generación, de ejecutar el cálculo de la función objetivo para cada solución en la población (estimada de tamaño entre 40 a 50 dependiendo de la disponibilidad del cluster) de manera paralela, una en cada núcleo o hilo. Considerando esto último, inicialmente se consideran aproximadamente 12 días para la realización de 60 generaciones durante el algoritmo. Estos tiempos son una estimación, la cual podría estar sujeta a una disminución o aumento según se realicen optimizaciones sobre el código y los recursos computacionales que se tengan disponibles, entre otros factores.

Según los resultados descritos por McIntyre et al. (2002), una de las limitaciones del modelo se encuentra en la representación de los canales iónicos, los cuales actualmente son una forma condensada de lo que existe en una fibra nerviosa, pero posiblemente equivalente a estas. Por esto, se desea experimentar variando primero solo los parámetros electrofisiológicos, realizando una búsqueda en detalle dentro del espacio de solución referente a estos. Como paso siguiente, y de no encontrar resultados lo suficiente satisfactorios, se expande la búsqueda considerando los parámetros morfológicos, observando si existe una mejora en las soluciones encontradas.

Ya definido el como realizar el procedimiento de CMA-ES sobre el modelo configurado en Python + NEURON, es necesario realizar este en dos iteraciones para distintos conjuntos de parámetros definidos generalmente de la siguiente forma:

- Parámetros electrofisiológicos.
- Parámetros electrofisiológicos y morfológicos.

Dada la naturaleza del CMA-ES donde es necesaria una solución inicial al menos, al terminar de operar sobre un conjunto de parámetros y pasar al siguiente, es posible utilizar la mejor solución encontrada en el anterior, si es que esta representa una mejora respecto a los

parámetros originales utilizados.

Por cada conjunto de parámetros sobre los que se opera, se realizan estas operaciones para distintos rangos de valores descritos en la siguiente sección. Por último, la ejecución del algoritmo CMA-ES debe considerar la definición de un tamaño de población y una cantidad de generaciones a realizar, esto último en base a una métrica para definir una condición de termino como puede ser el estancamiento, entre otras.

Por cada ejecución que se haga del algoritmo CMA-ES para cada conjunto de parámetros del modelo MRG se deben registrar las soluciones generadas en cada ejecución del algoritmo para estudiar su evolución y como influyen las distintas variables entre cada ejecución.

1.5.3 Descripción de los datos o corpus

Respecto a las variables electrofisiológicas a modificar, estas se encuentran definidos en la tabla 1.1, indicando sus valores iniciales y rangos donde se restringen.

Tabla 1.1: Resumen de variables electrofisiológicas a optimizar con restricciones.

Fuente: Elaboración propia (2021)

Variable	Valor Inicial	Mínimo	Máximo
Capacitancia nodal	2 $\mu F/cm^2$	0.5 $\mu F/cm^2$	3 $\mu F/cm^2$
Capacitancia internodal	2 $\mu F/cm^2$	0.5 $\mu F/cm^2$	3 $\mu F/cm^2$
Conductancia MYSA	0.001 S/cm^2	0.0001 S/cm^2	0.01 S/cm^2
Conductancia máxima Na^+ rápido	3 S/cm^2	0.01 S/cm^2	10 S/cm^2
Conductancia máxima K^+ lento	0.09 S/cm^2	0.001 S/cm^2	1 S/cm^2
Conductancia máxima Na^+ persistente	0.01 S/cm^2	0.001 S/cm^2	1 S/cm^2
Conductancia del filtrado nodal	0.007 S/cm^2	0.00001 S/cm^2	0.01 S/cm^2

Respecto a los parámetros morfológicos del modelo, una consideración a realizar es debido a que los parámetros registrados en McIntyre et al. (2002) se definen para diámetros de fibra desde 5.7 μm hasta 16 μm , mientras que los diámetros de fibra definidos en Crosby et al. (2017) van desde 1.8 μm hasta 5.6 μm , es necesario realizar una extrapolación de los distintos parámetros de la fibra nerviosa definidos por McIntyre para extrapolar los correspondientes a los diámetros de fibra utilizados en cada simulación de SCS generada dentro de la iteración. A modo de ejemplo, la longitud FLUT se puede expresar en función del diámetro de fibra en conjunto a una serie de constantes, y si se desea ajustar este parámetro, el valor a variar en cada generación del algoritmo para este caso son las constantes de esta función. De esta forma, la solución generada resulta en un modelo aplicable a distintos diámetros de fibra nerviosa.

Dado esto último, para los parámetros morfológicos a optimizar se definen primero las funciones a utilizar para extrapolar cada parámetro con la forma $f(x) = Ad(x) + B$, siendo A y

B constantes y $d(x)$ en función del diámetro de fibra x . Los valores dados para estos parámetros se definen en la tabla 1.2. Tanto del diámetro x como los parámetros están en μm .

Tabla 1.2: Funciones de extrapolación de parámetros morfológicos con constantes iniciales.

Fuente: Elaboración propia (2021)

Parámetro	Función	A inicial	B inicial
Separación internodal	$Ax + B$	97.18	54.046
Diámetro Nodal/MYSA	Ae^{Bx}	1.1468	0.1001
Largo FLUT	$A \ln(x) + B$	23.865	7.8005
Largo STIN	$Ax + B$	15.751	3.848
Diámetro FLUT/STIN	$A \ln(x) + B$	5.9917	1.2499
Número de laminillas de mielina	$[A \ln(x) + B]$	58.58	14.869

Con base a esto, las variables morfológicas a optimizar durante la ejecución del algoritmo corresponden a estos valores A y B presentes en la función definida para cada parámetro morfológico del modelo. Por esto, se restringen los valores que pueden tomar estas variables a los presentes en la tabla 1.3. Estos valores se calculan en base al valor inicial para A y B, multiplicando ambos valores por 0.4 y 1.6, representando esto un margen de un 60% respecto a los valores de referencia en los que se basó el modelo, considerando que estos representen un rango de valores biológicamente factibles.

Tabla 1.3: Rangos posibles a tomar por los valores A y B para cada función correspondiente a un parámetro del modelo MRG.

Fuente: Elaboración propia (2021)

Parámetro	Constante A		Constante B	
	Mínimo	Máximo	Mínimo	Máximo
Separación internodal	24.295	170.065	13.5115	94.5805
Diámetro Nodal/MYSA	0.2867	2.0069	0.025	0.1751
Largo FLUT	5.9663	41.7638	1.9501	13.6509
Largo STIN	3.9378	27.5642	0.962	6.734
Diámetro FLUT/STIN	1.4979	10.4854	0.3125	2.1873
Número de laminillas de mielina	14.645	102.515	3.7173	26.0208

Cualquier otro parámetro que no se haya mencionado en las tablas se considera como constante dentro del modelo y se utiliza tal cual es definido en el trabajo de McIntyre et al. (2002).

1.5.4 Software y hardware requerido

Respecto a software, la base de todo el código realizado se hace en Python 3.8.x, utilizando NEURON 7.8.1.1 para simulación del modelo MRG y DEAP 1.3.1 para implementación de CMA-ES. Pueden necesitarse otras bibliotecas para tareas menores.

En materia de entorno de desarrollo, el computador utilizado para desarrollar corresponde a uno ensamblado que cuenta con un procesador AMD Ryzen 5 2600 de 6-núcleos a 3.4 GHz y 16 GB de memoria ram DDR4 a 2933 Mhz. El sistema operativo en el que se desarrollo es Debian 10, en su versión estable. De ser necesario, se contempla utilizar alguna plataforma de computo en cluster, como la disponible en el departamento de informática de la Universidad de Santiago de Chile, u otra plataforma en la nube que pueda utilizarse para ejecutar los algoritmos de forma paralela.

1.6 ORGANIZACIÓN DEL DOCUMENTO

El cuerpo del documento está conformado por cinco capítulos sin contar esta introducción, los cuales se detallan a continuación:

- El capítulo 2 corresponde al marco teórico, donde se explican los conceptos necesarios para la comprensión de este documento, considerando lo que es sistema nervioso central y modelado computacional de este, estimulación de la médula espinal, métodos de optimización y herramientas de análisis utilizadas.
- El capítulo 3 corresponde al estado del arte, donde se explora la literatura actual relevante al problema. Se abordan temas como la estimulación de la médula espinal, el modelo de McIntyre, Richardson y Grill y su uso en la literatura, y optimización de parámetros en modelos de fibras nerviosas.
- El capítulo 4 describe la investigación realizada, definiendo la estructura, características y detalles del software a desarrollar para realizar esta, desde la definición del modelo, la implementación del optimizador y la ejecución del algoritmo en el clúster.
- El capítulo 5 muestra los resultados obtenidos del desarrollo de los experimentos descritos en el capítulo anterior, analizando el proceso de optimización observado y las soluciones obtenidas de este, además de un análisis de sensibilidad de las variables estudiadas.

- Por último, el capítulo 6 muestra las conclusiones obtenidas del trabajo, tanto del desarrollo del experimento como los resultados, y establece el trabajo a realizar a futuro.

CAPÍTULO 2. MARCO TEÓRICO

2.1 SISTEMA NERVIOSO Y MODELADO COMPUTACIONAL

El sistema nervioso, a diferencia de otros sistemas de órganos se encarga principalmente del tratamiento de señales y el procesamiento de información, actuando como un mecanismo de comunicación (Micheli-Tzanakou, 2014, p. 1). Este se compone de dos partes, el sistema nervioso central y el sistema nervioso periférico. El primero lo compone el cerebro y la médula espinal, en ambos presentes células denominadas neuronas, especializadas y capaces de conducir señales eléctricas entre ellas para comunicar sobre el ambiente externo a células adyacentes, ayudando así a regular el ambiente interno del cuerpo y respondiendo a estímulos externos. El sistema nervioso periférico, en cambio, lo componen los nervios espinales, craneales y periféricos junto a sus terminaciones sensoriales y motoras, los cuales se extienden desde el cerebro y la médula espinal y se pueden clasificar en aferentes, llevando información desde el sistema central al cuerpo, o eferentes, llevando información del resto del cuerpo al sistema central (Rea, 2015, p. 1-3).

En la medula espinal, su materia gris, zona con mayor concentración de cuerpos neuronales, la componen tres categorías de células nerviosas: células sensoriales correspondientes a funciones sensoriales y de arco reflejo, motoras alimentando el musculo esquelético y motoras alimentando el sistema nervioso autónomo (Rea, 2015, p. 9-10). La estructura la de la médula se organiza en cuatro secciones, cervical, torácica, lumbar y sacra. De estos se extienden 31 segmentos definidos por los pares de nervios que salen de la médula y la sección de esta, dividido en 8 cervicales, 12 torácicos, 5 lumbares, 5 sacros y un nervio coxígeo (Dafny, 2020), los cuales se presentan junto a su respectiva región de la médula en la figura 2.1.

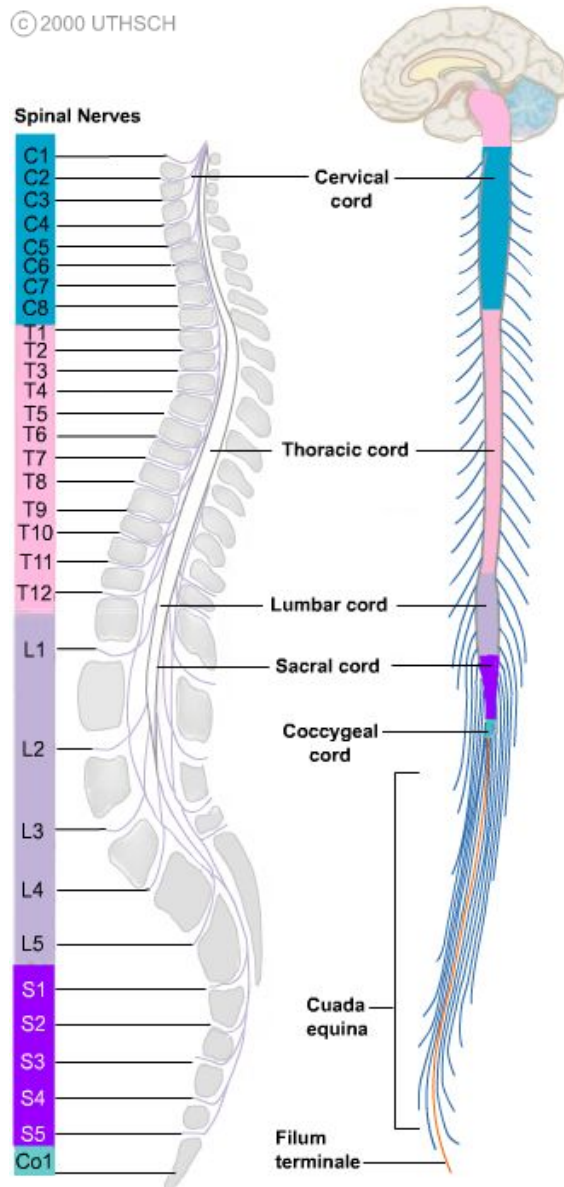


Figura 2.1: Diagrama de las regiones en médula espinal y nervios principales en sus respectivas salidas entre las vértebras.

Fuente: Dafny (2020)

Para analizar en mayor detalle el funcionamiento del sistema nervioso, la médula espinal y el comportamiento de las neuronas que lo componen es común utilizar modelos contruidos en base a teoría, buscando predecir y entender aquello observado de forma empírica, usualmente realizando suposiciones sobre elementos o mecanismos no cuantificados y/u observados. En el caso de las neuronas, éstas usualmente se modelan respecto a su comportamiento electrofisiológico, basándose en conceptos como el potencial de membrana e intercambio iónico en conjunto a modelado de circuitos eléctricos (Sterratt et al., 2011). En la figura 2.2 se muestra un ejemplo simplificado del circuito eléctrico equivalente de un trozo de

membrana de una fibra nerviosa. La pared celular se representa como una capacitancia, al existir un potencial eléctrico dado por la diferencia de electrones en ambos lados de la capa; los canales iónicos de sodio y potasio se modelan como una resistencia en serie con una batería según el flujo de los iones, y una fuente de corriente que representa un electrodo.

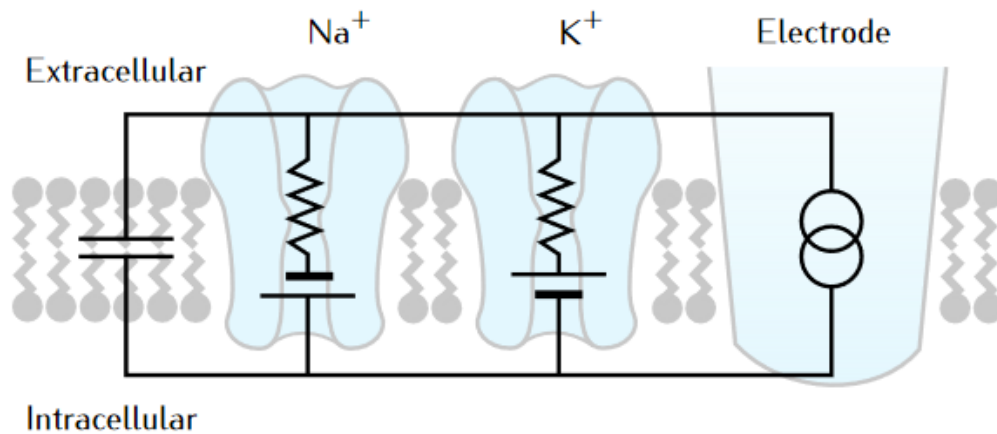


Figura 2.2: Diagrama de ejemplificación de un trozo de membrana de fibra nerviosa modelado como un circuito eléctrico.

Fuente: Sterratt et al. (2011)

Modelos como el anterior se considera que tienen un potencial de membrana efectivamente constante, lo cual en la práctica no se da mayormente, pues la corriente fluye a lo largo de la fibra y los potenciales a lo largo de la membrana varían con el tiempo. Para abordar esto consideramos un modelo de tipo compartimental, donde cada componente tiene un largo l y diámetro d , y por cada uno puede fluir corriente tanto por la capacitancia de la membrana o a través de las resistencias, a lo cual se suma la posibilidad de fluir mediante resistencias externas e internas representando el medio extra-celular e intra-celular respectivamente. En la figura 2.3 se observa un ejemplo del modelado compartimental de una membrana, donde por cada compartimento se considera la capacitancia producto de la pared celular C_m , la resistencia y batería correspondiente a los canales iónicos R_m y E_L , además de la resistencia existente en el medio intracelular R_a conectando los compartimientos internamente. La resistencia extra-celular, dado que se puede considerar muchísimo menor que las resistencias internas a la membrana en ese caso se asume como cero y se incluye una conexión a tierra, considerando que actúa en forma isopotencial (mismo potencial en todos los puntos). Según el modelo, se pueden considerar resistencias extra-celulares mayores a cero de ser necesario (Sterratt et al., 2011).

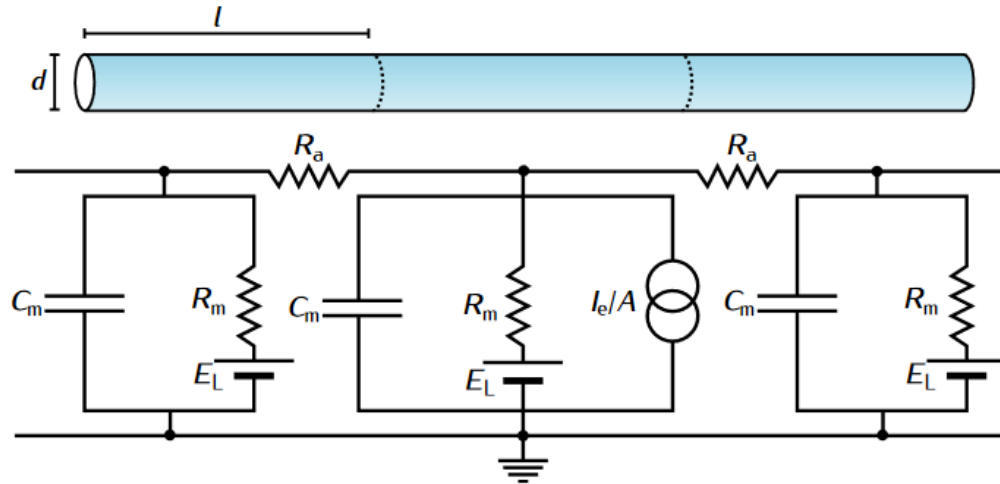


Figura 2.3: Diagrama de ejemplo del modelado compartimental de una fibra nerviosa.

Fuente: Sterratt et al. (2011)

Otro concepto a entender en el funcionamiento de las fibras nerviosas es el de potencial de acción, visto también como impulso nervioso o *spike*. Esto corresponde a la inversión del potencial de la membrana ante un estímulo con una intensidad mayor a un valor umbral propio de la fibra; si el estímulo tiene una intensidad menor al umbral no se genera el potencial de acción en ninguna medida, pero de pasar este umbral el evento se gatilla en su totalidad, con una amplitud y duración bien definida (Chambers et al., 2019, p. 225).

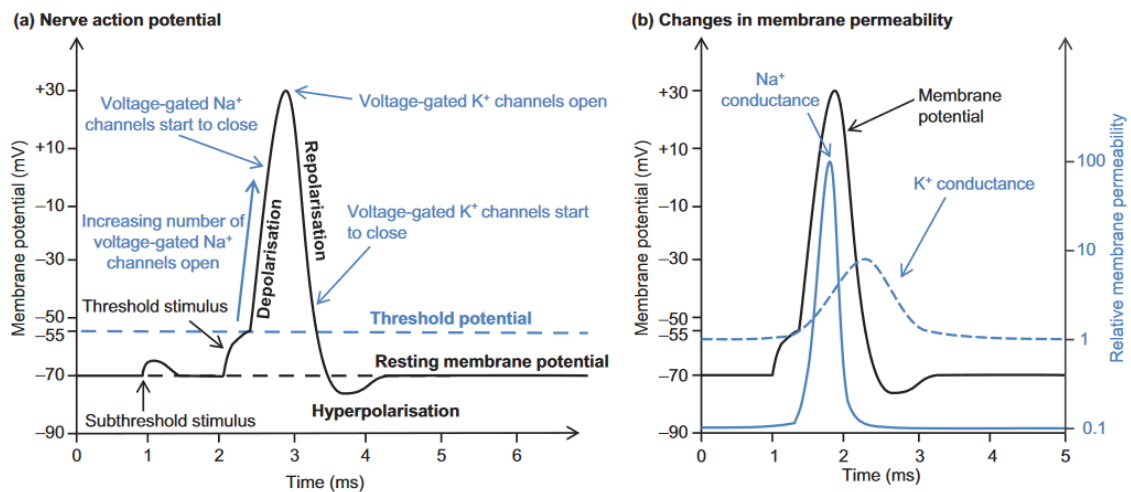


Figura 2.4: Diagrama del potencial de acción (a) y cambios en la permeabilidad de la membrana (b).

Fuente: Chambers et al. (2019)

En la figura 2.4 se observa en la parte izquierda el como el potencial de la membrana, una vez recibido un estímulo que sobrepasa el potencial de umbral comienza un proceso de

despolarización, con un incremento en la apertura momentánea de canales de sodio Na^+ lo cual incrementa el potencial de la membrana llegando a una polaridad positiva. Llegado cierto punto, los canales de sodio comienzan a cerrarse, y canales de potasio K^+ empiezan a abrirse momentáneamente dando paso a la repolarización, reduciendo el potencial de la membrana mientras se van cerrando los canales de potasio, llegando el potencial nuevamente bajo el umbral y pasando por un punto de hiperpolarización, posteriormente volviendo al potencial de reposo que tenía originalmente (Chambers et al., 2019, p. 225-226). En la parte derecha de la figura se observa como según cambia el potencial de acción, la permeabilidad de la membrana varía según se abren y cierran canales de sodio y potasio.

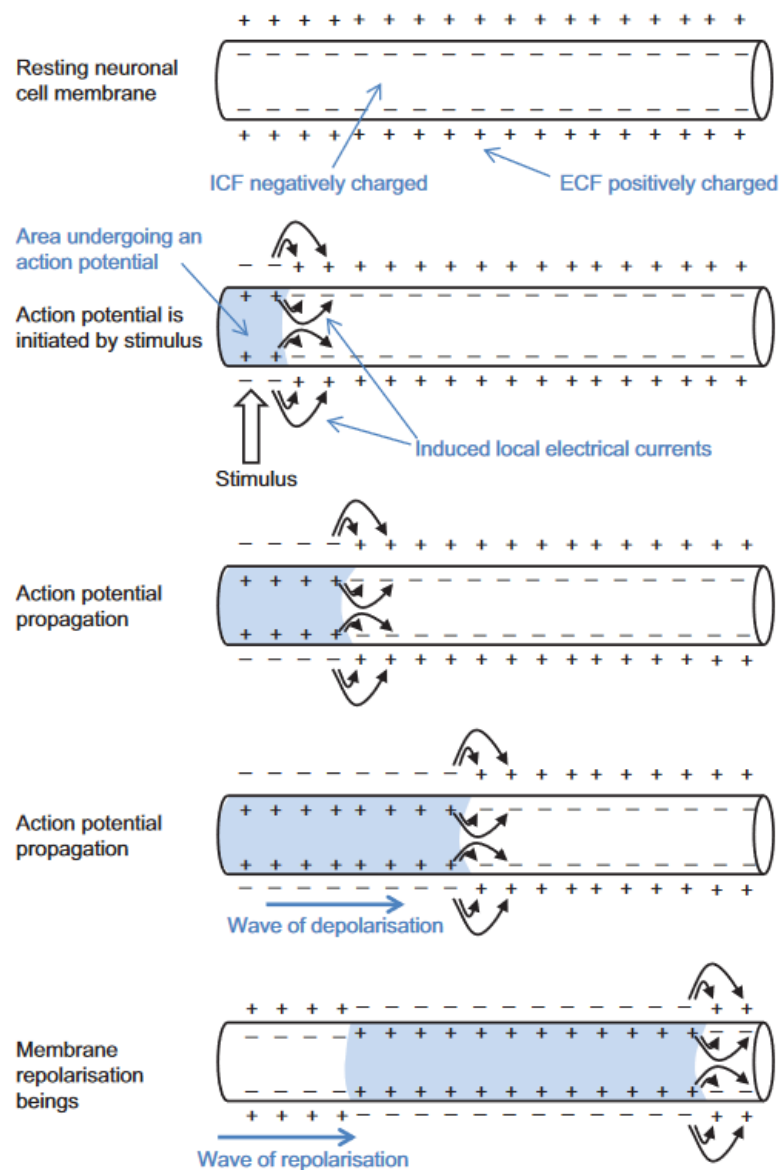


Figura 2.5: Diagrama de propagación del potencial eléctrico en una fibra nerviosa.
Fuente: Chambers et al. (2019)

La propagación del potencial de acción, como se describe en la figura 2.5, se produce en base a que en este se despolariza una sección de membrana, resultando en que la sección interna de la superficie quede cargada positivamente, y el movimiento de iones en los bordes de la membrana despolarizada produce un flujo de corriente, resultando en la despolarización de la sección adyacente (Chambers et al., 2019, p. 225-226). La corriente propagada a lo largo de la fibra nerviosa decae, pero mientras la despolarización propagada de la sección anterior sea suficiente para llegar al umbral de potencial, se genera el potencial de acción en la sección.

2.2 ESTIMULACIÓN DE LA MÉDULA ESPINAL (SCS)

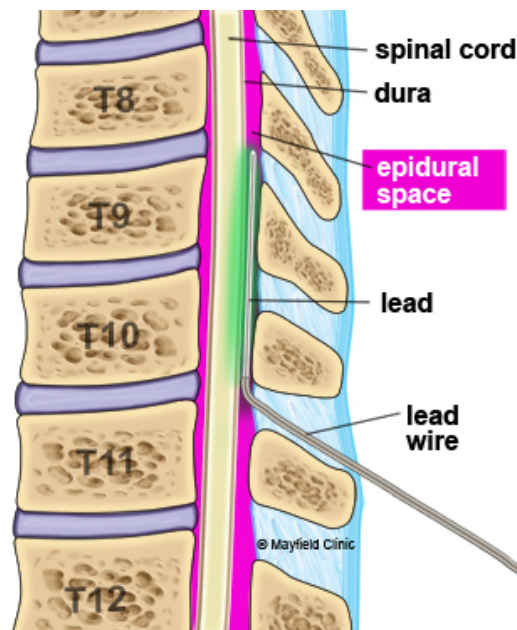


Figura 2.6: Diagrama de la inserción de electrodos en el espacio epidural de la columna.

Fuente: Mayfield Clinic (2021)

La estimulación de la médula espinal tiene sus bases en la teoría de la compuerta, descrita por Melzack et al. (1965) donde se propone que el control del dolor puede ser conseguido mediante la activación de determinadas fibras nerviosas. En las últimas décadas los métodos de SCS se han desarrollado introduciendo diferentes modalidades de estimulación, como puede ser mediante tecnología de alta frecuencia, estimulación del ganglio de la raíz dorsal o estimulación por ráfagas (Sdrulla et al., 2018). La SCS se aplica usando un estimulador que cuenta con electrodos, los que se colocan en el espacio epidural presente entre la vertebra y la médula (figura 2.6), posicionado paralelo a esta última. Mediante los electrodos se aplican pulsos eléctricos que

pueden activar fibras de la medula que, de acuerdo con la teoría de la compuerta, podrían inhibir la información del dolor que viaje por otras vías. De este modo, se altera y/o cubre la señal de dolor en dirección al cerebro, aunque la efectividad de este método varía de según el paciente (Mayfield Clinic, 2021). Alrededor de la mitad de los pacientes experimenta alivio del dolor, por lo que aún hay un gran margen de mejora de la terapia SCS (Zhang et al., 2014).

2.3 MÉTODOS DE OPTIMIZACIÓN

La necesidad de optimizar es algo que está presente en diversas disciplinas, ya sea en la necesidad de maximizar la eficacia y minimizar costos de producción, o minimizar el error de un sistema. El proceso de optimizar se compone principalmente de un modelo computacional que represente el problema objetivo y un algoritmo de búsqueda (Yang & Koziel, 2011). La forma general de definir un problema de optimización es mediante una o más funciones objetivo a minimizar/maximizar, como se muestra en (2.1), para el cual se definen restricciones que limitan el espacio solución según requiera el problema como se muestra en (2.2). Generalmente, estas restricciones representan distintas condiciones del problema que debe respetar el modelo, como puede ser un límite de producción según la capacidad de una fábrica, o mantener una equivalencia entre materias primas generadas y utilizadas, para evitar falta o desperdicio de esta, entre otros factores.

$$\text{minimizar } f_i(x), i \in \{1, 2, 3, \dots, M\} \quad (2.1)$$

$$g_k(x) \leq 0, k \in \{1, 2, 3, \dots, K\} \quad (2.2)$$

Los algoritmos de optimización se pueden clasificar en distintas categorías, por ejemplo, exhaustivos donde se recorre todo el espacio solución (fuerza bruta), heurísticos donde se utiliza una regla para guiar la búsqueda hacia una buena solución, meta-heurísticos donde se combina búsqueda local con estrategias de alto nivel para una búsqueda robusta en el espacio solución, o hiper-heurísticos donde se usan estrategias de alto nivel para operar sobre un espacio de heurísticas a utilizar en vez de directamente en el espacio de soluciones (Burke et al., 2019). Además, se pueden dividir entre aquellos basados en derivadas (gradiente) como el método de Gauss-Newton y aquellos que no requieren de derivadas como el método Nelder-Mead, o entre algoritmos de trayectoria, basados alrededor de una sola solución como el método de enfriamiento simulado, o métodos de población basados en múltiples soluciones por iteración como son los algoritmos genéticos. También pueden clasificarse según si son deterministas o estocásticos, o

basados en memoria o no, entre otras categorías (Yang & Koziel, 2011).

Para determinar el tipo de algoritmo a utilizar, entre los factores a considerar existe un tercer elemento dentro del proceso de optimización como se menciona en Yang & Koziel (2011) que corresponde al simulador, véase, aquella parte del proceso donde se evalúa la función objetivo a optimizar, que puede ser desde calcular una simple ecuación lineal, evaluar un múltiples integrales, realizar simulaciones enteras de un experimento o hasta resolver un subproblema de optimización. Según la complejidad del simulador y el coste computacional que requiera evaluar una solución, por ejemplo, el tiempo requerido para evaluar la solución para cierto equipo computacional, el uso de algunos tipos de algoritmos que necesitan realizar múltiples evaluaciones como puede volverse infactible, requiriendo el uso de otros métodos como algoritmos heurísticos que buscan una solución no necesariamente óptima, pero que se acerque lo suficiente a una como para ser de utilidad, esto requiriendo un uso reducido del simulador y por ende una reducción en el coste computacional.

Dentro de los algoritmos de optimización heurísticos existen dos tipos en particular: aquellos con heurísticas específicas dependientes del problema, y aquellos basados en metaheurísticas que son generales y aplicables a distintos problemas (Talbi, 2009, p. 21). En este trabajo, debido a que el problema a resolver considera una función objetivo cuya forma se desconoce y conlleva un alto coste computacional para su cálculo, los algoritmos que más nos interesan son los correspondientes a metaheurísticas, y en especial los que entran en categoría de algoritmos de población como son los algoritmos genéticos o estrategias evolutivas. Generalmente, estos algoritmos se basan en una población de soluciones generadas, y de forma iterativa se van evaluando estas soluciones, de modo que se genera una nueva población según algún criterio establecido (Talbi, 2009, p. 191). Un esquema general de estos se describe en el Algoritmo 2.1, donde P_t es la población dada para una generación t , y P'_t es la población descendiente de P_t generada mediante, por ejemplo, estrategias de cruce.

Algoritmo 2.1: Esquema general de algoritmos de población.

```
1  $P_0$ (Población Inicial);
2  $t = 0$ ;
3 repetir
4    $P'_t = \text{Generar}(P_t)$ ;
5    $P_{t+1} = \text{Seleccionar}(P_t \cup P'_t)$ ;
6 hasta que se alcance la condición de parada;
Resultado: Mejor solución encontrada
```

2.3.1 Estrategia evolutiva mediante adaptación de matrices de covarianza (CMA-ES)

Dentro de las metaheurísticas, en el subconjunto de los algoritmos evolutivos existen las denominadas estrategias evolutivas, las cuales son mayormente aplicadas a problemas de optimización continuos donde la representación hace uso de valores reales. Usualmente utilizan un reemplazo elitista de los individuos en la población y una distribución normal específica (Talbi, 2009, p. 202).

Los pasos del algoritmo en general para un problema con N variables y una población λ están dados a continuación, como se describen por Hansen & Ostermeier (2001). Se parte definiendo primero caminos de evolución vacíos $P_\sigma^{(0)} = 0$ y $P_c^{(0)} = 0$ para usarse más adelante en lo que corresponde al control del tamaño de los pasos de mutación y la adaptación de la matriz de covarianza respectivamente, la última inicializada como una matriz identidad $C^{(0)} = I$. Es necesario además definir un tamaño inicial del paso de mutación $\sigma^{(0)} \in \mathbb{R}_+$ usado, y una media de la distribución normal $m^{(0)} \in \mathbb{R}^N$ según sea el problema (una solución factible conocida por ejemplo).

Por cada generación g partiendo de 0, considerando una media m^g , varianza $(\sigma^{(g)})^2$ y matriz de covarianza $C^{(g)}$, el conjunto de individuos $x_k^{(g+1)}$ de esta se obtiene en base al muestreo de una distribución normal en base a los parámetros dados como se muestra en (2.3).

$$x_k^{(g+1)} \sim N(m^g, (\sigma^{(g)})^2 C^{(g)}) \quad \forall k = 1, \dots, \lambda \quad (2.3)$$

En base a estos individuos generados, se seleccionan los μ mejores individuos mediante los cuales se obtiene una nueva media de la distribución, lo que correspondería a la recombinación dentro de la estrategia evolutiva, esto según el peso w_i que se asigne a cada individuo i .

$$m^{(g+1)} = \sum_{i=1}^{\mu} w_{i:\lambda}^{(g+1)}, \quad \sum_{i=1}^{\mu} w_i = 1 \quad (2.4)$$

Posteriormente, se procede a actualizar el camino de evolución $P_\sigma^{(g+1)}$ y el tamaño del paso $\sigma^{(g+1)}$ en base a sus valores de la generación anterior g , siendo μ_{eff} la masa de selección de varianza efectiva, c_c representa la tasa de aprendizaje de la acumulación de la actualización de ranking-uno de la matriz y d_σ es un factor de amortiguamiento.

$$P_\sigma^{(g+1)} = (1 - c_\sigma)P_\sigma^{(g)} + \sqrt{c_\sigma(2 - c_c)\mu_{eff}} C^{(g)^{-\frac{1}{2}}} \frac{m^{(g+1)} - m^{(g)}}{\sigma^{(g)}} \quad (2.5)$$

$$\sigma^{(g+1)} = \sigma^{(g)} \exp \left(\frac{c_\sigma}{d_\sigma} \left(\frac{\|P_\sigma^{(g+1)}\|}{E\|N(0, I)\|} - 1 \right) \right) \quad (2.6)$$

c_σ representa una tasa de aprendizaje para la acumulación del control del tamaño del paso, el que se obtiene según la ecuación (2.7), mientras que el factor de amortiguamiento d_σ se obtiene como se describe en la ecuación (2.8).

$$c_\sigma = \frac{\mu_{eff} + 2}{n + \mu_{eff} + 3} \quad (2.7)$$

$$d_\sigma = 1 + 2 \left(0, \sqrt{\frac{\mu_{eff} - 1}{n + 1}} - 1 \right) + c_\sigma \quad (2.8)$$

c_c es usado también en determinar el camino de evolución $P_c^{(g+1)}$. μ_{cov} es el parámetro para ponderar entre la actualización de ranking-uno y ranking- μ , el cual es generalmente apropiado sea equivalente a μ_{eff} . Estos valores se pueden determinar como se muestra en (2.9):

$$c_c = \frac{4}{n + 4}, \quad \mu_{cov} = \mu_{eff} = \frac{1}{\sum_{i=1}^{\mu} w_i^2} \quad (2.9)$$

Ya con esto, la actualización de la matriz de covarianza se realiza primero adaptando el camino de evolución correspondiente y en base a este la matriz, como se describen en las ecuaciones (2.10) y (2.11). $h_\sigma^{(g+1)}$ representa una función de heaviside usada para frenar la actualización de $P_c^{(g)}$ si $\|P_\sigma^{(g+1)}\|$ resulta ser demasiado grande, previniendo incrementos de la matriz C demasiado rápidos. Esta se encuentra dada por la ecuación (2.12). Por último, c_{cov} corresponde a la tasa de actualización de la matriz de covarianza, la cual se puede obtener mediante la ecuación (2.13).

$$P_c^{(g+1)} = (1 - c_c)P_c^{(g)} + h_\sigma^{(g+1)} \sqrt{c_c(2 - c_c)\mu_{eff}} \frac{m^{(g+1)} + m^{(g)}}{\sigma^{(g)}} \quad (2.10)$$

$$C^{(g+1)} = (1 - c_{cov})C^{(g)} + \frac{c_{cov}}{\mu_{cov}} P_c^{(g+1)} P_c^{(g+1)T} + c_{cov} \left(1 - \frac{1}{\mu_{cov}}\right) \sum_{i=1}^{\mu} w_i OP \left(\frac{x_{i:\lambda}^{(g+1)} - m^{(g)}}{\sigma^{(g)}} \right) \quad (2.11)$$

OP : *Producto externo*

$$h_{\sigma}^{(g+1)} \begin{cases} 1 & \text{si } \frac{||P_{\sigma}^{(g+1)}||}{\sqrt{1-(1-c_{\sigma})^{2(g+1)}}} < (1.5 * + \frac{1}{n-0.5})E||N(0, I)|| \\ 0 & \text{en caso contrario} \end{cases} \quad (2.12)$$

$$E||N(0, 1)|| \approx \sqrt{n}(1 - \frac{1}{4n} + \frac{1}{21n^2})$$

$$c_{cov} = \frac{1}{\mu_{cov}} \frac{2}{(n + \sqrt{2})^2} + (1 - \frac{1}{\mu_{cov}}) \min(1, \frac{2\mu_{eff} - 1}{(n + 2)^2 + \mu_{eff}}) \quad (2.13)$$

El cálculo de estos parámetros “por defecto” definidos por Hansen (2006) se consideran robustos y aplicables a un amplio rango de funciones a ser optimizadas, lo que deja a criterio del usuario el tamaño de la población λ , la cantidad de individuos seleccionados por etapa μ y el peso que tiene cada individuo seleccionado $w_{i=1 \dots \mu}$. No obstante, se definen las ecuaciones mostradas en (2.14) como valores recomendados para la población e individuos seleccionados. Incrementar el tamaño de la población usualmente mejora la robustez del algoritmo para converger a soluciones de mejor calidad a costa de una velocidad de convergencia disminuida.

$$\lambda = 4 + [3 \ln(n)], \quad \mu = [\lambda/2] \quad (2.14)$$

2.4 HERRAMIENTAS DE ANÁLISIS

2.4.1 Histogramas y funciones de densidad de probabilidad

Un histograma se puede definir como un gráfico de barra para datos cuantitativos, con una componente segmentada en intervalos adyacentes, y otra componente la frecuencia presente en cada intervalo (Witte & Witte, 2019, p. 33).

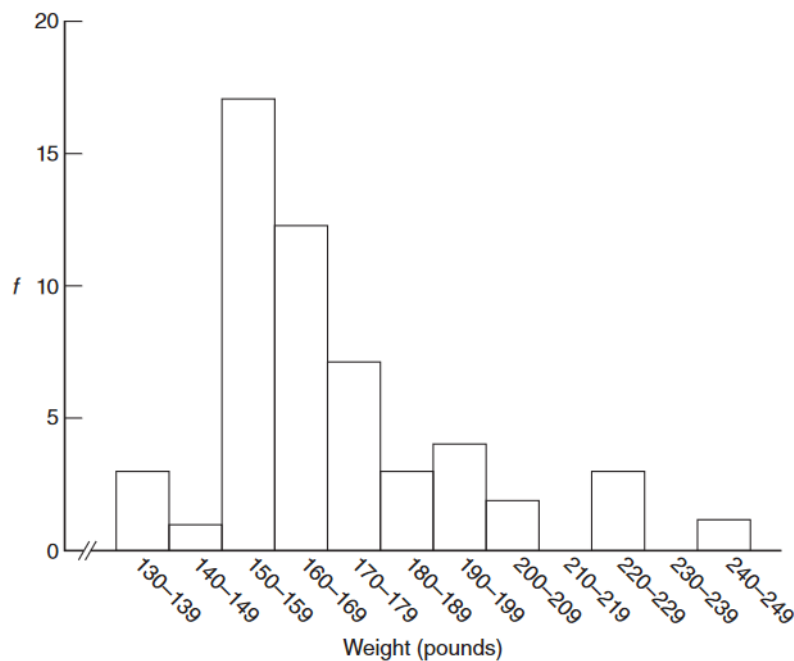


Figura 2.7: Ejemplo de histograma describiendo frecuencia según peso.
Fuente: (Witte & Witte, 2019, p. 33)

El tipo de histograma mostrado en la figura 2.7 representa específicamente un histograma de frecuencia. De normalizar cada frecuencia por tamaño de su intervalo correspondiente y la cantidad total de observaciones se obtiene una curva cuya integral equivale a 1. Esto representa un histograma de densidad (Scott, 2015, p. 56), el cual entrega una medida de la probabilidad por unidad del eje de los intervalos, lo que sería análogo a una función de densidad de probabilidad, correspondiente a la primera derivada de una función de distribución (Forbes, 2011, p. 3). En ambos casos, al ser funciones de probabilidad por unidad de la variable medida, al integrar esta en un intervalo se tiene la probabilidad de ocurrencia de una observación de la variable dentro del intervalo.

Siguiendo con esto, un tipo de histograma utilizado en el estudio de estímulos como puede ser impulsos en fibras nerviosas son los histogramas de tiempo de peristímulo o PSTH de sus siglas en inglés. Este histograma mide la frecuencia de impulsos ocurridos dentro de intervalos de tiempos regulares, generalmente producido midiendo un tren de impulsos repetidas veces para una determinada duración como se muestra en la figura 2.8, y asignando los impulsos detectado en cada tren al intervalo correspondiente dentro del histograma dando una frecuencia total, la cual se promedia según el número de mediciones para entregar un histograma de impulsos promedio por intervalo de tiempo. Esto último se puede dividir por el ancho de cada intervalo para obtener un gráfico de puntas promedio por unidad de tiempo (Bretschneider & De Weille, 2018, p. 188-189).

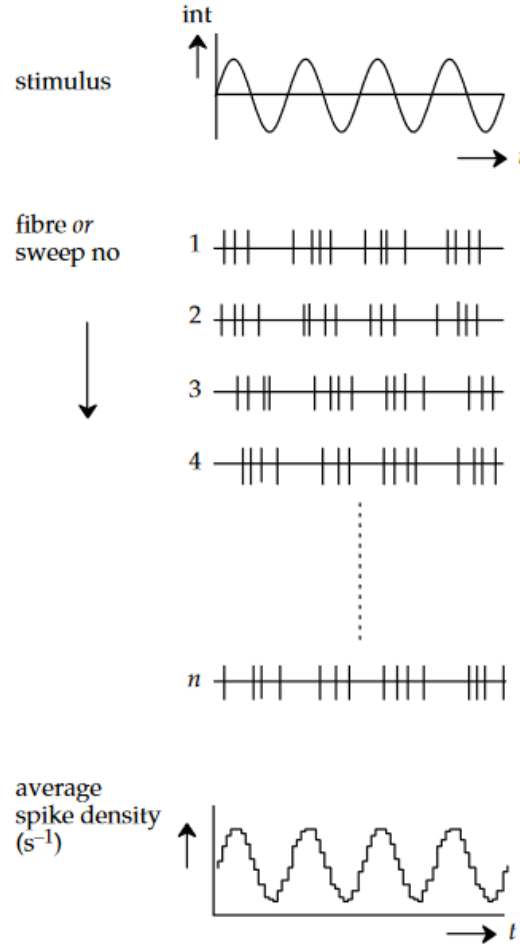


Figura 2.8: Principios para la determinación de un gráfico PSTH en base a la medición repetida de un estímulo.

Fuente: (Bretschneider & De Weille, 2018, p. 33)

Otra técnica para estimar la densidad de la frecuencia de una variable o la probabilidad de esta y que se usa en este trabajo es la estimación de la densidad del Kernel. Para entender esta, definimos lo que es un kernel normal (gaussiano) para el caso de una variable como se muestra en la ecuación (2.15). Este y cualquier otro kernel $K(u)$ debe ser una función de densidad de probabilidad con media cero y varianza acotada (Gramacki, 2018, p. 26).

$$K(u) = (2\pi)^{1/2} \exp\left(-\frac{1}{2}u^2\right) \quad (2.15)$$

Siguiendo dentro del caso para una variable, el estimador de la densidad del kernel para una muestra de n elementos $X_1, X_2, X_3, \dots, X_n$ provenientes de una función de densidad f está dada por la ecuación (2.16), donde h es el ancho de banda, que se interpreta como la desviación estándar σ en una distribución normal (Gramacki, 2018, p. 27).

$$\hat{f} = \frac{1}{nh} \sum_n^{i=1} K\left(\frac{x - X_i}{h}\right) = \frac{1}{n} \sum_n^{i=1} K_h(x - X_i) \quad (2.16)$$

$$K_h(u) = h^{-1} K(h^{-1}u) \quad (2.17)$$

Para mostrar como se interpreta el ancho de banda con la desviación estándar para un kernel gaussiano, consideremos la formula de una función de densidad de probabilidad normal para una media μ y desviación estándar σ como se muestra en (2.18). Si consideramos el caso para una distribución estándar normal $N(0, 1)$ como el kernel a utilizar como se muestra en (2.19), reemplazando en (2.16) y comparando con (2.18) se muestra que el estimador de la función de probabilidad corresponde a la suma ponderada de n distribuciones normales donde la desviación estándar σ corresponde al ancho de banda h (Gramacki, 2018, p. 27-29).

$$\mathcal{N}(x; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right) \quad (2.18)$$

$$K\left(\frac{x - x_1}{h}\right) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(x - X_i)^2}{2h^2}\right) \quad (2.19)$$

$$\begin{aligned} \hat{f}(x, h) &= \frac{1}{n} \sum_n^{i=1} \frac{1}{h\sqrt{2\pi}} \exp\left(-\frac{(x - X_i)^2}{2h^2}\right) \\ \hat{f}(x, h) &= \frac{1}{n} \sum_n^{i=1} \mathcal{N}(x; X_i, h^2) \end{aligned} \quad (2.20)$$

CAPÍTULO 3. ESTADO DEL ARTE

3.1 ESTIMULACIÓN DE LA MÉDULA ESPINAL

Dentro de los estudios relacionados a la estimulación de la médula espinal, aquel de mayor importancia para este trabajo y alrededor del cual se centra este estudio es aquel realizado por Crosby et al. (2017), que presenta pruebas experimentales de modulación de la actividad y conducción en axones de la columna dorsal mediante el uso de SCS con frecuencias por el orden de los Kilohertz. El objetivo principal de este trabajo fue el de cuantificar la activación y el bloqueo de conducción en los axones producto de este tipo de estimulación para un rango de frecuencias y tipos de onda, los cuales, como resultado de los estudios, se producían mayormente sobre el umbral motor, mostrando resultados similares dentro del rango de los 5 a 20 kHz, para tipos de onda bifásica y sinusoidal. Los resultados obtenidos logran caracterizar la actividad en el sistema nervioso central para modular la percepción del dolor y parestesia.

Los estudios realizados en Crosby et al. (2017) fueron realizados sobre un conjunto de 52 ratas anestesiadas, colocando una paleta de plomo bipolar de forma centrada bajo la lámina de la sección C5 en la columna, sobre C4, y una segunda paleta de plomo bipolar colocada de forma rostral bajo la lámina de la sección T10/T11. Un micro-electrodo bipolar fue colocado en T12/T13 para tomar registros. Para el análisis de la actividad en el axón, se utilizó un ordenamiento mediante análisis de componentes principales para la identificación de impulsos nerviosos. El diámetro de las fibras fue calculado en base a la velocidad de conducción y la relación con el diámetro de fibras sensoriales y la profundidad de grabación fue estimada en base a la penetración del micro-electrodo y el ángulo de inserción. Estos datos se muestran en la figura 3.1, señalando para diámetro, profundidad de grabación y umbral motor la media, intervalos de confianza y rango en el que se encuentran los ejemplares utilizados en cada caso y cuantos de estos se incluyeron.

SCS Waveform	n	Fiber Diameter, μm			Recording Depth, μm			Motor Threshold, mA	
		Mean	95% CI	Range	Mean	95% CI	Range	Mean	95% CI
Biphasic 10 kHz	39	3.3	3.0–3.7	1.8–5.6	221	179–264	12–531	1.2	1.0–1.4
SNI 10 kHz	21	3.4	3.1–3.7	2.1–4.7	127†	106–148	59–223	0.95	0.9–1.0
Sinusoid 10 kHz	21	3.6	3.1–4.1	1.8–5.6	206	155–257	13–385	1.2	0.9–1.5
Biphasic 1 kHz	18	3.8	3.1–4.5	2.0–6.6	258	189–328	29–591	0.6‡	0.3–0.8
Biphasic 5 kHz	18	3.8	3.2–4.5	2.0–6.6	179	128–230	29–408	1.25	1.0–1.5
Biphasic 20 kHz	21	3.6	3.1–4.1	2.0–5.8	185	136–234	29–408	1.3	1.1–1.6
Biphasic 50 Hz	13	4.3*	3.4–5.3	1.8–7.6	171	110–231	55–369	0.082‡	0.07–0.09

* $P = 0.01$; † $P = 0.006$; ‡ $P < 0.0001$ vs. biphasic 10 kHz (1-way ANOVA with Tukey's HSD). CI, confidence interval; SCS, spinal cord stimulation; SNI, spared nerve injury.

Figura 3.1: Propiedades de axones de la columna dorsal registrados durante pruebas de SCS.

Fuente: Crosby et al. (2017)

En cada prueba se identificaron los impulsos producto de la estimulación antidrómica, clasificando el resto de los impulsos como producto de la estimulación de la espina a frecuencia de kilohertz (KHF-SCS). Para cuantificar estos últimos se graficaron curvas de la tasa de disparo de impulsos producto del KHF-SCS mediante una ventana deslizante de ancho de 1s y paso de 0.1 s. Además se utilizan histogramas de tiempo periestímulo promediados para 30 segundos de SCS para nueve amplitudes de este y suavizados utilizando un núcleo gaussiano con $\sigma = 0.5$.

Entre los resultados obtenidos en este estudio se encuentran aquellos para la estimulación a 10 kHz. Para una muestra de 39 individuos se obtuvieron resultados en los cuales la conducción del potencial antidrómico no se veía afectado (figura 3.2A), o era bloqueado por el SCS (figura 3.2B), donde la señal antidrómica y la correspondiente al SCS se detienen en los primeros segundos, luego seguidas únicamente por artefactos de la estimulación.

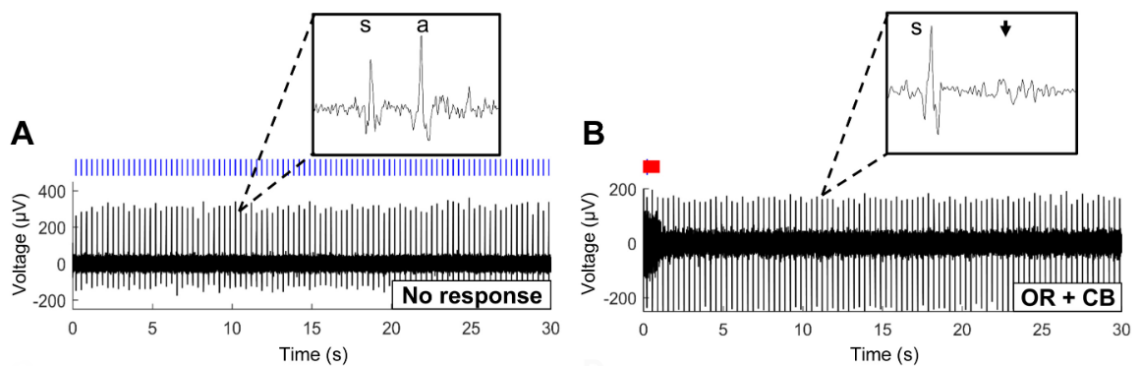


Figura 3.2: Patrones de disparo mostrados en respuesta a SCS bifásico a 10 kHz, para el caso sin efectos sobre el potencial antidrómico (A) y con respuesta de activación inicial y bloqueo de conducción (B).

Fuente: Crosby et al. (2017)

3.2 MODELO DE FIBRA NERVIOSA DE MCINTYRE, RICHARDSON Y GRILL (MRG)

McIntyre et al. (2002) definen un modelo de cable doble de fibras nerviosas motoras de mamíferos, representando de forma explícita los nodos de Ranvier y secciones paranodales e internodales del axón. La geometría y dinámicas de la membrana fueron basadas en mediciones experimentales de humanos, gatos y ratones obtenidas de distintos estudios anteriores. En la figura 3.3 se observa en la parte superior la representación geométrica de la fibra nerviosa, indicando en el centro el nodo de Ranvier, los segmentos de acoplamiento de mielina (MYSA), segmentos paranodales principales (FLUT) y segmentos internodales (STIN). En la parte inferior

de la figura se muestra la representación eléctrica del modelo, en la izquierda representando los compartimientos fuera del nodo mediante una estructura de cable doble, con conductancias lineales representando la capa de mielina (G_m en paralelo con C_m) y el axolema internodal (G_i en paralelo con C_i). Los compartimientos nodales en cambio se representan con las conductancias de canales rápidos y persistentes de sodio, canales lentos de potasio y un canal de fuga lineal en paralelo con la capacitancia del nodo (C_n).

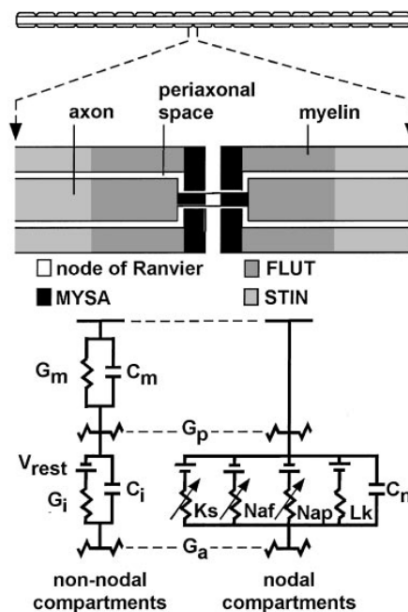


Figura 3.3: Diagrama del modelo de fibra nerviosa MRG.
Fuente: McIntyre et al. (2002)

En las figuras 3.4 y 3.5 se muestran los parámetros morfológicos y electrofisiológicos respectivamente. En la literatura en la cual se basaron los parámetros no existen mayores detalles respecto al error de medición. Para los parámetros eléctricos, en Huxley & Stämpfli (1949), la capacitancia media de la mielina, MYSA, FLUT y STIN tiene un error estándar que puede ir entre un 3% a un 34% del valor observado, y la conductancia media un error estándar entre un 12% y 30%, dependiendo de la especie de rana utilizada en el experimento, mientras Scholz et al. (1993) para el caso de la conductancia de los canales de potasio muestra un error estándar de entre un 5% y 10% del valor medido. En el caso de los parámetros morfológicos, en Rydmark (1981), los diámetros nodales se encuentran en un rango de $\pm 1.06\mu m$ para los menores diámetros, y $\pm 2.41\mu m$ para los mayores.

	Fiber Diameter								
	5.7	7.3	8.7	10.0	11.5	12.8	14.0	15.0	16.0
Node-node separation ^a	500	750	1,000	1,150	1,250	1,350	1,400	1,450	1,500
Number of myelin lemella ^b	80	100	110	120	130	135	140	145	150
Node length ^{c,d}	1	1	1	1	1	1	1	1	1
Node diameter ^{c,d}	1.9	2.4	2.8	3.3	3.7	4.2	4.7	5.0	5.5
MYSA length ^e	3	3	3	3	3	3	3	3	3
MYSA diameter ^{b,e}	1.9	2.4	2.8	3.3	3.7	4.2	4.7	5.0	5.5
MYSA periaxonal space width ^e	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002
FLUT length ^e	35	38	40	46	50	54	56	58	60
FLUT diameter ^{b,e}	3.4	4.6	5.8	6.9	8.1	9.2	10.4	11.5	12.7
FLUT periaxonal space width ^e	0.004	0.004	0.004	0.004	0.004	0.004	0.004	0.004	0.004
STIN length ^a	70.5	111.2	152.2	175.2	190.5	205.8	213.5	221.2	228.8
STIN diameter ^b	3.4	4.6	5.8	6.9	8.1	9.2	10.4	11.5	12.7
STIN periaxonal space width ^e	0.004	0.004	0.004	0.004	0.004	0.004	0.004	0.004	0.004

Figura 3.4: Parámetros morfológicos del modelo MRG. Valores expresados en μm .

Fuente: McIntyre et al. (2002)

Nodal capacitance ^a (c_n)	2 $\mu F/cm^2$
Internodal capacitance ^b (c_i)	2 $\mu F/cm^2$
Myelin capacitance ^{c,d,e} (c_m)	0.1 $\mu F/cm^2$
Axoplasmic resistivity (ρ_a)	70 Ωcm
Periaxonal resistivity (ρ_p)	70 Ωcm
Myelin conductance ^{c,d,e} (g_m)	0.001 S/cm ²
MYSA conductance (g_s)	0.001 S/cm ²
FLUT conductance (g_f)	0.0001 S/cm ²
STIN conductance (g_i)	0.0001 S/cm ²
Maximum fast Na ⁺ conductance ^{f,g,h} ($g_{Na f}$)	3.0 S/cm ²
Maximum slow K ⁺ conductance ^{f,h,i} ($g_{K s}$)	0.08 S/cm ²
Maximum persistent Na ⁺ conductance ^j ($g_{Na p}$)	0.01 S/cm ²
Nodal leakage conductance (g_{Lk})	0.007 S/cm ²
Na ⁺ Nernst potential ^k (E_{Na})	50.0 mV
K ⁺ Nernst potential ^k (E_K)	-90.0 mV
Leakage reversal potential (E_{Lk})	-90.0 mV
Rest potential ^k (V_{rest})	-80.0 mV

Figura 3.5: Parámetros eléctricos del modelo MRG.

Fuente: McIntyre et al. (2002)

Dentro de los resultados obtenidos con este modelo permitieron reproducir distintos conjuntos de datos experimentales, por ejemplo, soportando la hipótesis de que existe una contribución de canales activos y pasivos al pospotencial de depolarización, y solo activos a la hiperpolarización. No obstante el modelo presentó limitaciones según los mismos autores. Primero, los tipos de canales iónicos, densidades y dinámicas de membranas no se encontraban completamente caracterizadas y la representación de estos fue mediante datos experimentales de algunos canales iónicos presentes, lo cual logra representar distintos tipos de excitación, con lo que se asume un canal iónico puede ser equivalente a múltiples. La segunda limitación corresponde a la representación de la membrana axonal bajo la mielina como una conductancia lineal en paralelo, existiendo diferentes tipos de conductancia en la realidad, pero se decide usar

la línea por temas de simpleza computacional y la falta de caracterización de sus propiedades dinámicas. El uso de esta conductancia resulta en ligeras discrepancias respecto a algunos registros experimentales como la acomodación de la frecuencia de impulsos nerviosos. La tercera limitación es la falta de representación de acumulación de potasio en el espacio extracelular, lo cual afecta particularmente cuando un axón se activa a altas frecuencias por periodos extendidos de tiempo, lo cual incrementa la concentración de potasio extracelular y esto incrementa la excitabilidad del axón. Revisando los resultados descritos en Bostock & Bergmans (1994) en lo cual se basa el trabajo de McIntyre et al. (2002) para esto último, los periodos entre los cuales se realizaron los experimentos son de al menos un minuto a 300 impulsos por segundo.

3.3 USO DEL MODELO DE FIBRA NERVIOSA MRG EN LITERATURA

Este modelo fue utilizado en Gaines et al. (2018) como referencia para modelar fibras nerviosas motoras y sensoriales de axones en el brazo, donde se incluyó un canal rápido de potasio en el nodo y la adición de canales internodo. En el caso de las fibras motoras, se utilizaron los mismos parámetros que el modelo MRG en lo posible, exceptuando los canales agregados para los cuales se utilizó el modelo descrito por Howells et al. (2012) como referencia. Este mismo modelo MRG revisado para fibras motoras fue utilizado como base para el modelo de fibra nerviosa, donde los parámetros se calcularon usando la razón entre esos canales descrita por Howells. Ambos modelos fueron validados usando simulaciones de un axón centrado en un solo nervio fascicular, evaluando velocidad de conducción, forma del potencial de acción y la relación entre fuerza y duración, además de análisis de sensibilidad para determinar los efectos de cada conductancia en el umbral de activación, esto en comparación con resultados experimentales registrados en la literatura, donde se obtuvieron resultados similares durante las pruebas.

En Durá et al. (2019) el modelo se utiliza como base para determinar la activación de fibras nerviosas $A\beta$ localizadas en la columna y raíz dorsal, del cual se derivó una ecuación para determinar el potencial de membrana en cada nodo. Los parámetros morfológicos utilizados corresponden aproximadamente a los originales descritos por McIntyre et al. (2002) para fibras de $12.8 \mu\text{m}$ en el caso de fibras de la columna dorsal, y $15.0 \mu\text{m}$ para fibras de la raíz dorsal. Respecto a los parámetros eléctricos, se usó como base los del modelo original, modificando la conductancia de los canales de sodio persistente (0.005 S/cm^2) y el canal de fuga (0.08 S/cm^2), reduciendo el potencial de Nernst del canal de potasio (-84 mV) y del canal de fuga (-83.38 mV), mientras que el potencial de reposo se deja en -82 mV .

3.4 OPTIMIZACIÓN DE PARÁMETROS EN MODELOS DE FIBRA NERVIOSA

En Gaines et al. (2018) el modelo de neurona desarrollado utiliza los mismos parámetros originales del modelo MRG u otros descritos en la literatura, ambos obtenidos por medición experimental, y de igual manera en el caso del modelo utilizado en Durá et al. (2019). Existe una necesidad de encontrar otros parámetros para el modelo MRG que se adapten a los resultados de un determinado tipo de fibra nerviosa, es correspondiente revisar metodologías de búsqueda de parámetros para este tipo de modelos. El trabajo actual con mayor relevancia en este sentido se encuentra descrito por Xu et al. (2019) para la optimización de parámetros de un modelo MAT (Multi-timescale Adaptative Threshold) de neurona cortical (presente en el cerebro) mediante el uso de una estrategia evolutiva para adaptación de matrices de covarianza (referida como CMA-ES). En este trabajo se realiza una comparación de algoritmos evolutivos, analizando el caso de los algoritmos genéticos y CMA-ES, concluyendo que el rendimiento del algoritmo CMA-ES es en promedio mejor que el de métodos convencionales como algoritmos genéticos o métodos de Nelder-Mead (Xu et al., 2019), pero la capacidad de ajuste de este depende en gran parte de las condiciones iniciales, en algunos casos dando peores resultados que los otros métodos. Ante esto último, se propone un algoritmo CMA-ES agregado, donde se ejecuta este algoritmo múltiples veces para acercarse a un óptimo global. Por lo general, en distintos trabajos se menciona CMA-ES como uno de los mejores optimizadores para resolver problemas difíciles de caja negra, véase, donde el problema es representado por una función no lineal, no convexa, con ruido, entre otros factores que dificulten conocer la forma de la función y usar un optimizador en base a esta (Varelas et al., 2018; Szykiewicz, 2018).

Rumbell et al. (2016) plantea una metodología de optimización de la conductancia de canales iónicos utilizando una aproximación mediante el uso de múltiples funciones de error. Se utilizaron 4 tipos de funciones definidas para neuronas clasificadas en jóvenes y viejas, llegando a tener alrededor de 20 funciones. Estas se combinan mediante una ponderación encontrada utilizando maximización de distancias mínimas para muestreo por hipercubo latino (LHS), con tal de evitar que algunas fuentes de error se superpongan a otras. Estas funciones de error son utilizadas tanto para la optimización mediante un algoritmo de evolución diferencial (DE) utilizando solo una función objetivo, y optimización mediante evolución diferencial con ordenamiento en base a soluciones no denominadas (NDSE) utilizando múltiples funciones objetivo. Los resultados obtenidos muestran la utilidad del uso de LHS para encontrar pesos de múltiples funciones objetivo para utilizarse ponderadas como único objetivo o separadas en multi-objetivo, no obstante los resultados obtenidos para las pruebas con NSDE no logran superar en eficiencia la metodología de objetivo único utilizada.

En Loewe et al. (2016) se presenta también una aproximación a la optimización de parámetros de formulaciones de corrientes iónicas, para lo cual se decide utilizar dos métodos de optimización, por un lado un algoritmo heurístico de optimización de partículas (PSO), y otro basado en gradiente de región de confianza reflectante (TRR). Dado que en pruebas sintéticas, el algoritmo TRR era superado por el de PSO y se veía de forma frecuente atrapado en mínimos locales, se realiza una aproximación utilizando primero el algoritmo PSO para encontrar un conjunto de soluciones, las cuales se ingresan las mejores como soluciones iniciales al algoritmo TRR (Figura 3.6A), lo cual si bien entregó mejores resultados que el uso de cada algoritmo de forma separada, no fueron satisfactorios. Ante esto ultimo se presenta un enfoque híbrido como se muestra en la figura 3.6B, donde cada conjunto de soluciones generado en una iteración de PSO se entrega como soluciones iniciales al algoritmo TRR, el cual se ejecuta por K iteraciones, entregando las soluciones resultantes como entrada nuevamente al algoritmo PSO inicial, formando un ciclo de L iteraciones. Una vez terminada la ultima iteración se TRR, se pasa el conjunto de soluciones a un segundo algoritmo TRR hasta converger a un resultado definitivo. Esta aproximación entregó mejores resultados que las anteriores, incluyendo en condiciones con ruido incorporado, dando estimaciones confiables de los parámetros.

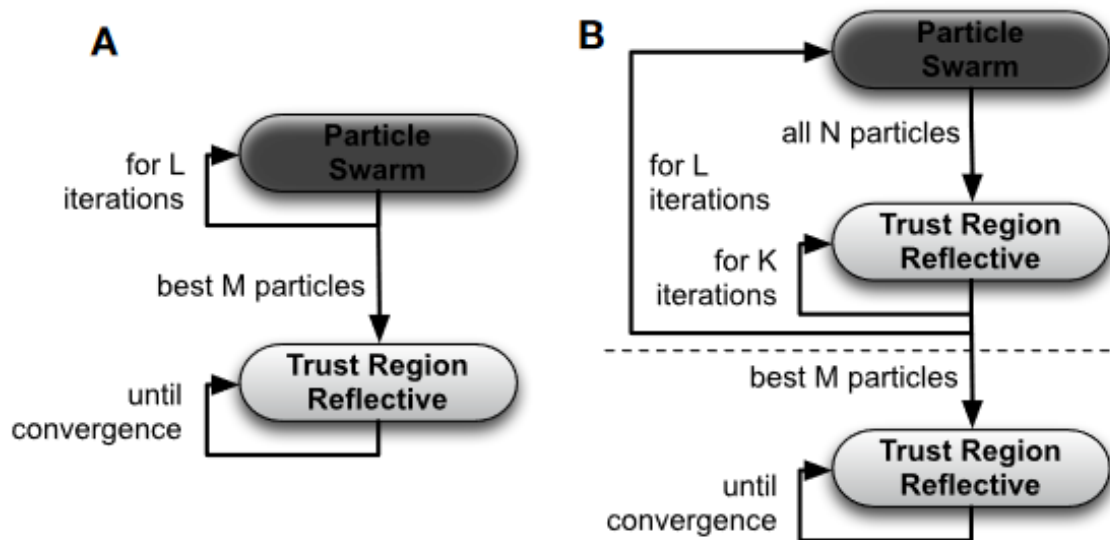


Figura 3.6: Diagrama del uso de PSO y TRR en forma de 2 etapas (A) e híbrida (B).

Fuente: Loewe et al. (2016)

La importancia de todos estos métodos basados en metaheurísticas radica primero que todo en su utilidad para encontrar soluciones aproximadas a problemas de optimización complejos donde el uso de técnicas tradicionales de optimización exacta resultan en un coste computacional demasiado alto para su realización. Dada la formulación del modelo MRG

como un modelo eléctrico con componentes no lineales y la cantidad de parámetros que se consideran en este, la optimización de este se puede considerar como un problema complejo para el uso de técnicas tradicionales exactas, prefiriendo el uso de técnicas heurísticas que entreguen una solución aproximada a menor coste computacional. Otro punto a considerar de las metaheurísticas, es el ser en gran parte independientes del problema, y es posible utilizarse la misma para optimizar distintos modelos. Dado esto último, es posible utilizar una función objetivo calculada en base a un programa que internamente implemente los cálculos necesarios, mientras que desde el algoritmo de optimización solo se llama este modificando los parámetros necesarios. La pregunta que queda es si al aplicar estos métodos basados en metaheurísticas para optimizar el modelo MRG es posible llegar a un conjunto de parámetros que ajusten de manera precisa este modelo a ciertos tipos de neuronas como son las representadas en el trabajo de Crosby et al. (2017), siendo posible reproducir el comportamiento de estas ante la aplicación de SCS utilizando el modelo ajustado.

CAPÍTULO 4. DESARROLLO DE LA INVESTIGACIÓN

La investigación se realizó en torno al ajuste de los parámetros del modelo MRG mencionado en la sección 3.2, el cual se modela como un problema de optimización para el cual se desarrolla una pieza de software de experimentación que implementa el algoritmo CMA-ES integrado con NEURON para encontrar soluciones de este de mejor calidad a las conocidas, esto en conjunto a herramientas de paralelización para su uso en un clúster.

4.1 ESTRUCTURA DEL SOFTWARE EXPERIMENTAL DESARROLLADO

El software experimental que implementa el método de optimización aplicado al problema en cuestión consta de tres partes principales como se muestran en la figura 4.1: un modelo con sus distintos parámetros independientes y forma de evaluación, un simulador que implementa este modelo en forma computacional retornando el resultado de la evaluación, y un optimizador que hace uso de este simulador para ir evaluando distintos conjuntos de parámetros e ir aproximando soluciones de mejor calidad en base a los resultados obtenidos durante cada iteración. Este software, particularmente desde el punto de vista del optimizador y simulador, se implementó de forma escalable con tal de ser utilizado en un clúster, permitiendo obtener resultados de forma acelerada según que tantos recursos computacionales tengamos disponibles.

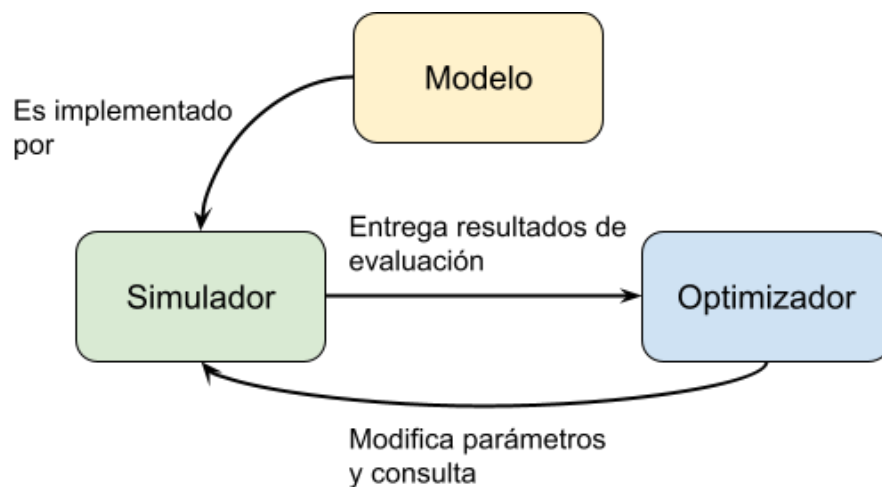


Figura 4.1: Diagrama de la estructura general del software desarrollado.

Fuente: Elaboración propia (2021)

4.1.1 Modelo y simulador

El modelo del problema a optimizar se basa en el modelo de fibra nerviosa MRG mencionado en el subcapítulo 3.2. Este modelo y distintas mecánicas relacionadas a este se encuentran actualmente implementadas internamente en NEURON como parte del simulador, por lo que cálculos relacionados a estas partes se abstraen y se consideran en forma de caja negra, particularmente por la cantidad y complejidad de estos. No obstante, todavía tenemos control sobre distintos parámetros que se utilizan en este modelo, y que es lo necesario para poder aplicar una metaheurística como método de optimización.

Dentro de los parámetros definidos en la sección 1.5.3, se consideran los siguientes parámetros electrofisiológicos a optimizar.

- Capacitancia nodal e internodal (C_{dc})
- Conductancia MYSA (g_{MYSA})
- Conductancia máxima NA^+ rápido (g_{na})
- Conductancia máxima K^+ lento (g_k)
- Conductancia máxima NA^+ persistente (g_{nap})
- Conductancia del filtrado nodal (g_l)

Respecto a los parámetros morfológicos, se consideran los siguientes parámetros según se extrapolan en la tabla 1.2, todos en función de un diámetro de fibra $FiberD$ y sus constantes de extrapolación respectivas, estos últimos siendo los parámetros a optimizar.

- Separación internodal (δ_x , en función de δ_x^A y δ_x^B)
- Diámetro nodal/MYSA ($NodeD$, en función de $NodeD^A$ y $NodeD^B$)
- Diámetro del Axón/FLUT/STIN ($AxonD$, en función de $AxonD^A$ y $AxonD^B$)
- Largo FLUT (l_{FLUT} , en función de l_{FLUT}^A y l_{FLUT}^B)
- Número de laminillas de mielina (N_{ml} , en función de N_{ml}^A y N_{ml}^B)

Los parámetros correspondientes a la longitud STIN se omiten ya que esta se calcula en función del resto de los parámetros, como se muestra en la ecuación (4.1), considerando parámetros base como la longitud del nodo igual a 1.

$$l_{STIN} = \frac{\delta_x - 7 - 2l_{FLUT}}{6} \quad (4.1)$$

Respecto a las restricciones, se utilizan aquellas señaladas en las tablas 1.1 y 1.3 para valores máximos y mínimos de cada parámetro. En el caso de l_{STIN} , dado que este parámetro se calcula en función de otros, pero sigue siendo un parámetro físico el cual tiene limites, se definen cotas inferiores en función al menor y mayor diámetro posible de muestrear que se utilizará ($FiberD_{min}$ y $FiberD_{max}$), y las constantes base definidas para su función de extrapolación, calculando una longitud base en cada caso, y el limite inferior de l_{STIN} siendo el 20% de este valor para la menor y mayor fibra posible como se muestra en (4.2).

$$\begin{aligned} 0.2(15.751FiberD_{min} - 3.848) &\leq l_{STIN}(FiberD_{min}) \\ 0.2(15.751FiberD_{max} - 3.848) &\leq l_{STIN}(FiberD_{max}) \end{aligned} \quad (4.2)$$

Aparte de todo lo que respecta al modelo de fibra nerviosa MRG, se debe considerar también la metodología de evaluación que lo envuelve. Dado que lo que se busca es encontrar un conjunto de parámetros que logre recrear el comportamiento descrito por Crosby et al. (2017), necesitamos en parte considerar esto en el modelo y simulador. Nos interesa en particular los resultados experimentales que se muestran en la figura 1.3, donde se exhiben patrones de activación en respuesta al SCS durante unos segundos, aumentando la intensidad según aumenta la amplitud. Por ende, nuestro enfoque es en generar estas curvas que son de densidad de puntas detectadas por segundo a lo largo del tiempo.

Para la generación de estas curvas consideramos primero los individuos bajo los que se realizaron el experimento. Según Crosby et al. (2017), se utilizaron 39 ejemplares de fibras nerviosas de ratones vivos, los que se sometieron a SCS bifásico a 10-khz, se registró la actividad eléctrica y mediante un detector de puntas se contabilizaron estas a lo largo del tiempo. Para la simulación de esto se utiliza NEURON, el cual recrea para una fibra y conjuntos de parámetros dados la aplicación de SCS, y posteriormente mediante Python se toman los resultados de los cuales se detectan y contabilizan las puntas. Esto lo realizamos para un conjunto de 30 fibras con diámetro muestreado desde en una distribución normal equivalente a los intervalos de confianza dados por Crosby, todas con el mismo conjunto de parámetros electrofisiológicos y morfológicos del modelo. La reducción en la cantidad de fibras es para reducir el coste computacional del simulador, pero manteniendo un grado de normalidad en la muestra. La aplicación de SCS para las 30 fibras se realiza para distintos intervalos de amplitud, siendo para este estudio de 0.2 a 0.4 mA, 0.4 a 0.6 mA y 0.6 a 0.8 mA, nuevamente utilizando solo un conjunto de los casos para reducir el coste computacional del simulador. Bajo el mismo principio de reducir el tiempo de computo, la simulación solo se hace para los primeros 2.8 segundos, lo suficiente para registrar la curva esperada y su descenso, donde después se mantiene constante y por ende no se considera en la simulación.

Dado que en el experimento original la curva de para cada rango de amplitud se hizo en base a histogramas de tiempo de peristímulo y luego suavizados con un kernel de σ de 0.5 s, como desconocemos la metodología de construcción del histograma original, generamos para cada rango de amplitud una curva equivalente en base a estimación de densidad de kernel para obtener una función de densidad de probabilidad, la cual multiplicamos por la cantidad de puntas totales promediadas por la cantidad de fibras, dando la densidad de puntas promedio por segundo a lo largo del tiempo.

Ya con el calculo de cada curva de densidad de puntas promedio, cada una se compara con una curva cercana a la esperada $f_E(A, t)$. según los resultados de Crosby et al. (2017), la cual se define en la ecuación (4.3), con A el valor máximo que toma la curva a los 0.42 segundos, siendo 16 para el rango de 0.2 a 0.4 mA, 20 para 0.4 a 0.6 mA y 24 para 0.6 a 0.8 mA, durante una instancia de tiempo t .

$$f_E(A, t) = \frac{A80^{t-0.42}}{(1 + 20^{t-0.42})^2} \quad (4.3)$$

La comparación se realiza mediante el error cuadrático medio entre el valor generado para la curva y la curva esperada correspondiente según el rango de amplitud. Esto para un rango de 100 puntos en el tiempo generados de forma equidistante entre el comienzo y final de la evaluación. Cómo esto se debe calcular para 3 curvas diferentes, los resultados de cada una de estas se promedia, con lo que se tiene el error final. En la ecuación (4.4) se describe el calculo del error para un conjunto de parámetros P, considerando una función $f_d(P, A, t)$ que describe la curva de densidad para el conjunto de parámetros dado, un rango de amplitudes A y un instante en el tiempo t, y $f_E(A, t(i))$ que define la función de probabilidad esperada.

$$E(P) = \frac{1}{3} \sum_{A \in \text{amps}} \frac{1}{100} \sum_{i=1}^{100} [f_d(P, A, t(i)) - f_E(A, t(i))]^2 \quad (4.4)$$

Finalmente, en base a esta función de error podemos definir el problema de optimización del modelo para un conjunto de parámetros P que incluye todos los parámetros morfológicos y electrofisiológicos, esto descrito como se muestra en (4.5). Hay que destacar que las restricciones correspondientes salvo las dos últimas corresponden a valores aproximados que marcan lo esperable y/o para acotar el espacio de búsqueda dentro de un margen, por lo que pueden diferir de lo que podría verse en forma práctica, ya sea aceptando o restringiendo en mayor medida los valores.

$$\begin{aligned}
& \text{Minimizar } E(P) \\
& 0.5 \leq C_{dc} \leq 3 \\
& 0.0001 \leq g_{MYSA} \leq 0.01 \\
& 0.01 \leq g_{na} \leq 10 \\
& 0.001 \leq g_k \leq 1 \\
& 0.001 \leq g_{nap} \leq 1 \\
& 0.00001 \leq g_l \leq 0.01 \\
& 24.295 \leq \delta_x^A \leq 170.065 \\
& 13.5115 \leq \delta_x^B \leq 94.5805 \\
& 0.2867 \leq NodeD^A \leq 2.0069 \\
& 0.0250 \leq NodeD^B \leq 0.1752 \\
& 1.4979 \leq AxonD^A \leq 10.4854 \\
& 0.3125 \leq AxonD^B \leq 2.1873 \\
& 5.9663 \leq l_{FLUT}^A \leq 41.7638 \\
& 1.9501 \leq l_{FLUT}^B \leq 13.6509 \\
& 14.645 \leq N_{ml}^A \leq 102.515 \\
& 3.7173 \leq N_{ml}^B \leq 26.0208 \\
& 0.2(15.751FiberD_{min} - 3.848) \leq l_{STIN}(FiberD_{min}) \\
& 0.2(15.751FiberD_{max} - 3.848) \leq l_{STIN}(FiberD_{max})
\end{aligned} \tag{4.5}$$

4.1.2 Implementación del optimizador

Ya con el modelo y simulador de este listo, el siguiente paso es definir como se implementa la metaheurística para encontrar una mejor solución al problema de optimización en base al comportamiento eléctrico esperado mostrado en las curvas de densidad de impulsos, en este caso CMA-ES. Para su implementación se utilizó el framework DEAP (Distributed Evolutionary Algorithms in Python) (Fortin et al., 2012) que nos entrega una serie de herramientas para implementar este algoritmo, incluyendo la estrategia en sí la cual se basa en el paradigma de generación y actualización, implementando en su base todo lo necesario para generar una población en base al muestreo de una distribución Normal, y el ajuste de esta mediante la actualización de la matriz de covarianza, siguiendo lo descrito por Hansen & Ostermeier (2001).

Ya con esto, es necesario solamente adaptar este hacer uso del simulador del modelo a optimizar para el cálculo de la función objetivo, así como también implementar las restricciones necesarias.

Respecto a esto último, dado que CMA-ES funciona en base al muestreo de una distribución continua, para implementar restricciones nos basamos en el estudio realizado por Biedrzycki (2020) para el manejo de restricciones en CMA-ES, donde se comparan múltiples formas de manejar estas para distintos tipos de problemas, y utilizamos una mezcla de técnicas de penalización, proyección y remuestreo para las soluciones generadas que violen las restricciones, debido a mostrar una buena efectividad general, además de su factibilidad de implementación.

Como caso base, se tiene el de las soluciones donde uno o más parámetros de esta violan una restricción, pero no se alejan más allá del 75% del valor del límite que cruzan. En este caso se penaliza la evaluación de esta de forma proporcional al grado de alejamiento del límite. En la ecuación (4.6) se describe esto, donde P_L y P_H son el conjunto de parámetros de P que excede una cota inferior o superior respectivamente, $P(i)$ es el valor del parámetro i , $R_L(i)$ es la restricción inferior y $R_H(i)$ es la restricción superior, ambos para el mismo parámetro i .

$$E_{penalizado}(P) = E(P) \left(1 + \sum_{i \in P_L} \frac{P(i) - R_L(i)}{R_L(i)} + \sum_{j \in P_H} \frac{P(j) - R_H(j)}{R_H(j)} \right) \quad (4.6)$$

Para parámetros que exceden el 75% de la restricción que están cruzando, dado que se entra en terreno de valores negativos, o valor positivos que pueden crecer infinitamente, la evaluación de la solución se vuelve infactible tanto a nivel físico (un diámetro no puede ser negativo por ejemplo), o en materia computacional produciendo errores de ejecución. Para estos casos se utiliza una reparación del parámetro que excede la restricción, asignándole como valor la misma restricción, y posteriormente penalizándola de igual forma que el caso anterior.

Por último, está el caso de la longitud STIN que se calcula en función de otras variables y tiene sus propias restricciones. Dado que está en función a otros parámetros, realizar una reparación en base a proyección se torna demasiado complejo, y los valores que cruzan esta restricción pueden ser infactibles como para solo penalizarse, por lo que en estos casos se opta por remuestrear la solución problemática. Según Biedrzycki (2020) esta técnica es de manejo de restricciones es la que ofrece mejores resultados, pero dependiendo de la cantidad y tipo de restricciones, puede volverse demasiado costosa al tener que tratar de muestrear múltiples veces para encontrar una solución factible, lo cual empeora al comienzo de la ejecución, cuando la distribución inicial puede exceder en gran medida el espacio solución factible. Por esto, se deja el remuestreo para este caso puntual, y se opta por penalización para la mayoría de los casos, y reparación por proyección para el resto de casos extremos.

Ya con esto, el resto de los parámetros de la estrategia evolutiva se dejaron como

estaban, utilizando como centroide los parámetros base del modelo definidos por McIntyre et al. (2002). Como condiciones de parada nos basamos principalmente en lo descrito por Hansen (2016), donde se recomiendan distintos criterios, del cual el que mejor se adapta a la implementación del optimizador es el de estancamiento, definido de la siguiente forma: Se lleva un registro de los mejores valores de ajuste así como también de la mediana obtenida en cada generación, particularmente el 20% de las últimas iteraciones, siempre y cuando hayan transcurrido $120 + 30n/\lambda$ siendo n el número de parámetros en la función objetivo y λ el tamaño de población. En ambos registros se toma la mediana del 30% de últimos valores y se compara con la mediana del 30% del comienzo, y si en ambos casos la mediana del final no es mejor que la del comienzo, se detiene el algoritmo al considerarse estancado. Debido a que el tiempo de ejecución que toma el modelo usando NEURON puede ser demasiado como para alcanzar el número de iteraciones mínimo para usar este criterio dentro del tiempo en el que se realiza el trabajo, consideramos el criterio general mencionado por Hansen (2016) donde generalmente el algoritmo debe ser detenido cuando se considere un desperdicio de tiempo computacional seguir, ya sea por llegar a una solución lo suficiente buena o no obtener mejoras considerables. En base a esto, para este caso utilizamos el criterio de estancamiento como una medida a observar durante la ejecución del algoritmo para determinar el avance que lleva, sin considerar una cantidad mínima de iteraciones y deteniendo este según los recursos computacionales disponibles.

El funcionamiento general del algoritmo de optimización mediante CMA-ES se muestra en el Algoritmo 4.1, donde P_i es la población en una generación i , λ es el tamaño de población y S es el conjunto de parámetros siendo utilizados por la estrategia CMA-ES según lo definido en la sección 2.3.1, siendo:

- *InicializarEstrategia*(P_0) la función para inicializar la estrategia S .
- *GenerarSoluciones*(N, S) la función para generar el conjunto de soluciones en base al muestreo de la distribución normal, según los parámetros definidos en S , como se muestra en la función (2.3).
- *GenerarSolución*(S), función utilizada para generar una única solución, para reemplazar soluciones infactibles.
- *ActualizarEstrategia*(S, P_i), función utilizada para actualizar la estrategia en base a la población muestreada, en base a lo visto desde la ecuación (2.4) hasta (2.13).
- *Reparar*(S, P_i, p) es una función que representa el proceso de elegir un nuevo parámetro para reparar la solución infactible.

Algoritmo 4.1: Algoritmo de optimización mediante CMA-ES.

```
1  $S = \text{InicializarEstrategia}(P_0);$ 
2 para cada  $i$  en  $N$  hacer
3    $P_i = \text{GenerarSoluciones}(N, S);$ 
4   para cada solución  $s$  en  $P_i$  hacer
5     si  $s$  excede la restricción para  $l_{STIN}$  entonces
6        $P_i[s] = \text{GenerarSolución}(S);$ 
7     en otro caso
8       para cada parámetro  $p$  infactible en  $P_i$  hacer
9          $P_i[p] = \text{Reparar}(S, P_i, p);$ 
10      fin
11    fin
12  fin
13   $E_{\text{Penalizado}}(P_i);$ 
14   $S = \text{ActualizarEstrategia}(S, P_i);$ 
15  si Se cumple la condición de parada (Estancamiento) entonces
16     $\text{Terminar ciclo}$ 
17  fin
18 fin
Resultado: Mejor solución encontrada
```

4.1.3 Pruebas del optimizador con funciones conocidas

Previo al uso del optimizador con el problema objetivo, necesitamos realizar pruebas del funcionamiento de este con modelos conocidos, por ejemplo, usando funciones simples a optimizar cuyo óptimo se conozca. Esto con el objetivo de verificar el correcto funcionamiento del algoritmo CMA-ES en cuanto a converger a soluciones óptimas o cercanas a estas, pero también verificar el correcto funcionamiento a nivel de software del optimizador en cuanto a evitar errores, verificar el guardado de las soluciones al finalizar la ejecución, etc...

Las primera de las funciones a utilizar corresponde a la de Rosenbrock, la cual tiene un óptimo global en un valle con forma parabólica al cual, si bien es fácil encontrar el valle, converger al mínimo es difícil (Molga & Smutnicki, 2005). Esta se define en la ecuación 4.7, siendo d las dimensiones a evaluar, las cuales se restringen principalmente en $x_i \in [-2.048, 2.048]$. El mínimo global se define como $f(x^*) = 0$ en $x^* = (1, \dots, 1)$.

$$f(x) = \sum_{d=1}^{i=1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2] \quad (4.7)$$

La segunda a utilizar corresponde a la de Rastrigin, la cual cuenta con mínimos locales frecuentes distribuidos de forma regular (Molga & Smutnicki, 2005). Esta se define en la ecuación 4.8, siendo d las dimensiones a evaluar, las cuales se restringen principalmente en $x_i \in [-5.12, 5.12]$. El mínimo global se define como $f(x^*) = 0$ en $x^* = (0, \dots, 0)$.

$$f(x) = 10d + \sum_{d=1}^{i=1} [x_i^2 - 10\cos(2\pi x_i)] \quad (4.8)$$

Para cada función se realizarán 100 ejecuciones con centroides ubicados de forma aleatoria dentro del espacio donde se restringen las soluciones. Como criterio de parada se utiliza el de estancamiento definido previamente considerando el mínimo de generaciones realizadas. Para cumplimiento de las restricciones dadas se utiliza penalización de la misma forma que se establece para el optimizador principal. Para el análisis se utilizan gráficos de caja con tal de observar la convergencia de cada ejecución, donde se toma la mediana de los valores encontrados que lleva cada ejecución para una iteración g y se ubican en un mismo gráfico de caja para comparar la consistencia en la convergencia del algoritmo en cierta generación, lo cual se repite para múltiples generaciones. Lo mismo se realiza para estudiar el estancamiento que se lleva en cada ejecución para determinadas generaciones, para la cual se utiliza una medida de mejora relativa definida por la ecuación 4.9, donde para un historial de valores objetivos medianos obtenidos en la ejecución del algoritmo hasta una determinada generación se toma el 20% final de estos, y en este tramo se toma el la mediana del 30% inicial M_1 y la mediana del 30% final M_2 , de los cuales se toma una proporción y se obtiene la mejora relativa según cuanto disminuyó la última mediana respecto a la anterior. Valores cercanos a 1 indican una alta disminución (mejora) del valor objetivo, mientras que al contrario, valores cercanos a 0 indican poca disminución o estancamiento.

$$Mejora\ relativa = 1 - \frac{M_2}{M_1} \quad (4.9)$$

Todo esto nos permitirá después determinar si la ejecución del experimento principal utilizando NEURON se encuentra cercano a un punto de estancamiento al compararlo con lo visto usando estas funciones de prueba.

4.1.4 Ejecución del algoritmo en clúster

El algoritmo implementado fue desarrollado con la intención de ejecutarse de forma escalable, utilizando múltiples núcleos en un clúster, con tal reducir el tiempo requerido para obtener un resultado. Pensando en esto, DEAP permite el uso de SCOOP, o “Scalable COncurrent Operations in Python”, desarrollado por Hold-Geoffroy et al. (2014) como módulo de multiprocesamiento. Este funciona similar a otros módulos de Python que definen una función de mapeo la cual recibe una función y una lista de conjuntos de parámetros, y distribuye la tarea de evaluar cada conjunto en la función en múltiples procesadores, pero con la diferencia de integrarse directamente con el gestor de carga SLURM utilizado en clústers y supercomputadores, detectando este y automáticamente distribuyendo las tareas de evaluación no solo en los procesadores de la máquina local, sino que todos los que se encuentren disponibles dentro del clúster. Además, SCOOP permite el uso de la función de mapeo de forma anidada, es decir, una tarea creada por la función de mapeo que está siendo ejecutada en un nodo puede volver a llamar a la función de mapeo y crear nuevas tareas, ejecutándose en el mismo nodo y otros según sea necesario.

Mediante esto, es posible distribuir la evaluación de cada solución generada por el algoritmo de optimización en distintos computadores, pero dado que la cantidad de soluciones que se evalúan por generación son 50, no se aprovecharían computadores adicionales, por lo que también se aprovecha la funcionalidad de tareas anidadas para que durante la evaluación de cada solución, al realizar cada una de las 30 simulaciones que se hacen por rango de amplitud se haga mediante una subtarea, de esta forma permitiendo escalar a una mayor cantidad de computadores y acelerar la ejecución del algoritmo.

Definido esto, el clúster donde se ejecuta este software corresponde al del Departamento de Ingeniería Informática de la Universidad de Santiago de Chile, el cual está conformado por 2 máquinas Dell PowerEdge 2950 de 8 núcleos cada una, 3 máquinas Dell PowerEdge R715 de 16 núcleos cada una y 2 máquinas Dell PowerEdge R815 de 24 núcleos cada una. El tiempo de cómputo del algoritmo usando 106 núcleos es de aproximadamente 8 horas por iteración.

4.2 RESUMEN

Siguiendo un esquema de modelo, simulador y optimizador, se definen como se maneja el modelo MRG, qué parámetros se optimizarán de este y que se simulará sobre este,

siendo en este caso la estimulación de la fibra nerviosa para 30 casos de fibras muestreadas de forma aleatoria, esto para 3 rangos de amplitudes, permitiendo generar curvas de densidad de puntas promedio para cada rango, las cuales se comparan con las curvas obtenidas por Crosby et al. (2017) mediante el error cuadrático medio de cada curva, los cuales se ponderan para obtener el error de la solución lo que representa la función objetivo del modelo. Con esto, como simulador se utiliza NEURON el cual permite abstraernos de los cálculos necesarios para evaluar el comportamiento en cada fibra, y Python que sirve tanto como interfaz para la comunicación con NEURON, además de permitir muestrear cada fibra generada y manejar los resultados de las múltiples simulaciones para cada fibra y rango de amplitud, realizando los cálculos necesarios del error. Con esto, el optimizador que implementa CMA-ES se construye en base a la biblioteca de Python DEAP, que implementa la generación de soluciones y la actualización de la matriz de covarianza que define la distribución de muestreo, y de forma aparte se define el manejo de restricciones utilizando una combinación de penalización, reparación por proyección y remuestreo de la solución según sea el caso. El algoritmo primero se prueba mediante la optimización de funciones con soluciones conocidas para estudiar su comportamiento y verificar su funcionamiento adecuado, y posteriormente se ejecuta en un clúster mediante la utilización de la biblioteca de Python SCOOP que se integra directamente con el gestor de carga SLURM utilizado comúnmente por los clusters, permitiendo distribuir las tareas de evaluación de cada solución generada y las varias simulaciones sobre fibras que deben realizarse en la cantidad de nodos de computo que se encuentren disponibles, de esta forma reduciendo el tiempo requerido para obtener resultados con el algoritmo de optimización.

CAPÍTULO 5. RESULTADOS EXPERIMENTALES

Durante el desarrollo del proceso de experimentación, primero se realizaron de forma local pruebas del software implementado buscando soluciones a funciones con óptimos conocidos como se menciona en la sección anterior usando el algoritmo CMA-ES descrito, y de igual forma se probó el software con las mismas funciones en el clúster a ser utilizado durante los experimentos principales con tal de verificar su funcionamiento adecuado antes de realizar estos. Ya con las pruebas hechas, se realizaron dos experimentos utilizando el software implementado para llevar a cabo la optimización mediante CMA-ES, ejecutados utilizando 104 núcleos del clúster de la Universidad de Santiago de Chile. En el primero, se realizó la optimización solo variando los parámetros correspondientes a componentes electrofisiológicas y durante 45 generaciones debido a límites de tiempo, con tal de dejar tiempo para realizar una mayor cantidad de generaciones del segundo experimento, donde se ejecutó el algoritmo de optimización aplicado a todo el conjunto de parámetros tanto electrofisiológicos como morfológicos, durante un total de 68 generaciones.

Este capítulo está dividido en cuatro secciones: En la primera se analizan los resultados obtenidos de las pruebas utilizando el software implementado aplicado a funciones con óptimos conocidos, observando la convergencia del algoritmo CMA-ES para múltiples ejecuciones. En la segunda se estudia el proceso de optimización como tal, considerando el como fueron convergiendo las soluciones encontradas y el funcionamiento de la penalización en generar soluciones factibles. La tercera corresponde a un análisis de las soluciones, tanto la original como las encontradas como resultados de los experimentos y realizando comparaciones entre ellas sobre la respuesta eléctrica observada en el modelo de fibra para cada conjunto de parámetros. En la última sección se hace un estudio de sensibilidad sobre los parámetros del modelo de fibra nerviosa, con tal de apreciar como los parámetros que se modificaron durante el proceso de optimización influyen individualmente en la respuesta del modelo de fibra.

5.1 PRUEBAS DEL SOFTWARE USANDO FUNCIONES CONOCIDAS

5.1.1 Función Rosenbrock

En las pruebas realizadas optimizando la función de rosenbrock, de 100 ejecuciones realizadas, 99 convergieron a valores cercanos al mínimo global, existiendo solo un caso donde se convergió de forma apresurada a un mínimo local. Los casos restantes se muestran en la figura

5.1, donde se observa que la mayoría de los valores medianos se encuentran cercanos al mínimo $f(x^*) = 0$ en $x^* = (1, \dots, 1)$, con valores por el orden de 10^{-25} .

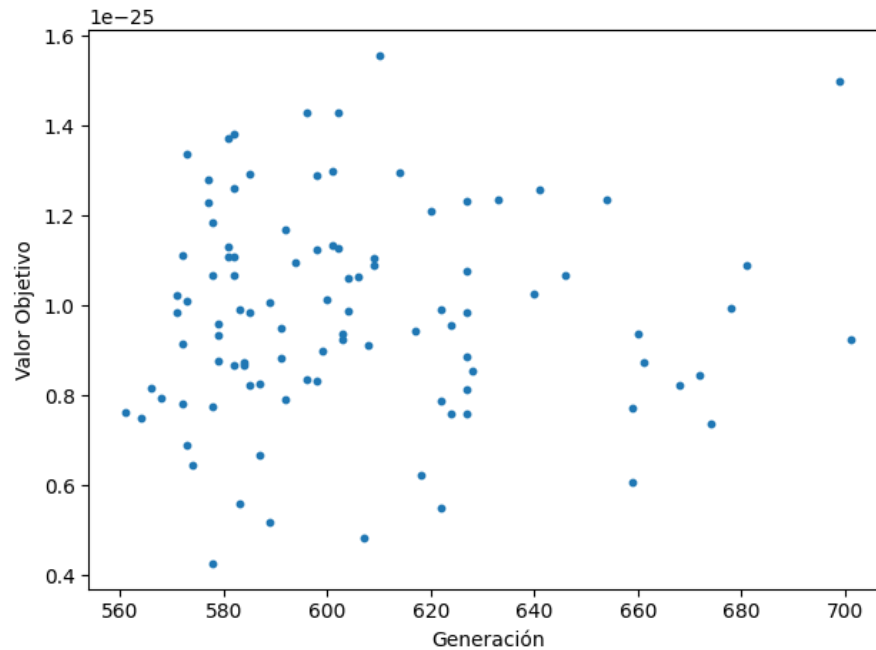


Figura 5.1: Gráfico de la mediana de valores objetivos obtenidos en cada ejecución optimizando la función de rosenbrock, respecto a la última generación realizada antes del estancamiento.
Fuente: Elaboración propia (2021)

En la figura 5.2 se observan los valores medianos obtenidos en las generaciones 100, 300, 500, 600 y 650 de cada experimento (mostrado solo de no haberse detenido antes de aquella generación). Se observa que al menos hasta la generación 100 todavía las soluciones no convergen al valor esperado, pero ya en la generación 300 se observa que los valores ya se encuentran cercanos al mínimo, y pasada la generación 500 comienzan a llegar a mantenerse en el orden de 10^{-25} .

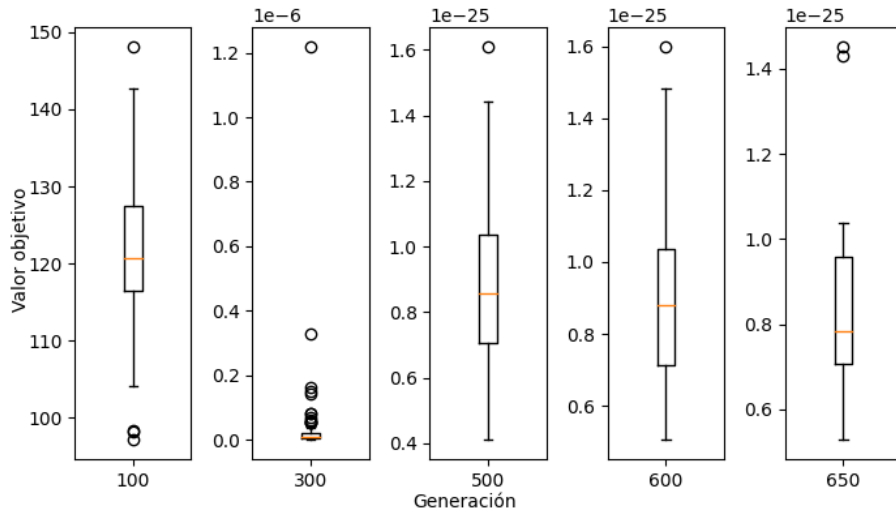


Figura 5.2: Gráficos de caja de la mediana de valores objetivos obtenidos en cada ejecución optimizando la función de rosenbrock, respecto a determinadas generaciones.

Fuente: Elaboración propia (2021)

Observando la figura 5.3 se ve que durante la generación 100 la mejora relativa se mantiene baja, subiendo abruptamente durante la generación 300 y 500, y ya pasada la 600 esta vuelve a bajar para las ejecuciones que todavía no terminan pero se aproximan a un punto de estancamiento. Esto nos indica que es posible que el algoritmo se mantenga por varias generaciones sin mejoras significativas, aumentando pasado cierto punto para luego converger a un mínimo.

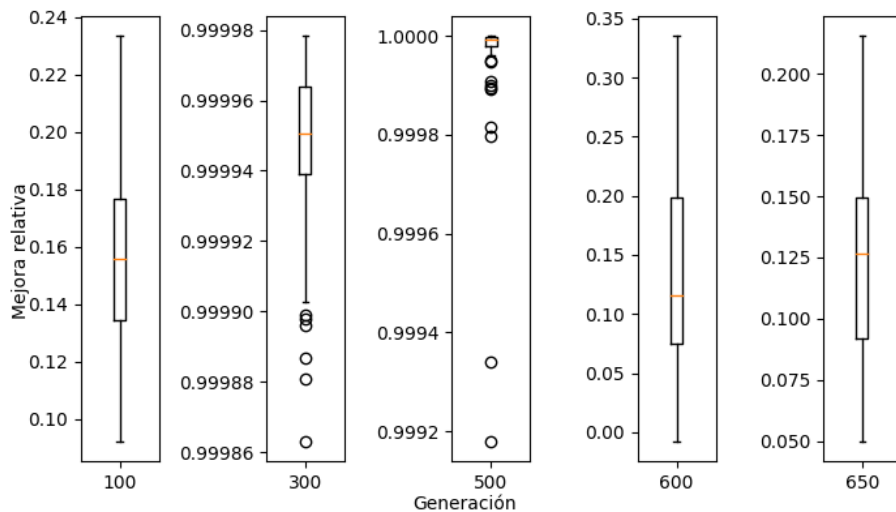


Figura 5.3: Gráficos de caja del estancamiento de la mediana de valores objetivos obtenidos en cada ejecución optimizando la función de rosenbrock, respecto a determinadas generaciones.

Fuente: Elaboración propia (2021)

5.1.2 Función Rastrigin

Para el estudio del comportamiento con esta función usamos el caso de 2 dimensiones tradicional y el caso de 16 dimensiones, similar al problema a resolver durante este trabajo. Para el caso de 2 dimensiones como se observa en las figuras 5.4 y 5.5 que el algoritmo tiende a converger rápidamente a mínimos, con la gran mayoría de las ejecuciones convergiendo al mínimo global pero algunos casos tienden a caer en mínimos locales y permanecer en ellos.

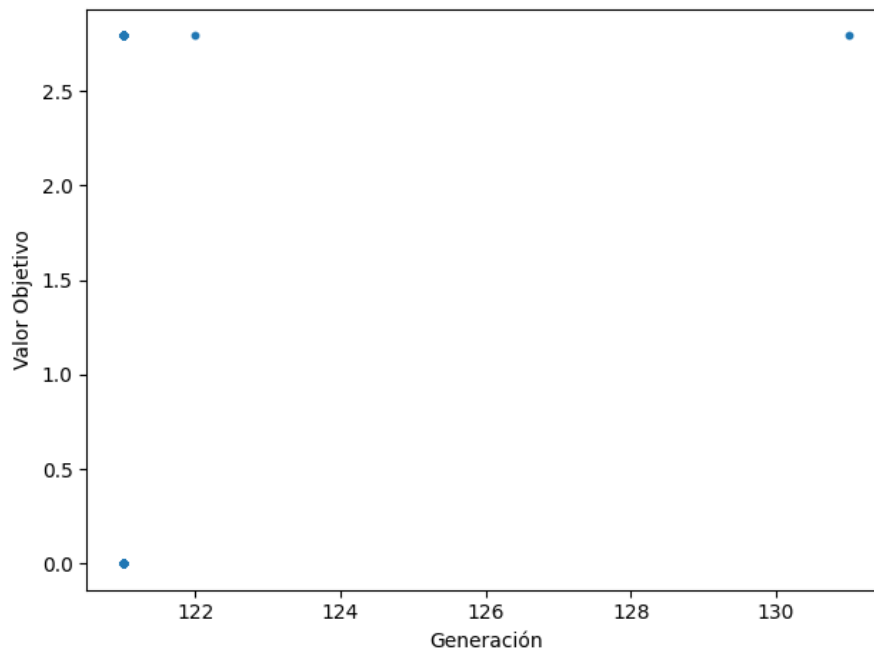


Figura 5.4: Gráfico de la mediana de valores objetivos obtenidos en cada ejecución optimizando la función de rastrigin para dos dimensiones, respecto a la última generación realizada antes del estancamiento.

Fuente: Elaboración propia (2021)

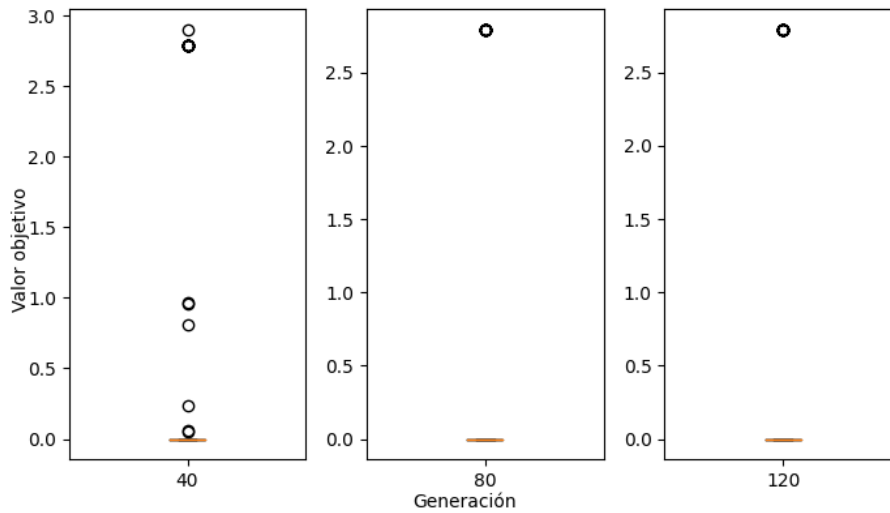


Figura 5.5: Gráficos de caja de la mediana de valores objetivos obtenidos en cada ejecución optimizando la función de rastrigin para dos dimensiones, respecto a determinadas generaciones.

Fuente: Elaboración propia (2021)

Pasando al caso de 16 dimensiones, en la figura 5.6 se observa que todas las ejecuciones del algoritmo esta vez convergieron únicamente a mínimos locales lejanos a l mínimo global (dentro del espacio de solución factible), alcanzando el estancamiento entre la generación 300 y 600. La figura 5.7 muestra que alrededor de la generación 100 hacía adelante los valores comienzan a estancarse dentro del mismo grupo de valores objetivos, lo cual se confirma observando la figura 5.8, donde ya en la generación 100 se ve una mejora baja y pasada la generación 200 esta empieza a ser cercana a 0, pasando el orden de 10^{-8} al llegar a la generación 300, lo cual indica prácticamente un estancamiento de las soluciones encontradas.

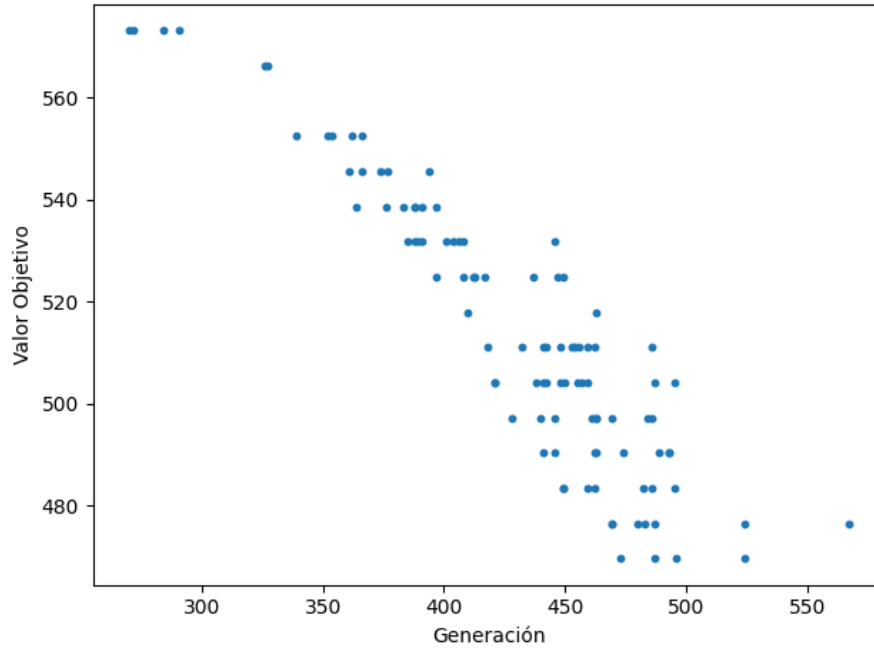


Figura 5.6: Gráfico de la mediana de valores objetivos obtenidos en cada ejecución optimizando la función de rastrigin para 16 dimensiones, respecto a la última generación realizada antes del estancamiento.

Fuente: Elaboración propia (2021)

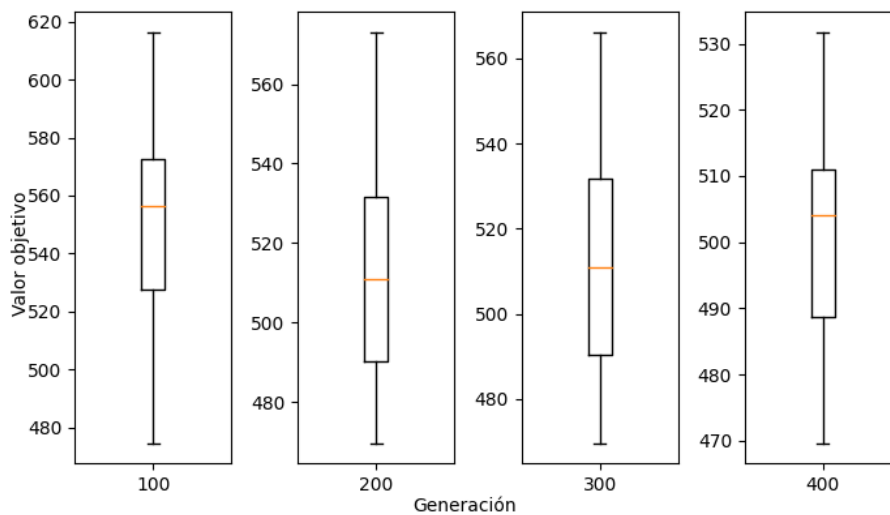


Figura 5.7: Gráficos de caja de la mediana de valores objetivos obtenidos en cada ejecución optimizando la función de rastrigin para 16 dimensiones, respecto a determinadas generaciones.

Fuente: Elaboración propia (2021)

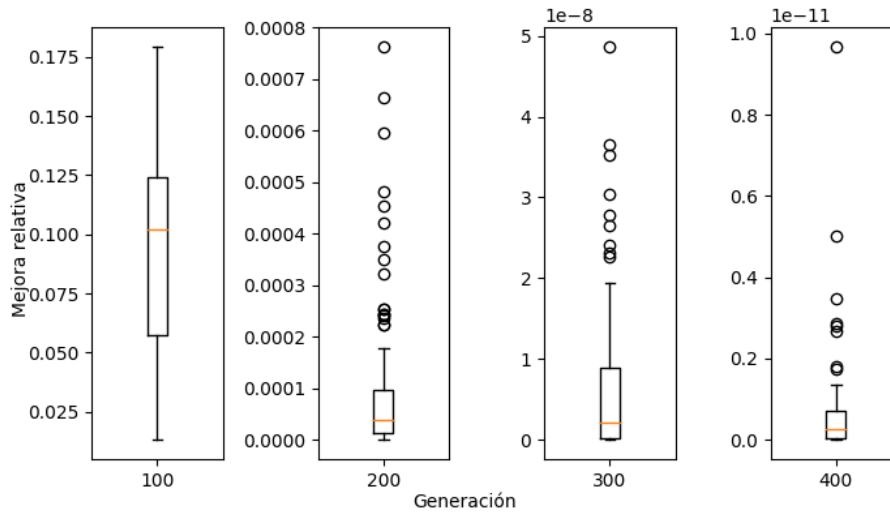


Figura 5.8: Gráficos de caja del estancamiento de la mediana de valores objetivos obtenidos en cada ejecución optimizando la función de rastrigin para 16 dimensiones, respecto a determinadas generaciones.

Fuente: Elaboración propia (2021)

En base a lo observado en estos experimentos, podemos esperar que a mayor dimensionalidad, problemas con un alto grado de mínimos locales como es la función de rastrigin pueden resultar en que el algoritmo se estanque fácilmente en estos, a diferencia de la función de rosenbrock donde teniendo una forma de valle y menor cantidad de mínimos locales, se logró aproximar una solución cercana a un mínimo global.

5.2 PROCESO DE OPTIMIZACIÓN

Para estudiar el comportamiento del proceso de optimización utilizando CMA-ES, primero nos enfocamos en el experimento realizado sobre únicamente los parámetros electrofisiológicos. En la figura 5.9 se observa la convergencia del valor objetivo o error a minimizar, obtenido al evaluar cada una de las soluciones muestreadas por generación, considerando el menor valor encontrado (correspondiente a la mejor solución encontrada hasta el momento), la media y mediana de los valores objetivos encontrados y la desviación estándar de estos.

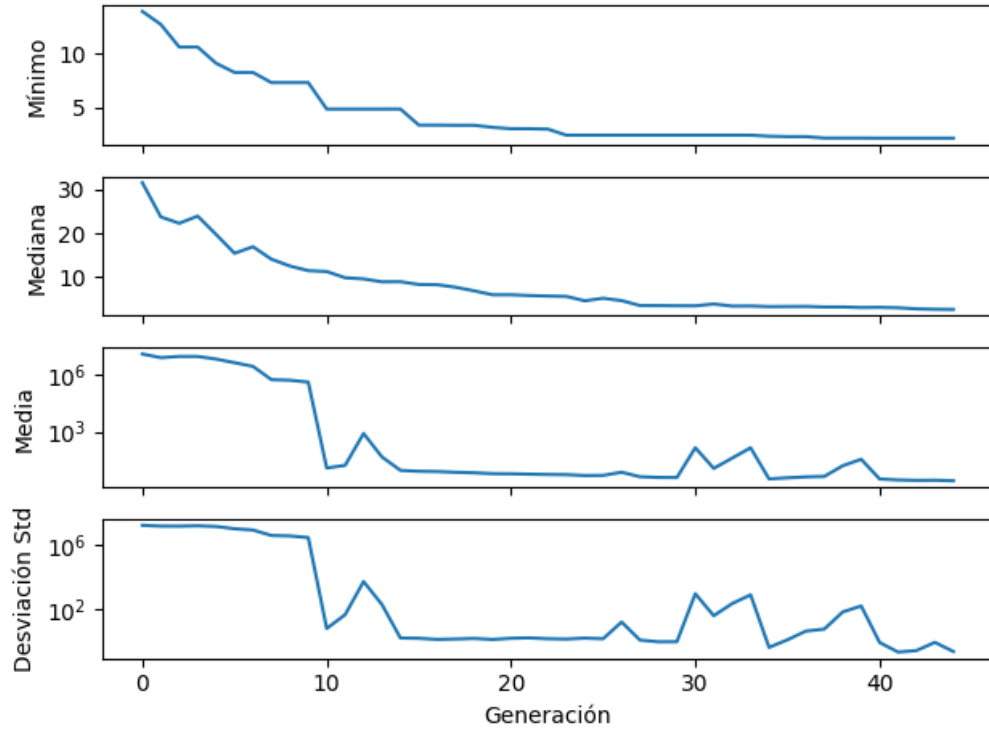


Figura 5.9: Análisis de la convergencia del valor objetivo, aplicando algoritmo CMA-ES sobre parámetros electrofisiológicos (Experimento 1).

Fuente: Elaboración propia (2021)

En estos gráficos se observa que tanto el mínimo como la mediana van disminuyendo de forma exponencial durante todo el proceso. La media y desviación estándar en cambio muestran magnitudes altas (cercanas a 10^7) desde un comienzo, lo cual es explicable debido a que estos valores son influenciados respecto a soluciones con valores extremos, especialmente al existir una alta penalización. Pasada la 10ma generación estos valores comienzan a disminuir, indicando una menor dispersión de los valores objetivos. Debido a que la cantidad de generaciones realizadas no cumple el mínimo esperado de $120 + 30m/\lambda$ para medir el estancamiento, el algoritmo fue detenido prematuramente por límites de tiempo. Tomando las 45 generaciones, se utiliza la 4.9 para obtener una mejora relativa en base a las medianas observadas, se obtiene un valor de 0.169, lo que indica que todavía continua mejorando la solución en aquel punto. No obstante, es difícil determinar una cercanía a un punto de estancamiento, ya que observando los casos de la función de Rosenbrock y Rastrigin, ambos muestran niveles de mejora relativa similares en las primeras 100 generaciones, pero en el caso de Rosenbrock se muestran los mismos niveles al final de la ejecución acercándose al óptimo. El tiempo de ejecución total para este experimento hasta completar la generación 45 fue de aproximadamente 15 días enteros utilizando 104 núcleos del clúster.

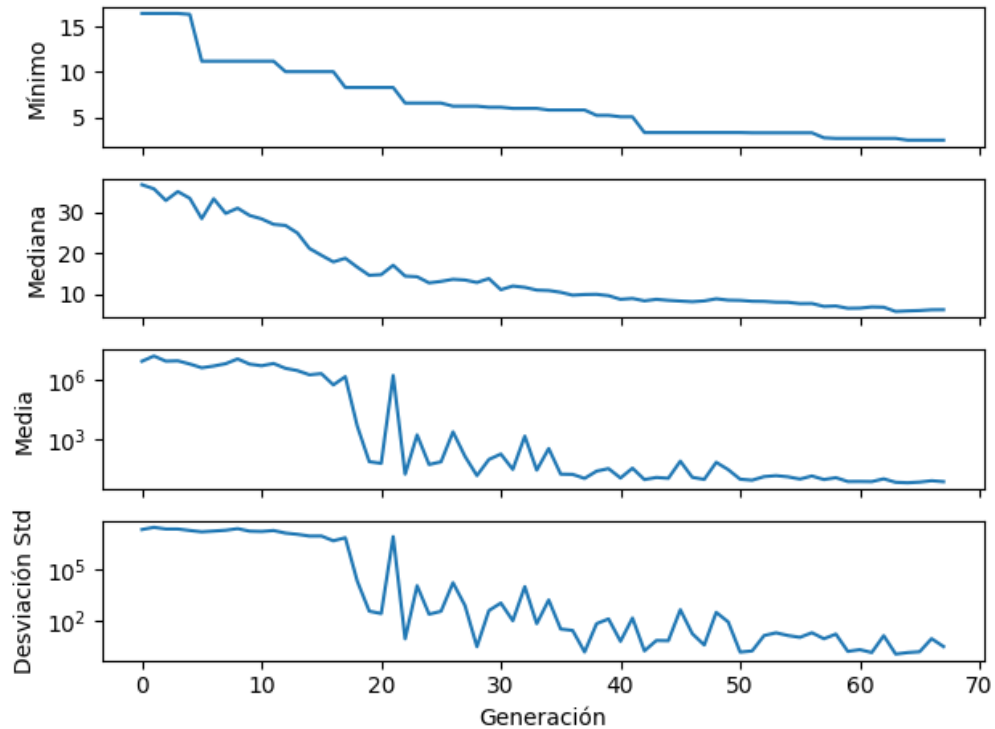


Figura 5.10: Análisis de la convergencia del valor objetivo, aplicando algoritmo CMA-ES sobre parámetros electrofisiológicos y morfológicos (Experimento 2).

Fuente: Elaboración propia (2021)

En la figura 5.10 se observa el resultado de la convergencia de valores objetivos para el experimento 2 con todos los parámetros, durante 68 generaciones. De forma similar al experimento anterior, el valor mínimo y mediano encontrado comienza disminuyendo gradualmente, mientras que la media y desviación estándar inicial comienzan de forma disparada por un orden de magnitud de 10^7 , mientras que alrededor de la generación 20 comienza a disminuir de forma errática. Pasada la generación 50, la media se estabiliza y la desviación estándar se mantiene en rangos similares, mientras el valor mínimo sigue disminuyendo en menor medida hasta pasar la generación 60 donde no se observan cambios significativos en este. La mejora relativa obtenida de la misma forma que el caso anterior para este experimento fue de 0.138, lo cual podría indicar que se encuentra una situación similar al experimento anterior, con valores que continúan disminuyendo pero no necesariamente indican una cercanía a un punto de estancamiento. El tiempo total de ejecución utilizado para llegar a completar la generación 68 fue de aproximadamente 23 días enteros, utilizando 104 núcleos del clúster, tomando cada generación aproximadamente 8 horas para evaluarse.

Por otro lado, dado que el algoritmo implementado utiliza una metodología de penalización para evitar soluciones infactibles fuera de las restricciones, analizamos las

soluciones muestreadas en cada generación. En la figura 5.11 se observa en el primer gráfico la cantidad de soluciones generadas por generación que no cumplen con al menos una restricción (considerable como una solución infactible), y en el segundo gráfico el promedio de parámetros por solución en violación de una restricción durante cada generación. Se observa que durante las primeras 15 generaciones aproximadamente, cada una de las 50 soluciones generadas mostraba al menos un parámetro fuera de una restricción, pero con un promedio de la cantidad de estos disminuyendo constantemente. Ya pasada la 15va generación comienzan a aparecer soluciones factibles, aumentando estas cada vez más y disminuyendo la cantidad de parámetros infactibles presentes en el resto. Al llegar a la generación final la cantidad de soluciones infactibles es cercana a 0.

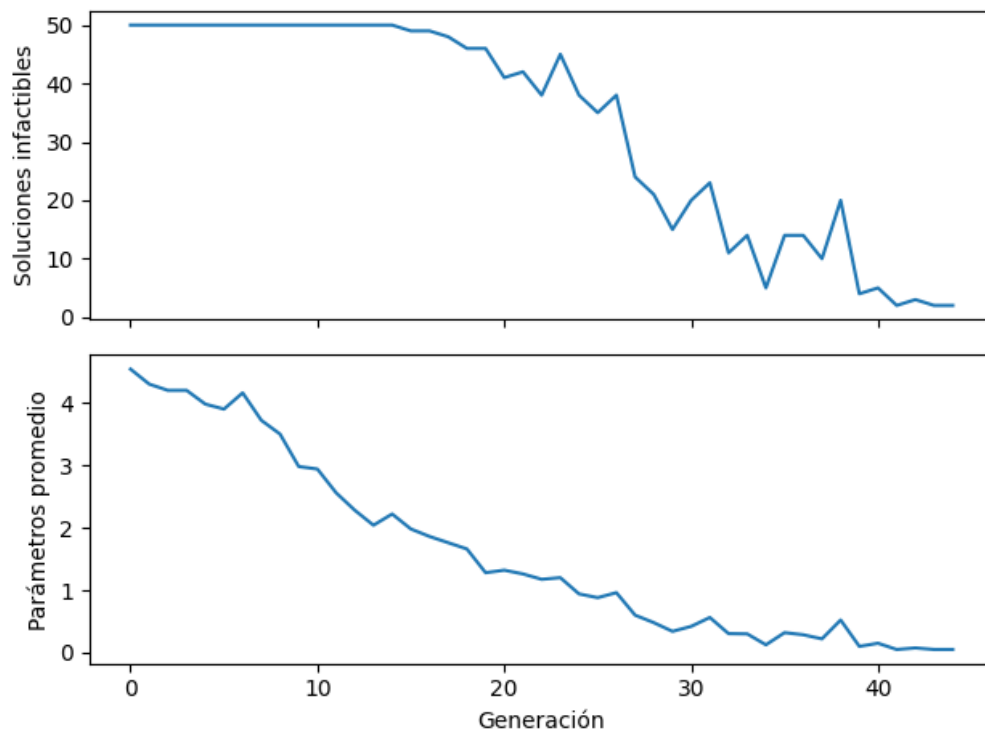


Figura 5.11: Análisis de cumplimiento de restricciones, por soluciones infactibles en cada generación y por parámetros promedio fuera de restricción por solución en cada generación, aplicando algoritmo CMA-ES sobre parámetros electrofisiológicos.

Fuente: Elaboración propia (2021)

Para el segundo experimento, en la figura 5.12 se muestra que toda las soluciones generadas durante el experimento tuvieron al menos un parámetro infactible, pero el promedio de parámetros en violación de una restricción fue disminuyendo de forma constante, estabilizándose cerca de 1 o 2 parámetros por solución. Hay que destacar y como se observa en la siguiente sección, la solución obtenida muestra solo 1 parámetro fuera de restricción y por un leve margen,

por lo que es posible, considerando las restricciones como rangos de valores esperados, que la solución todavía se encuentre dentro de lo factible para el modelo. Por otro lado, posiblemente hubiese sido necesario utilizar una estrategia de penalización un tanto más agresiva, con tal de alejarse en mayor medida de soluciones fuera de las restricciones.

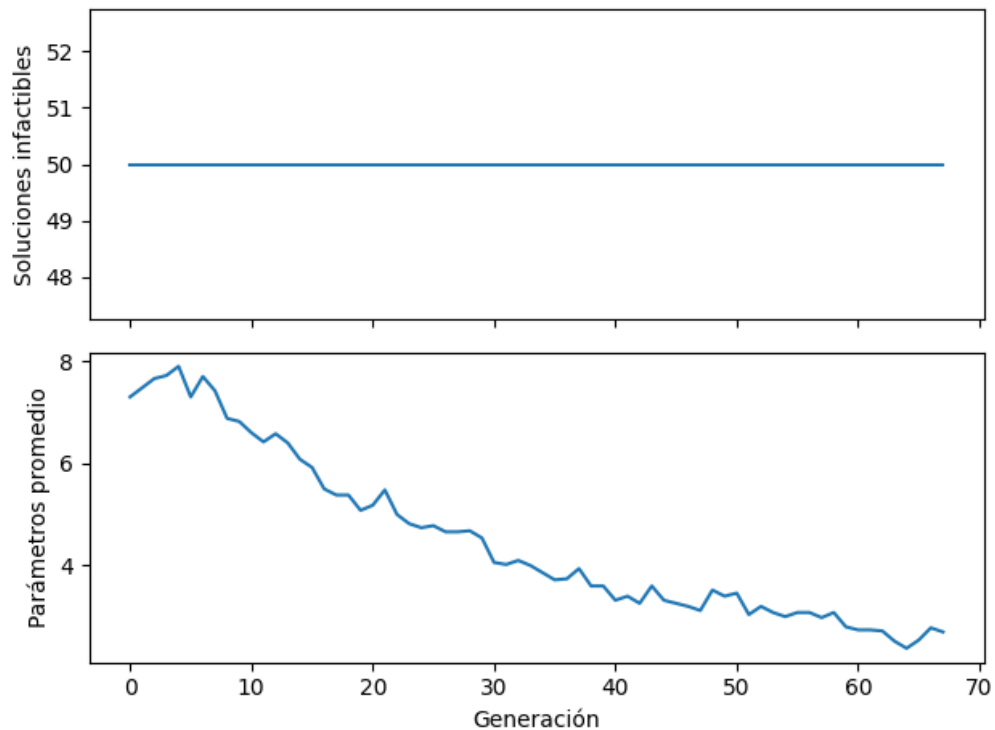


Figura 5.12: Análisis de cumplimiento de restricciones, por soluciones infactibles en cada generación y por parámetros promedio fuera de restricción por solución en cada generación, aplicando algoritmo CMA-ES sobre parámetros electrofisiológicos y morfológicos.

Fuente: Elaboración propia (2021)

5.3 ANÁLISIS DE SOLUCIONES OBTENIDAS

Para estudiar las soluciones obtenidas primero debemos analizar la solución original, correspondiente a los parámetros base del modelo. En la figura 5.13 se observan las curvas de densidad de impulsos para los parámetros base del modelo, y en la figura 5.14 la cantidad de impulsos promedio para rangos de un segundo, usando las mismas amplitudes. En comparación a las curvas obtenidas de forma experimental mostradas en la figura 1.3 correspondientes al experimento de Crosby et al. (2017), las curvas de densidad muestran una magnitud mucho menor a la esperada, además de no mostrar una curva de aumento y disminución en los primeros 3 segundos para el caso de la amplitud entre 0.2 y 0.4 mA, y 0.4 a 0.6 mA. Tampoco se muestra

un aumento gradual en la magnitud de la curva según va aumentando la amplitud. Además se muestra una diferencia notoria con las curvas obtenidas mediante simulaciones reportadas en la misma figura. Esto puede darse debido a adaptaciones y cambios realizados en el modelo MRG respecto a la implementación de este utilizada en las simulaciones originales, particularmente, el uso de parámetros morfológicos mediante funciones de extrapolación, con tal de poder utilizarse con una distribución continua de diámetros de fibras nerviosas, en vez de los valores fijos dados por McIntyre et al. (2002) que se muestra en la figura 3.4, donde el menor valor esperado es de $5.7 \mu\text{m}$, mientras que en este experimento se necesitan valores distribuidos entre 3.0 y $3.7 \mu\text{m}$ principalmente. Otro punto muy importante a mencionar y que puede afectar la generación de estas curvas tanto para los parámetros base como en los siguientes experimentos, es que para rangos de amplitud superiores, particularmente desde el 4to rango, NEURON empieza a arrojar advertencias, donde algunos valores que calcula en base a una función exponencial $f(x) = e^x$ no logra realizarlos si $x > 700$, por lo que reemplaza este valor por el correspondiente a e^{700} . Este fue un motivo adicional para utilizar solo los 3 primeros rangos en el proceso de optimización, pero es posible que afecte la visualización de los resultados para el resto de los rangos.

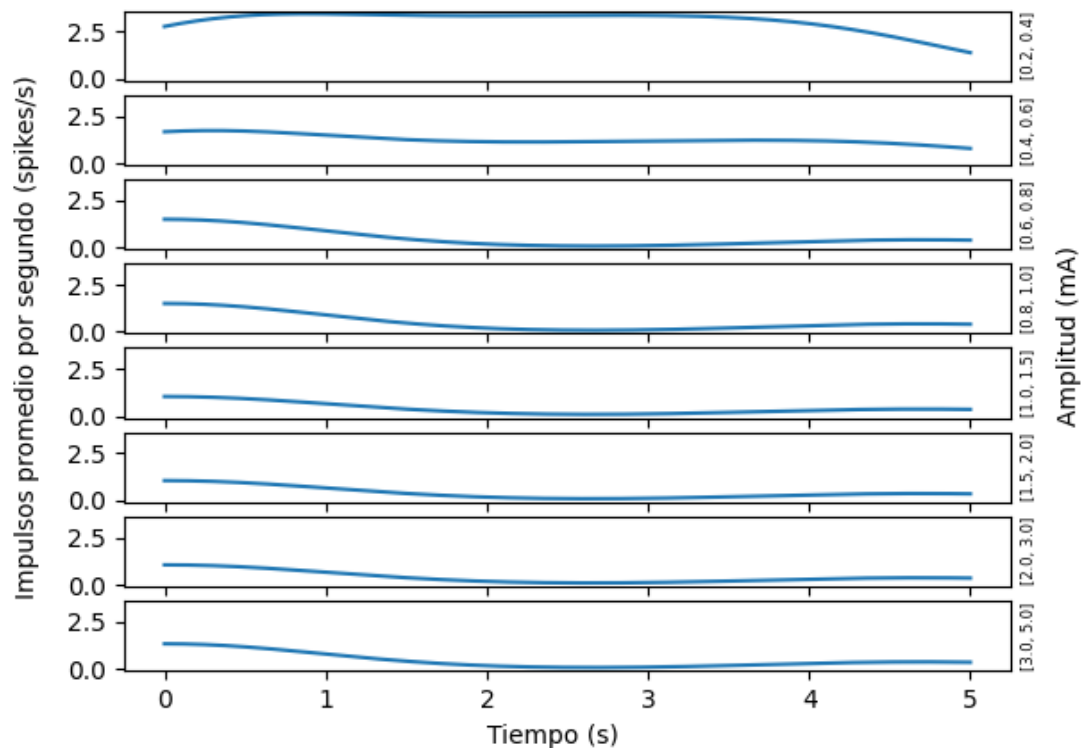


Figura 5.13: Curvas de densidad de impulsos promedio por fibra para distintas amplitudes con parámetros base en el modelo utilizado.

Fuente: Elaboración propia (2021)

En los gráficos de barra de la figura 5.14 se observa que para todas las amplitudes se muestra una concentración de impulsos en el primer segundo, y luego desapareciendo hasta los 4 segundos excepto para los dos primeros rangos, donde se continuaron registrando impulsos pero en menor densidad para el segundo caso. Durante el último segundo se registra para todos los casos al menos un impulso, el cual se explica debido a que el simulador se encuentra configurado para quitar el campo eléctrico correspondiente al SCS durante el 10% final del tiempo de simulación como se observa más adelante.

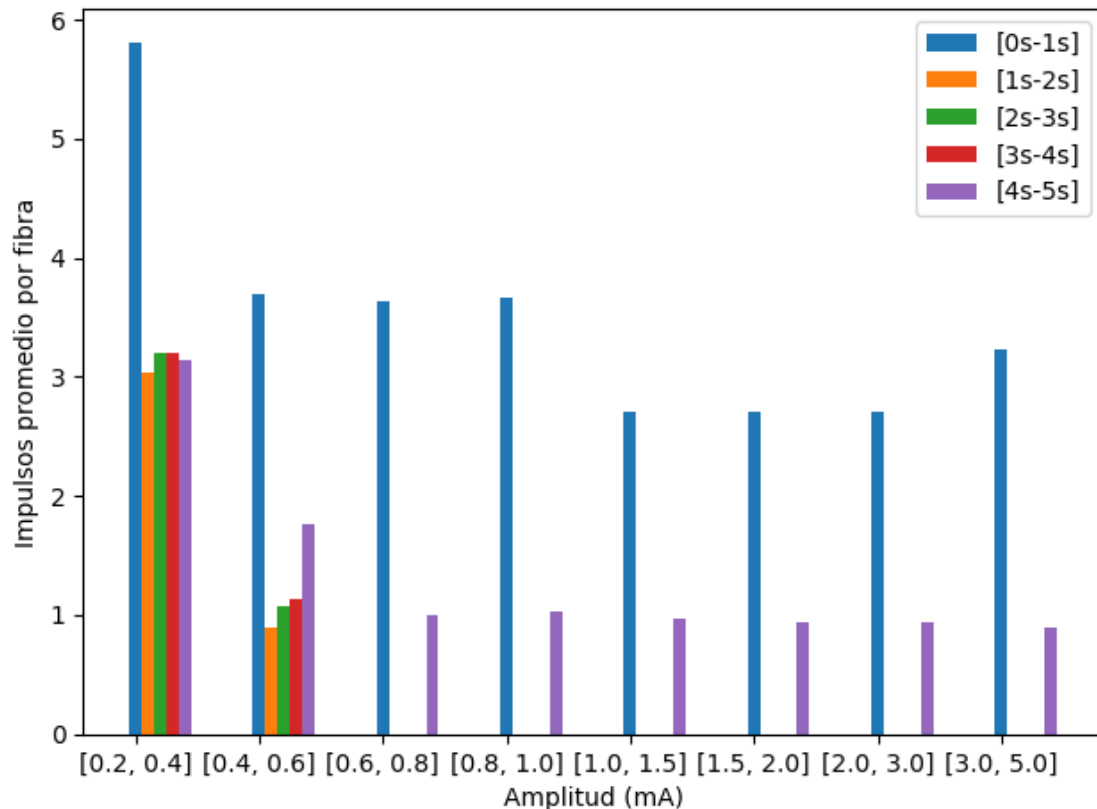


Figura 5.14: Cantidad de impulsos promedio por fibra dentro de cada segundo para distintas amplitudes usando parámetros base.

Fuente: Elaboración propia (2021)

Analizando en más detalle la respuesta eléctrica de la fibra, en la figura 5.15 observamos la respuesta registrada para 3 diámetros de fibra específicos, para amplitudes a la mitad de cada rango. En todos los casos se registra un conjunto de impulsos acumulados en el primer instante, pero en el primer caso se observa que posterior a eso, continúan un tren de impulsos constante, mientras que en el resto de los casos no se muestra ningún impulso hasta desactivar el campo eléctrico del SCS y se registra un impulso en el mismo instante para cada caso, exceptuando en fibras de mayor tamaño. Se puede deducir de esto que para todas las amplitudes salvo la primera se produjo un bloqueo de conducción pasado los impulsos iniciales,

y que una vez removido el campo eléctrico el bloqueo se detiene y la señal antidrómica vuelve a presentarse. El caso de la amplitud de 0.3 mA se vuelve la excepción a esto, donde la señal antidrómica sigue presentándose tanto antes como después de quitar el campo eléctrico.

El motivo porque para la amplitud de 0.5 mA se presenta bloqueo pero no en su rango correspondiente del gráfico de barra, es porque en este último se agrupan también casos desde 0.4 mA y 0.6 mA, y es posible que valores inferiores a 0.5 mA no presenten bloqueo de conducción y continúen mostrando impulsos, por lo que al comparar las barras del primer y segundo rango, este último presenta una menor cantidad de impulsos promedio pasado el primer segundo.

En el Anexo A, las figuras A.1 y A.2 muestran la respuesta para una única fibra, para 2.8 y 12 segundos. En el primer caso se observa que la respuesta antidrómica no alcanza a manifestarse luego de quitar el SCS por lo que no afecta el algoritmo CMA-ES que utiliza este tiempo, mientras que en la segunda figura se observa en mayor detalle como la señal antidrómica se sigue manifestando una vez quitado el campo eléctrico SCS.

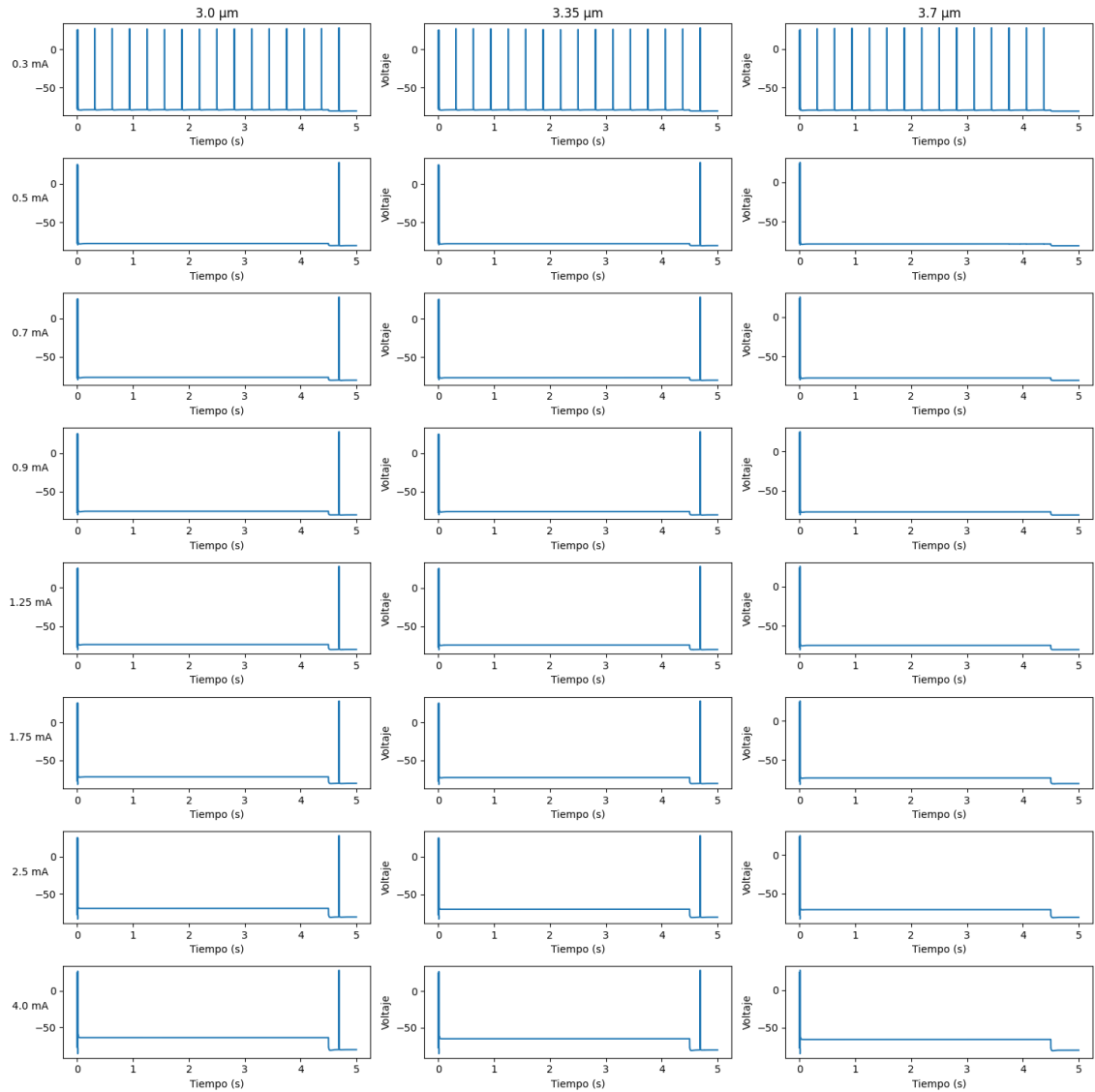


Figura 5.15: Respuesta eléctrica para fibras de distintos tamaños para cada rango de amplitud durante 5 segundos, usando parámetros base.

Fuente: Elaboración propia (2021)

Ya estudiados los parámetros base del modelo y el comportamiento que se tiene en base a estos, pasamos a analizar los resultados obtenidos mediante la optimización usando CMA-ES, primero para el caso de solo parámetros electrofisiológicos. En la tabla 5.1 se observan los valores obtenidos para los parámetros optimizados, comparados con los parámetros base del modelo.

Tabla 5.1: Valores obtenidos utilizando CMA-ES aplicados solo sobre parámetros electrofisiológicos comparados con los valores base del modelo.

Fuente: Elaboración propia (2021)

	Valor base	Valor obtenido	Diferencia
C_{dc}	2	2.0166	1%
g_{MYSA}	0.001	0,004387	339%
g_{na}	3	9.8987	230%
g_k	0.08	0,6118	665%
g_{nap}	0.01	0.09767	877%
g_l	0.007	0,005195	-26%

Los valores que presentaron el mayor cambio fueron los correspondientes a la conductancia máxima Na^+ o g_{na} , y la conductancia máxima K^+ o g_k , la conductancia máxima Na^+ persistente o g_{nap} y la conductancia MYSA o g_{MYSA} . Los otros parámetros se mantuvieron cercanos a los valores originales, y todos dentro de las restricciones.

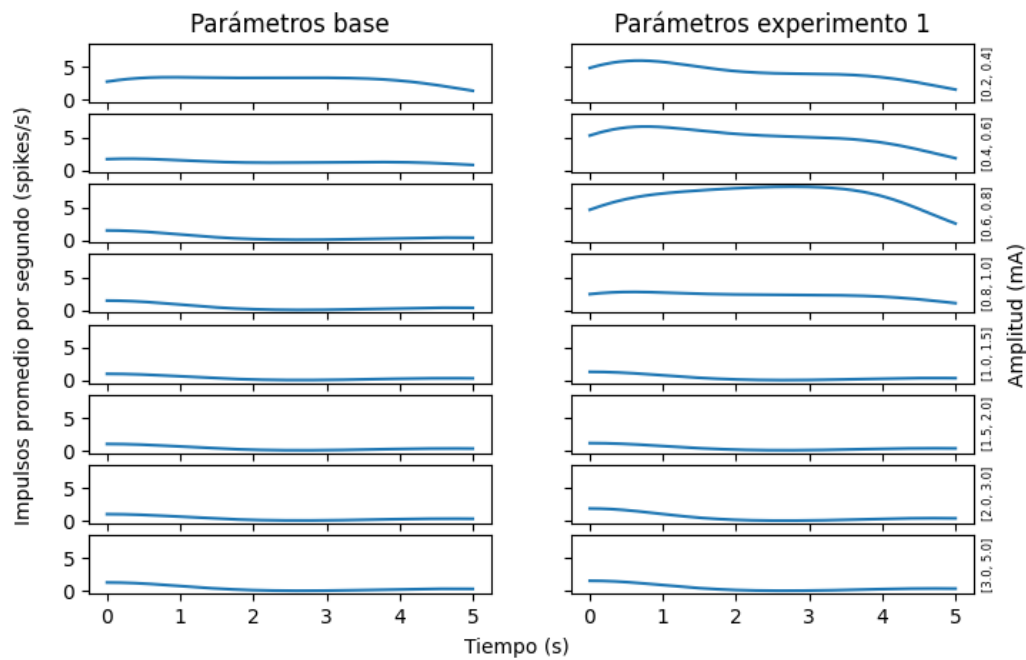


Figura 5.16: Curvas de densidad de impulsos promedio por fibra para distintas amplitudes usando parámetros obtenidos mediante CMA-ES solo variando parámetros electrofisiológicos, comparadas con curvas obtenidas utilizando los parámetros base.

Fuente: Elaboración propia (2021)

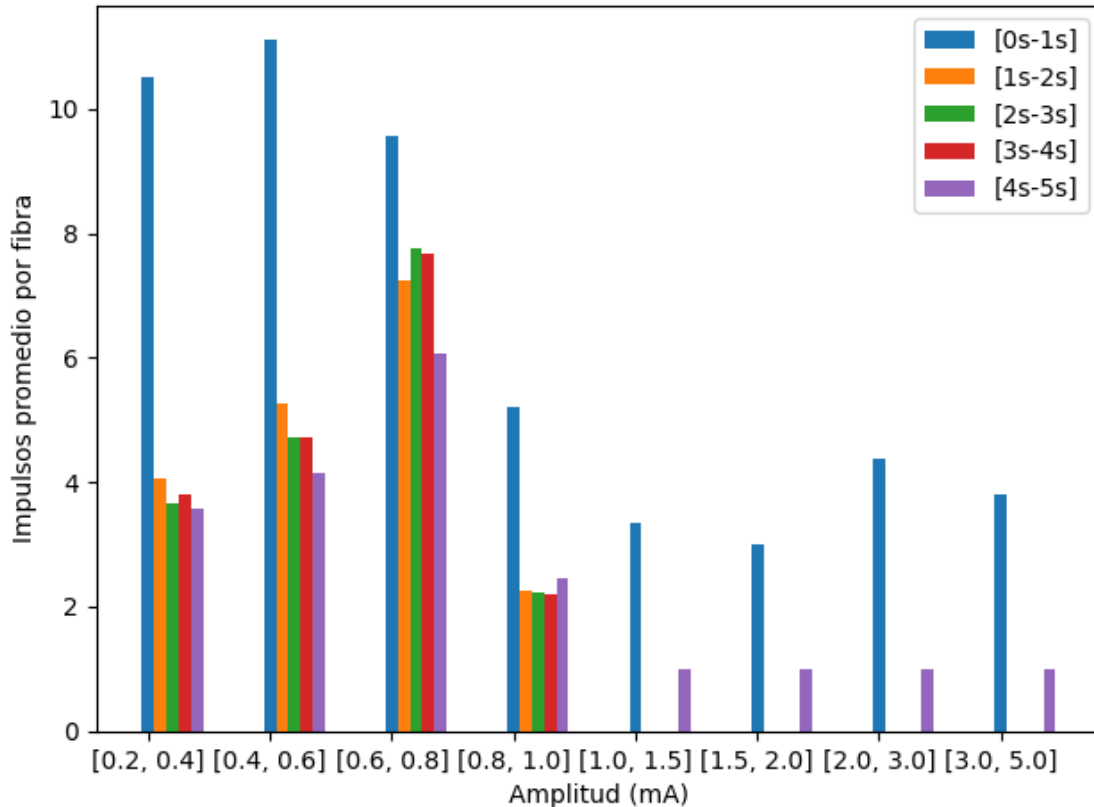


Figura 5.17: Cantidad de impulsos promedio por fibra dentro de cada segundo para distintas amplitudes usando parámetros obtenidos mediante CMA-ES, solo variando parámetros electrofisiológicos.

Fuente: Elaboración propia (2021)

En la figura 5.16 se observan las curvas de densidad de impulsos y en la figura 5.17 se observan los impulsos promedio por intervalo de tiempo. Se observa que la diferencia principal en comparación a los parámetros base es que para este caso, los tres primeros rangos y en menor medida el cuarto rango presentan magnitudes mayores en todo el intervalo de tiempo y de forma relativamente constante, sin presentar una forma de punta de crecimiento/decrecimiento como se esperaba salvo en los 2 primeros rangos donde hay una mayor concentración de impulsos al comienzo. Esto puede deberse a que las curvas de comparación utilizadas tenían un peak demasiado alto, con lo que de obtener una curva de densidad más alta, aunque no tenga la forma esperada esta se ajustaría más al objetivo. No obstante, sigue siendo deseable este aumento en la densidad de impulsos, especialmente de forma gradual según va aumentando la intensidad como se muestra en las figuras durante los 3 primeros intervalos, pero a partir del cuarto comienza a mostrarse un bloqueo de conducción general para los siguientes intervalos de intensidad. Es posible que esto también se deba al problema encontrado en NEURON al calcular exponenciales, y que de lograr solucionar esto o encontrar una implementación del modelo o

NEURON que permita calcular valores mayores los resultados observados sean diferentes para el 4to intervalo en adelante.

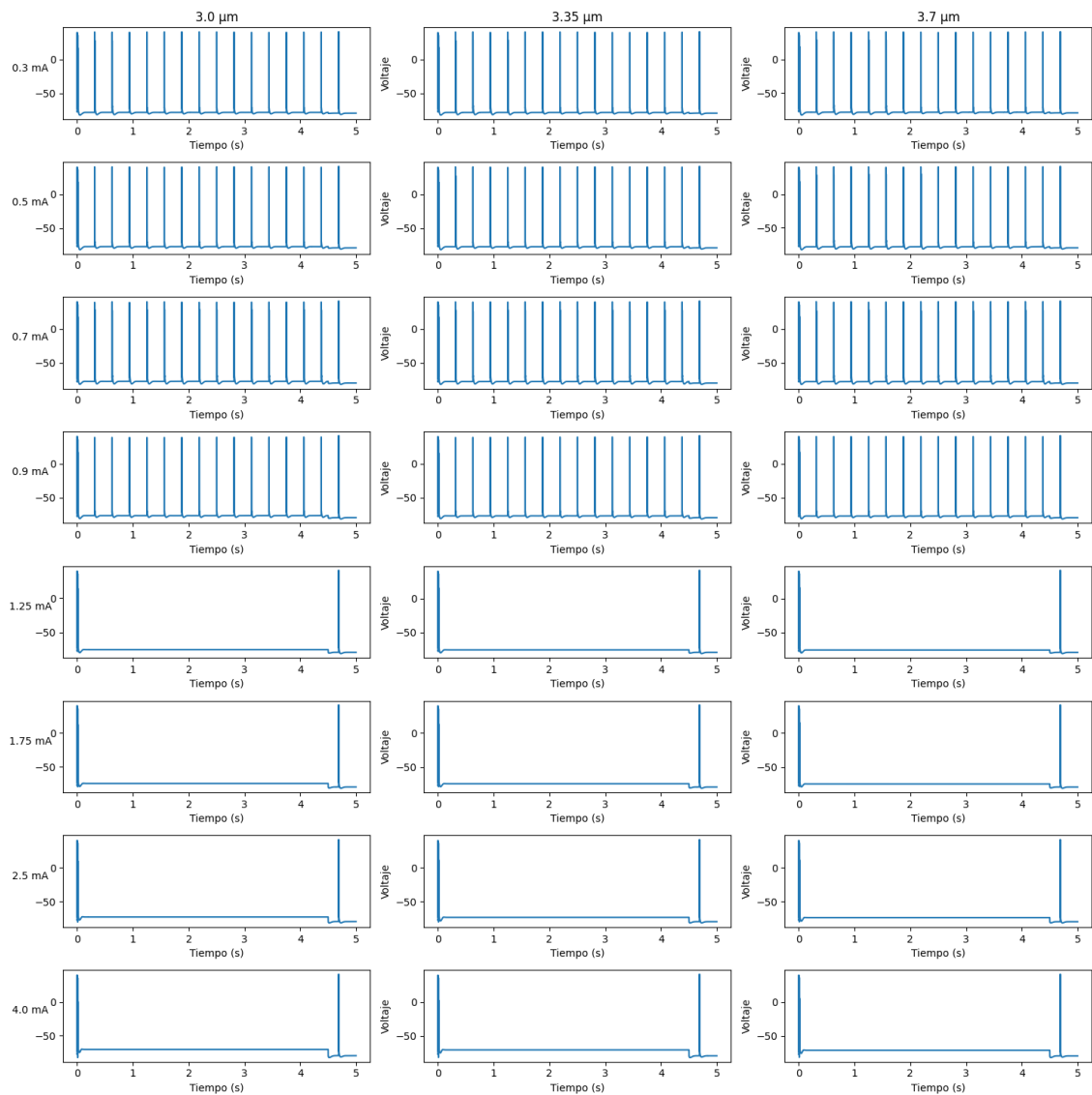


Figura 5.18: Respuesta eléctrica para fibras de distintos tamaños para cada rango de amplitud durante 5 segundos, usando parámetros obtenidos mediante CMA-ES, solo variando parámetros electrofisiológicos.

Fuente: Elaboración propia (2021)

En la figura 5.18 se observan que para estos cuatro intervalos se mantienen los impulsos y en los siguientes se observa un bloqueo de conducción. En el anexo A, las figuras A.3 y A.4 muestran el comportamiento eléctrico del modelo de fibra para estos parámetros con mayor detalle, durante 2.8 y 12 segundos. En el primero se observa el caso utilizado durante la optimización, mientras que en el segundo se aprecia de mayor manera como los impulsos se mantienen con y sin bloqueo de conducción. La figura A.3 permite observar en particular como se

concentran varios impulsos en pequeños intervalos de tiempo, lo cual explica en parte el porqué no se aprecia el aumento de la cantidad de impulsos en la figura 5.18.

En el anexo A, las figuras A.3 y A.4 muestran el comportamiento eléctrico del modelo de fibra para estos parámetros con mayor detalle, durante 2.8 y 12 segundos. En el primero se observa el caso utilizado durante la optimización, mientras que en el segundo se aprecia de mayor manera como los impulsos se mantienen con y sin bloqueo de conducción.

Pasando al caso de los resultados obtenidos mediante CMA-ES aplicado tanto a parámetros electrofisiológicos como morfológicos, en la tabla 5.2 se muestran los resultados obtenidos. Similar al caso anterior, g_{na} y g_k convergieron en gran medida y a valores cercanos para ambos casos, pero además se muestra un cambio mucho mayor en g_{nap} , mientras que el resto de los valores electrofisiológicos se mantuvieron cercanos a los originales. En el caso de los parámetros morfológicos, la mayoría de los parámetros convergió a valores similares a los originales, pero se observó una variación mayor en la segunda componente del diámetro nodal $NodeD^B$, y una reducción en ambas componentes del diámetro axonal $AxonD$, por lo que igualmente se consideran para análisis de sensibilidad más adelante. Hay que destacar que la variable g_{na} excedió el límite esperado original de 10 por aproximadamente un 5%, y al no ser una discrepancia mayor, puede ignorarse o proyectarse este valor al límite esperado.

Tabla 5.2: Valores obtenidos utilizando CMA-ES aplicados a parámetros electrofisiológicos y morfológicos del modelo

Fuente: Elaboración propia (2021)

	Valor base	Valor obtenido	Diferencia
C_{dc}	2	1.9296	-4%
g_{MYSA}	0.001	0.00053	-47%
g_{na}	3	10.4686	249%
g_k	0.08	0.7748	868%
g_{nap}	0.01	0.1238	1138%
g_l	0.007	0.009009	29%
δ_x^A	97.18	99.1754	2%
δ_x^B	54.046	50.1380	-7%
$NodeD^A$	1.1468	1.2222	7%
$NodeD^B$	0.1001	0.1482	48%
$AxonD^A$	5.9917	4.1236	-31%
$AxonD^B$	1.2499	0.5407	-57%
l_{FLUT}^A	23.865	24.6249	3%
l_{FLUT}^B	7.8005	8.7962	13%
N_{ml}^A	58.58	55.7587	-5%
N_{ml}^B	14.869	13.8877	-7%

En la figura 5.19 lo primero que se observa es la escala en la que se encuentran las curvas; Solo 3 curvas mantienen una densidad de impulsos elevada durante todo el transcurso del tiempo, pero a diferencia del caso base, los impulsos promedio por segundo se mantuvieron en una magnitud sobre los 5 impulsos promedio por segundo. Observando los parámetros obtenidos

en este experimento, las concentraciones mayores de sodio tienden a aumentar la tasa de disparo, lo cual explicaría el incremento en la magnitud de estas curvas, lo cual el algoritmo de optimización trató de lograr para acercarse en mayor medida a las curvas esperadas con una magnitud mucho más alta.

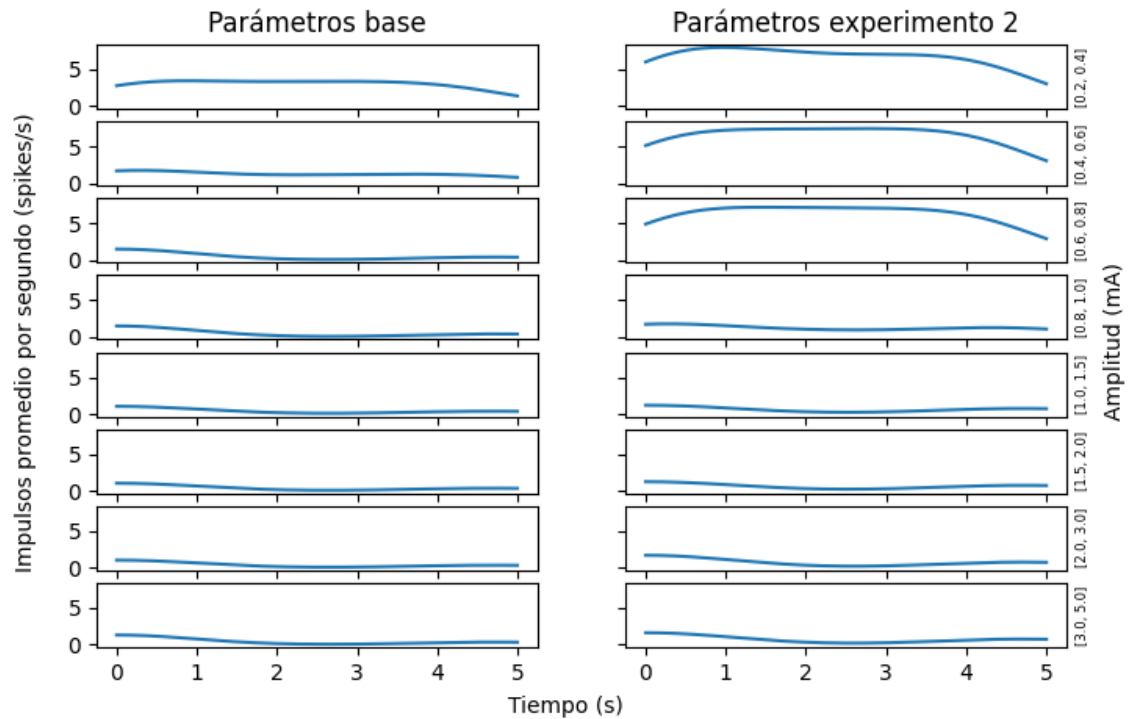


Figura 5.19: Curvas de densidad de impulsos promedio por fibra para distintas amplitudes usando parámetros obtenidos mediante CMA-ES solo variando parámetros electrofisiológicos y morfológicos, comparadas con curvas obtenidas utilizando los parámetros base.

Fuente: Elaboración propia (2021)

En la figura 5.20 se observa que para los cuatro primeros rangos se mostraron impulsos pasado el primer segundo, pero en el cuarto caso con una magnitud mucho menor, por lo que es posible solo unos pocos casos evaluados en este rango hayan presentado la ausencia de bloqueo de conducción y seguido mostrando impulsos, y dado la poca cantidad de casos, la curva de densidad se muestra similar a los casos con bloqueo de conducción.

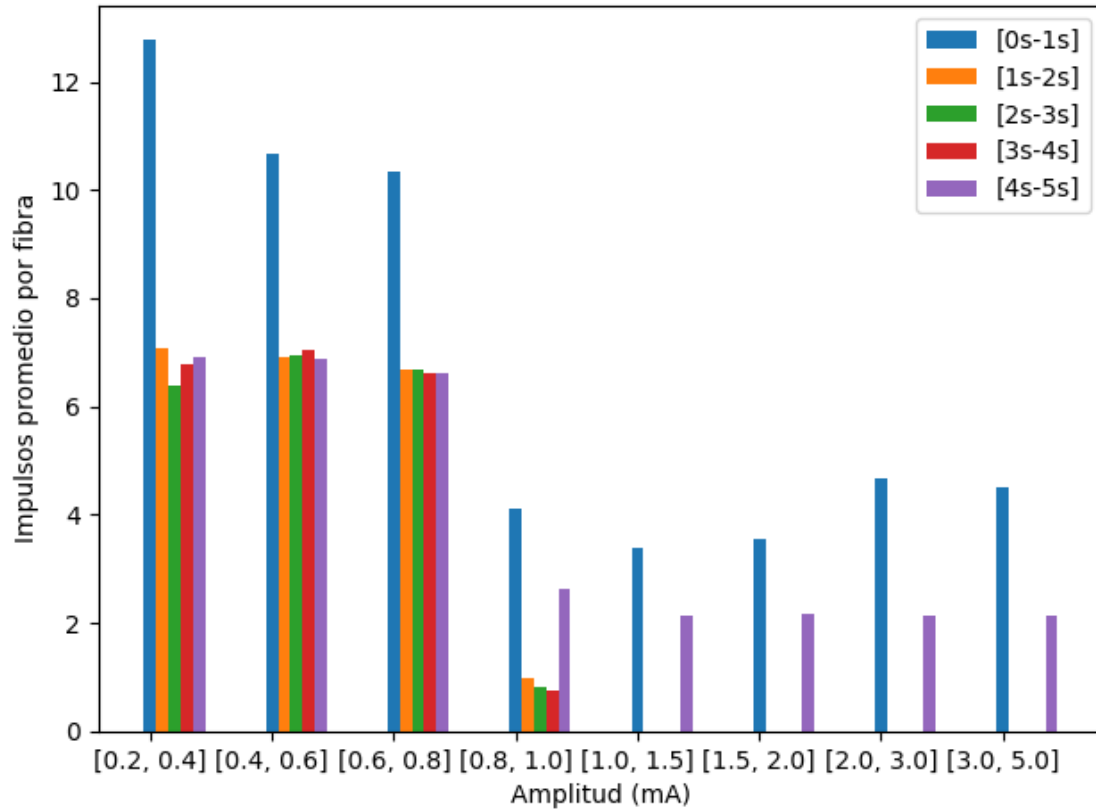


Figura 5.20: Cantidad de impulsos promedio por fibra dentro de cada segundo para distintas amplitudes usando parámetros obtenidos mediante CMA-ES, variando parámetros electrofisiológicos y morfológicos.

Fuente: Elaboración propia (2021)

En la figura 5.21 se observa nuevamente que para las 3 primeras amplitudes no se presenta bloqueo de conducción, mientras que para 0.9 mA si, por lo que de forma similar al caso anterior, para amplitudes menores a esta pero dentro del rango de los 0.8 a 1.0 m.A no se presente bloqueo. Más allá de eso, el resto de lo observado se mantiene similar al caso anterior. En el anexo A, las figuras A.5 y A.6 muestran el comportamiento durante 2.8 y 12 segundos, con el primero mostrando el caso usado con CMA-ES y el segundo permitiendo observar en mayor detalle el bloqueo de conducción.

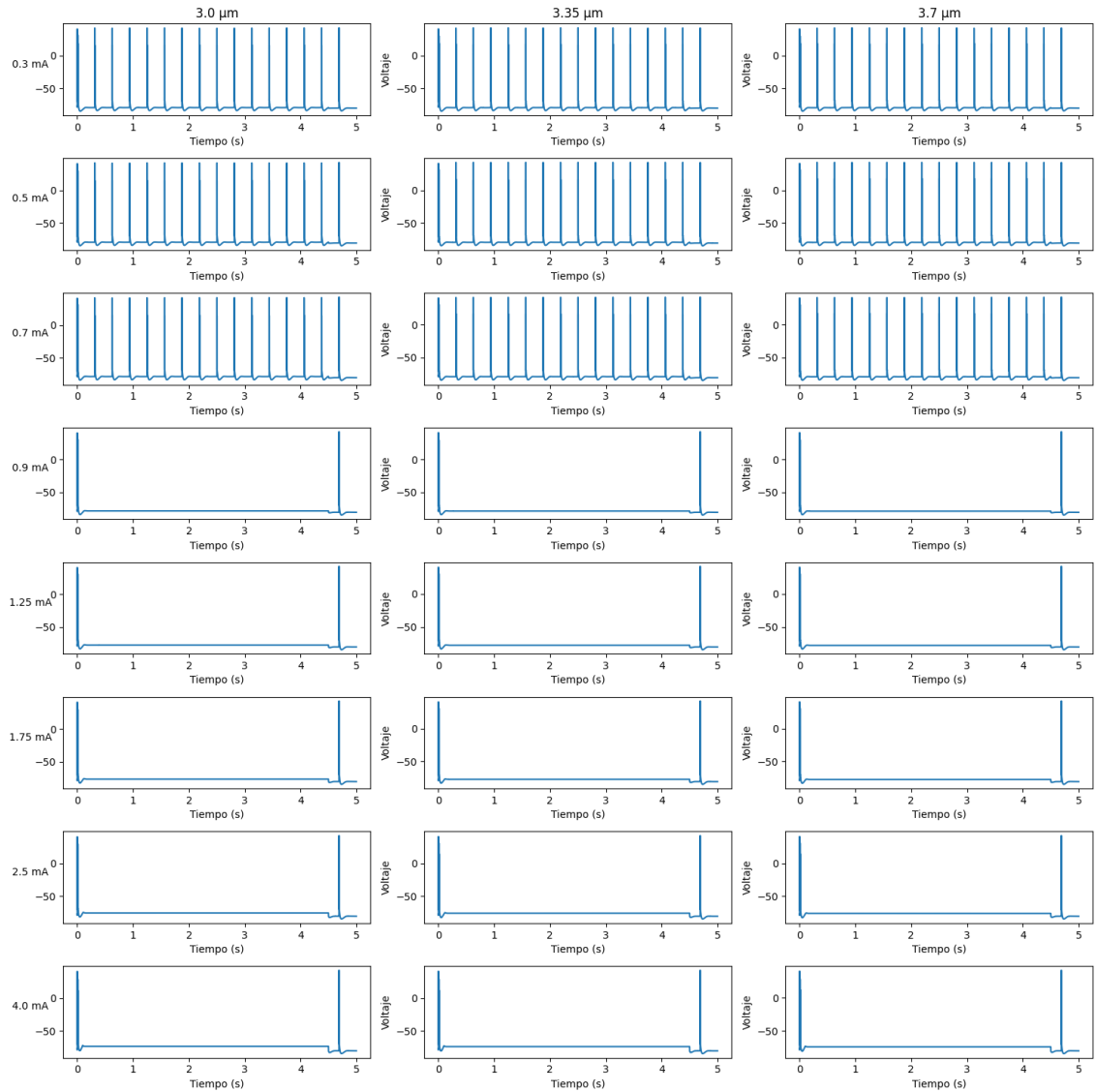


Figura 5.21: Respuesta eléctrica para fibras de distintos tamaños para cada rango de amplitud durante 5 segundos, usando parámetros obtenidos mediante CMA-ES, variando parámetros electrofisiológicos y morfológicos.

Fuente: Elaboración propia (2021)

5.4 ANÁLISIS DE SENSIBILIDAD

Ya observando los resultados obtenidos, se realiza un análisis de sensibilidad sobre las variables con cambios más significativos, particularmente utilizando los valores obtenidos en el último experimento correspondiente a la optimización sobre todos los parámetros, excepto para las pruebas de sensibilidad de g_{MYSA} , para el cual se utiliza tanto el valor obtenido en el primer experimento como el segundo. En todas las pruebas se utiliza el mismo diámetro de fibra de 3.35 micrómetros, y se realiza para los ocho rangos de amplitud.

5.4.1 Parámetros electrofisiológicos

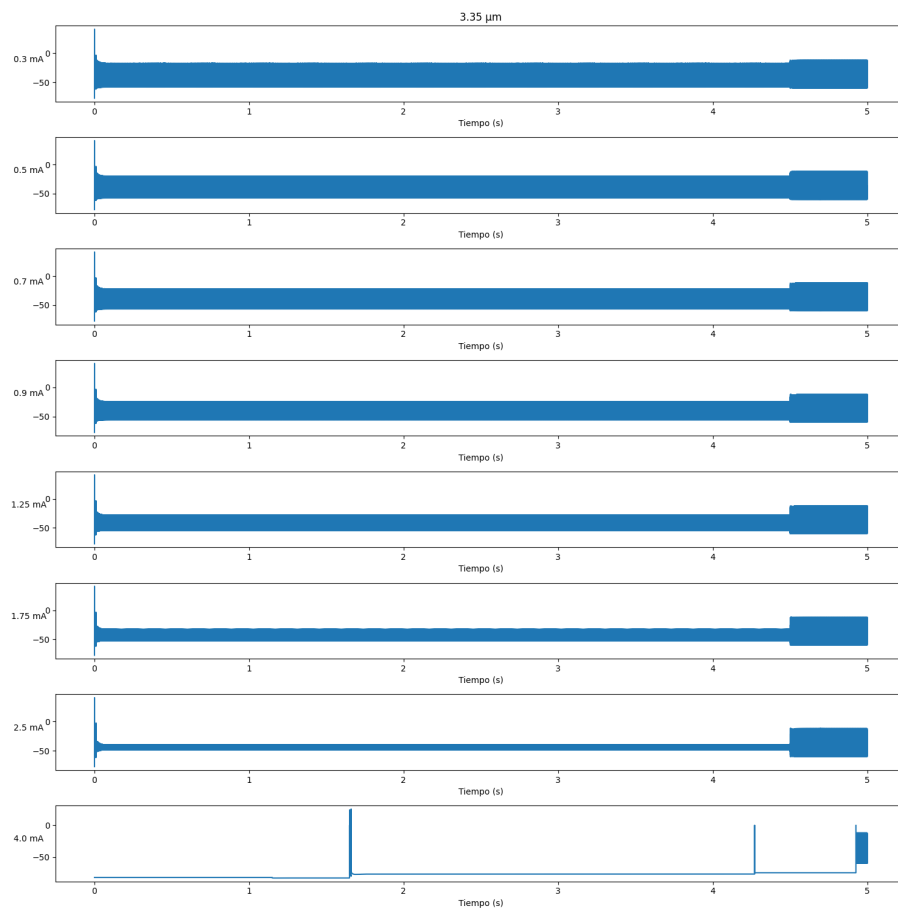


Figura 5.22: Respuesta eléctrica para una fibra de 3.35 micrómetros, modificando únicamente la conductancia máxima de sodio según el valor obtenido utilizando en el segundo experimento, manteniendo el resto de parámetros base.

Fuente: Elaboración propia (2021)

El primero en llamarnos la atención es la conductancia máxima de sodio g_{Na} , el cual tuvo un cambio del 249%. En la figura 5.22 se observa inmediatamente un cambio notorio, donde para todas las amplitudes excepto la última se muestra un patrón de disparo persistente, aumentando al final luego de quitar el campo eléctrico correspondiente al SCS. Se muestra además que la intensidad de estos impulsos va disminuyendo, en el último caso entregando un resultado diferente al resto de los observados posiblemente por limitaciones del simulador.

Dado que este valor está fuera de los rangos esperados, en el anexo B, la figura B.1 muestra el mismo análisis pero hecho para el valor obtenido para el primer experimento, el cual solo mostró un aumento del 206% y se encuentra dentro del rango esperado. En ese caso, el comportamiento es similar pero la intensidad de los impulsos registrados es mayor, incluyendo en el último rango donde no ocurrieron errores. El error observado usando el parámetro fuera del rango pudo ser debido a los impulsos generados cayendo a intensidades demasiado bajas para el simulador NEURON, dando el software valores erráticos durante la simulación para aquella amplitud, por lo que se recomendaría utilizar el parámetro con el valor dentro del rango establecido.

El segundo en llamarnos la atención es la conductancia máxima de potasio g_K , con un cambio de 868% en comparación al valor base. En la figura 5.23 se observa que para casi todos los casos solo se presentan los impulsos mostrados al comienzo y no se registran adicionales incluso sacado el campo eléctrico correspondiente al SCS. La única excepción es el caso de 4 mA, donde nuevamente se produjeron errores, mostrando un único impulso con una intensidad muchísimo mayor a lo visto en los otros casos o en la práctica.

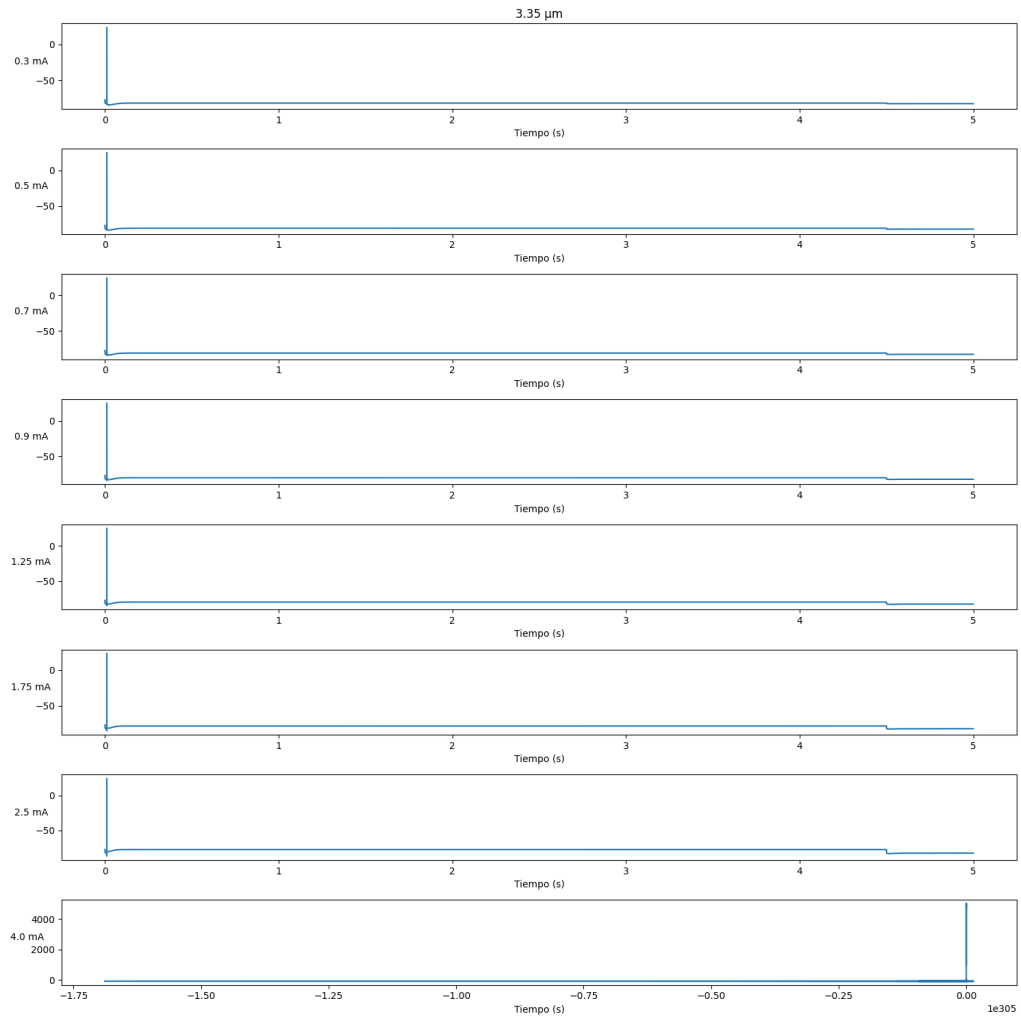


Figura 5.23: Respuesta eléctrica para una fibra de 3.35 micrómetros, modificando únicamente la conductancia máxima de potasio según el valor obtenido en el segundo experimento, manteniendo el resto de parámetros base.

Fuente: Elaboración propia (2021)

El tercero en llamarnos la atención, particularmente en los resultados vistos en la optimización sobre todos los parámetros es la conductancia máxima de sodio persistente g_{nap} , el cual muestra un aumento del 1138% sobre el valor original. En la figura 5.24 se muestra que para casi todos los rangos se muestra como un puro impulso, manteniendo un voltaje constante posteriormente salvo en el último caso, donde se puede apreciar un comportamiento errado, posiblemente otra vez por capacidad del simulador.

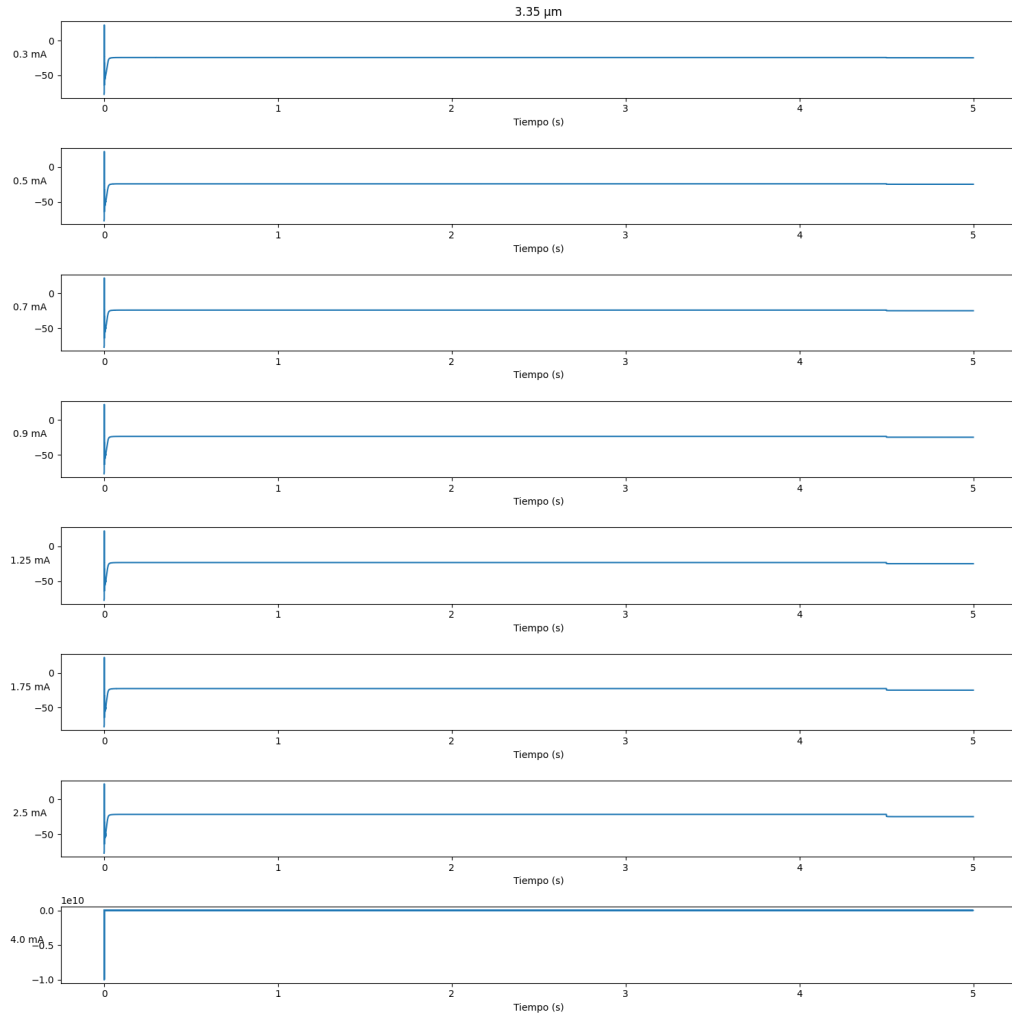


Figura 5.24: Respuesta eléctrica para una fibra de 3.35 micrómetros, modificando únicamente la conductancia máxima de sodio persistente según el valor obtenido en el segundo experimento, manteniendo el resto de parámetros base.

Fuente: Elaboración propia (2021)

En la figura 5.25 se observa la conductancia MYSA g_{MYSA} obtenida en el segundo experimento, la cual disminuyó un 47% respecto al valor base. Se observa que durante el primer segundo se generan varios impulsos concentrados, pero más allá de eso no presenta mayores diferencias respecto al caso base, salvo para una amplitud de 1.25 mA donde se generaron valores erróneos.

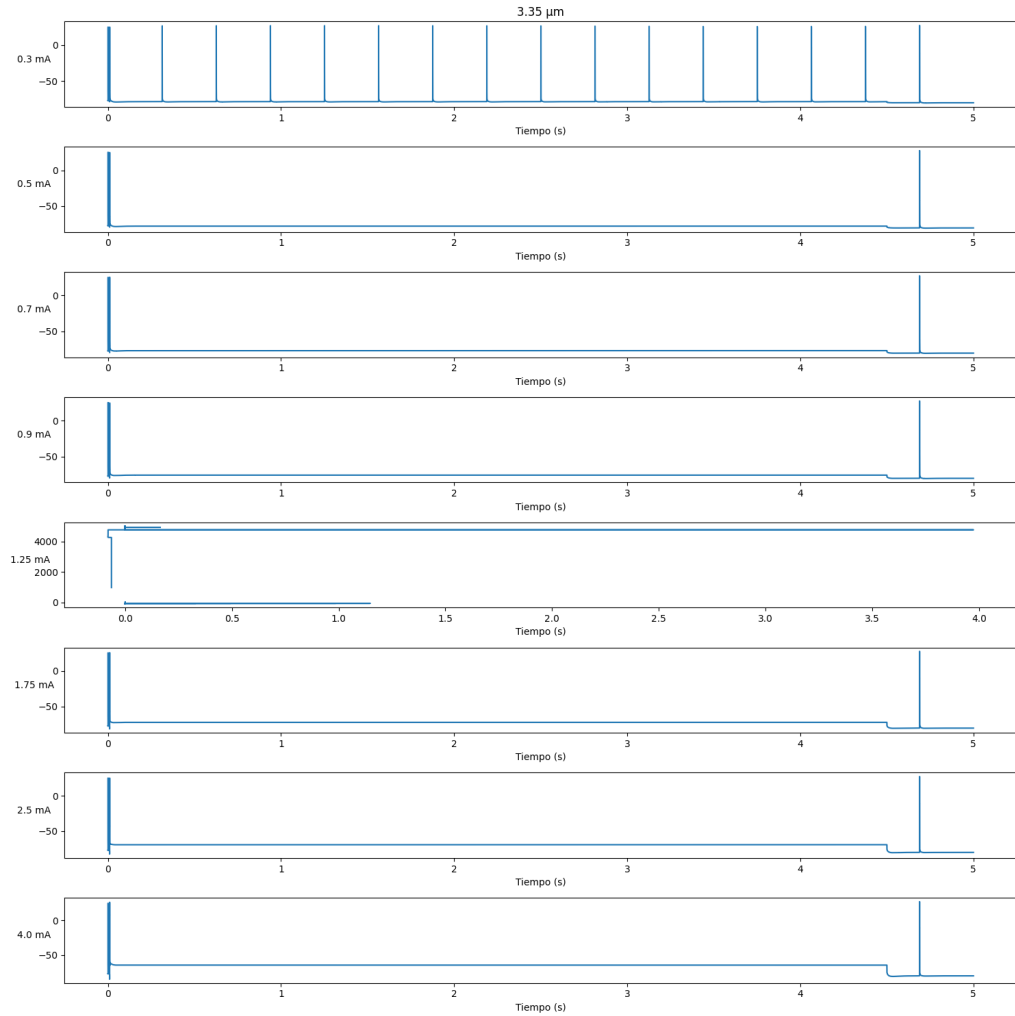


Figura 5.25: Respuesta eléctrica para una fibra de 3.35 micrómetros, modificando únicamente la conductancia MYSA según el valor obtenido utilizando CMA-ES en el segundo experimento, manteniendo el resto de parámetros base.

Fuente: Elaboración propia (2021)

Por ultimo, aparte de los resultados obtenidos en el segundo experimento, en la figura 5.26 se observa la conductancia MYSA g_{MYSA} para el primer experimento, la cual aumentó un 339% respecto al valor base. La mayor diferencia respecto al caso donde se disminuyó es el hecho que una vez quitado el campo eléctrico, no se observa ningún impulso al final.

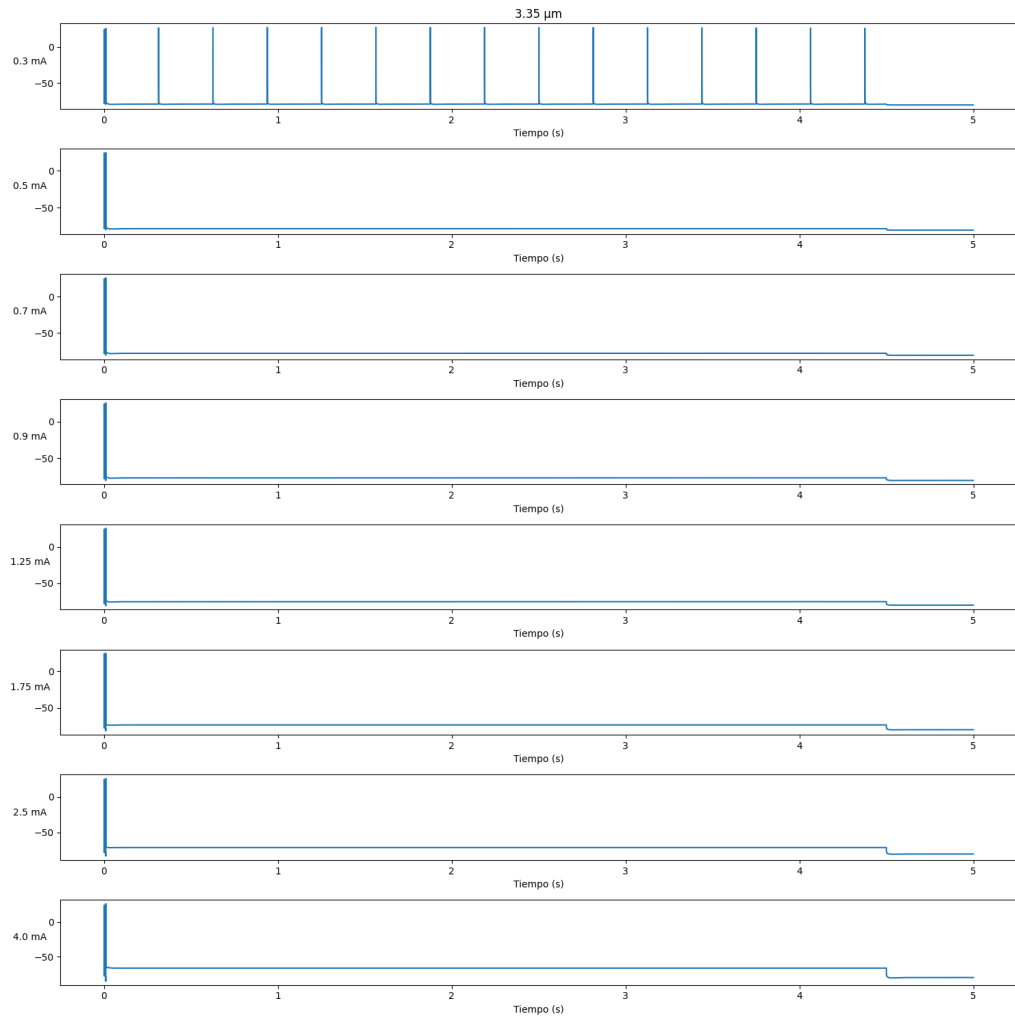


Figura 5.26: Respuesta eléctrica para una fibra de 3.35 micrómetros, modificando únicamente la conductancia MYSA según el valor obtenido utilizando CMA-ES en el primer experimento, manteniendo el resto de parámetros base.

Fuente: Elaboración propia (2021)

5.4.2 Parámetros morfológicos

Estudiando el caso de los parámetros morfológicos, el primero que consideramos es el correspondiente a la segunda componente para calcular el diámetro nodal, la cual cambió un 48% respecto al valor base. En la figura 5.27 se observa que para la primera amplitud se muestra un patrón normal de impulsos, pero en el segundo estos solo comienzan pasado los 2.5 segundos y ya para los siguientes no se muestran impulsos pasado el inicio, salvo para 0.9 mA donde se observan errores en los valores generados por el simulador.

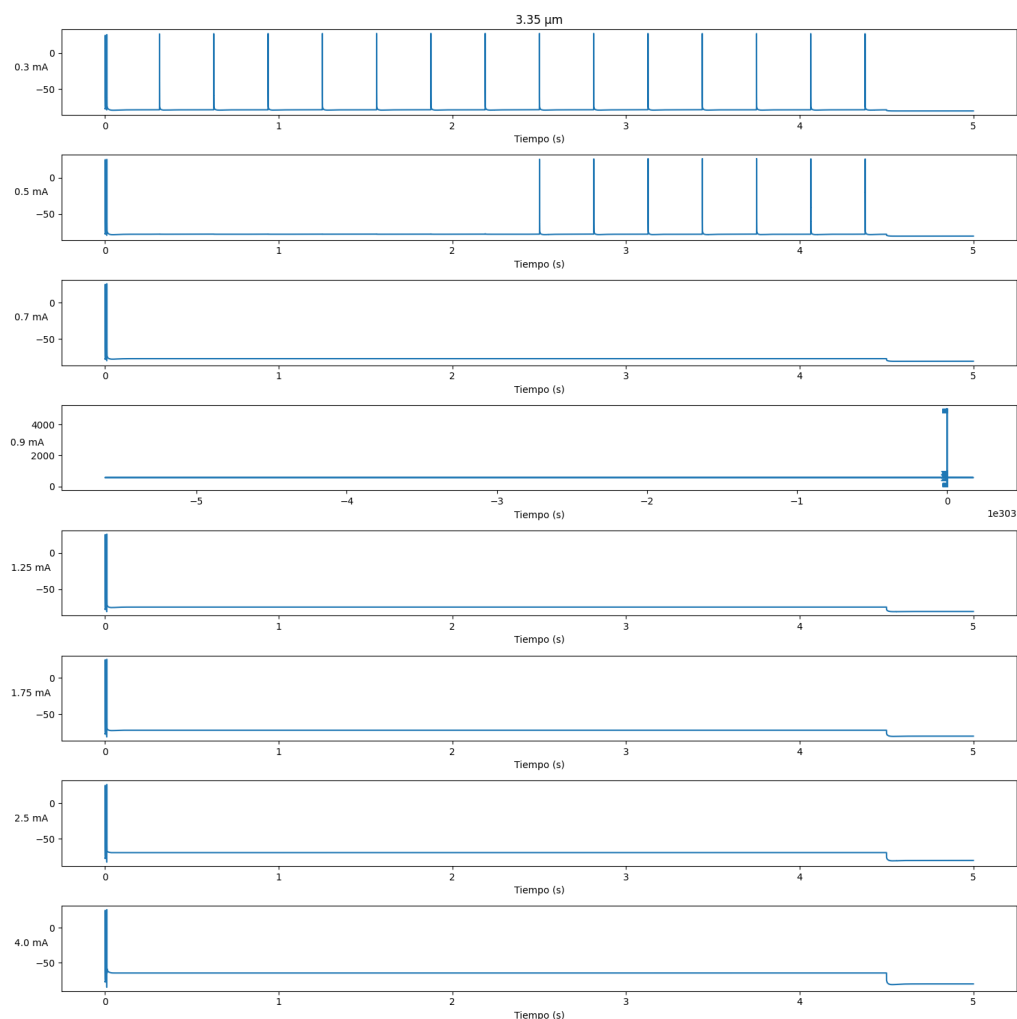


Figura 5.27: Respuesta eléctrica para una fibra de 3.35 micrómetros, modificando únicamente la segunda componente del diámetro nodal según el valor obtenido utilizando CMA-ES para todos los parámetros.

Fuente: Elaboración propia (2021)

Además de ese parámetro, dentro de los morfológicos consideramos los parámetros correspondientes a las dos componentes del diámetro axonal. En la figura 5.28 se observa la

respuesta solo modificando la primera componente, donde se aprecia un comportamiento muy similar al de los parámetros base, salvo para la amplitud de 4.0 mA donde se produce un error del simulador.

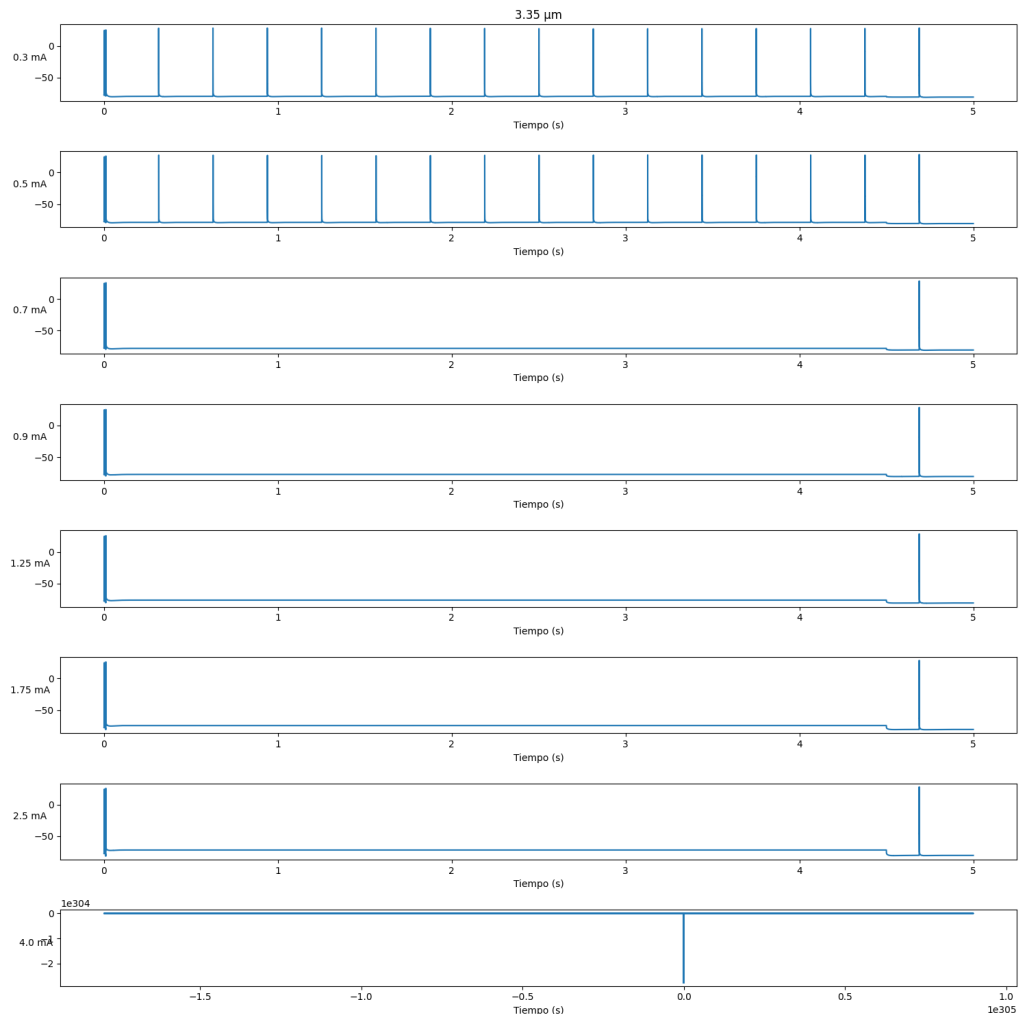


Figura 5.28: Respuesta eléctrica para una fibra de 3.35 micrómetros, modificando únicamente la primera componente del diámetro axonal según el valor obtenido utilizando CMA-ES para todos los parámetros.

Fuente: Elaboración propia (2021)

En la figura 5.29 se observa la respuesta al modificar la segunda componente donde el comportamiento nuevamente es similar pero esta vez mostrando bloqueo de conducción en todos los casos excepto para la primera amplitud.

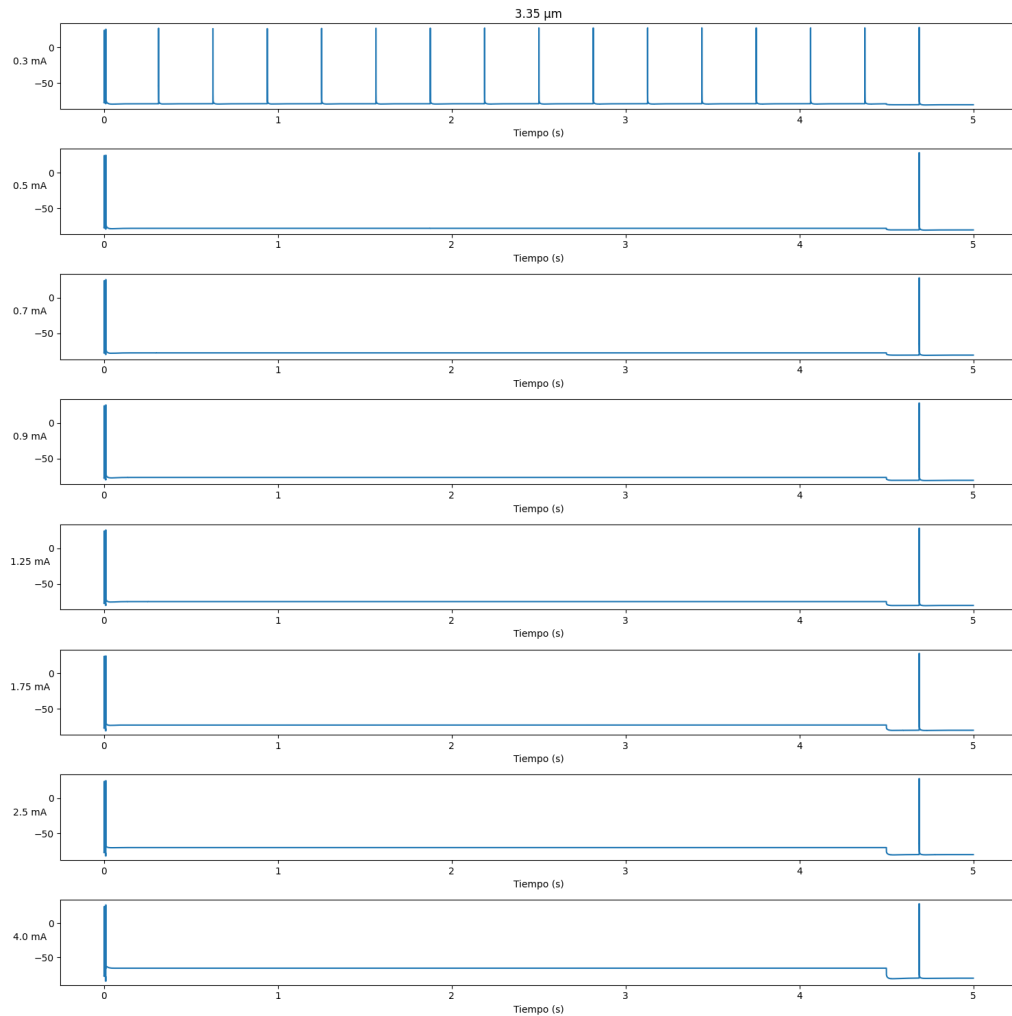


Figura 5.29: Respuesta eléctrica para una fibra de 3.35 micrómetros, modificando únicamente la segunda componente del diámetro axonal según el valor obtenido utilizando CMA-ES para todos los parámetros.

Fuente: Elaboración propia (2021)

A modo de complemento y dado que estas dos componentes modifican el mismo parámetro, en el anexo B, en la figura B.2 se observa la respuesta al modificar ambas componentes según los valores obtenidos, la cual no difiere mayormente de la respuesta dada al modificar únicamente la segunda componente, por lo que es posible esta predomine sobre la otra.

CAPÍTULO 6. CONCLUSIONES

A modo de conclusión, primero se presentan aquellas referentes al procedimiento experimental y las dificultades encontradas durante este, luego se abordan las conclusiones obtenidas respecto a los resultados y seguido de esto, las conclusiones respecto a cada objetivo específico, el objetivo general y la hipótesis planteada. Para cerrar, se habla del trabajo a realizar a futuro y observaciones finales.

6.1 DESARROLLO EXPERIMENTAL

En lo que respecta al desarrollo del experimento, la primera dificultad encontrada fue respecto al uso de NEURON mediante Python para simular el experimento original de Crosby. En este paso se utilizó código existente para realizar la simulación, pero se tuvieron que modificar distintas partes tanto del lado de Python como el modelo de NEURON escrito en HOC, particularmente para poder manipular directamente los parámetros del modelo desde Python y así poder probar distintas fibras nerviosas según parámetros generados durante la ejecución. Fue necesario un amplio estudio de NEURON y su comunicación con Python, y la mayor traba en este punto fue la documentación de NEURON; esta se encuentra escrita en un punto de vista del investigador del área de la neurociencia y no de la ingeniería de software, explicando todo desde la necesidad de la neuroingeniería para que un investigador de esta área lo utilice con una metodología particular, dejando de lado gran parte del contenido técnico del software. Se sugiere mucho utilizar ciertos métodos con llamados específicos para modelar el elemento neuronal que se desee, pero no se especifica mayormente en temas como la definición de las clases y métodos, entradas y retornos de funciones, entre otras cosas, lo cual dificulta trabajar con esta herramienta bajo un esquema que se sale del ejemplificado, como es en este caso integración con un algoritmo de optimización usando computación paralela, dejando gran parte del desarrollo con NEURON en base a ensayo y error para observar que hace cada método y clase.

Ya manejando NEURON y su integración con Python, realizar una simulación del experimento de Crosby no fue mayormente un problema. La primera dificultad en este punto fue la creación de las curvas de densidad de impulsos, dado que no se definía exactamente como en el trabajo de Crosby se generan y suavizan los histogramas, por lo que se decidió utilizar una técnica equivalente como es la estimación de densidad de Kernel.

Desde el lado del método de optimización, dado que se utilizó un framework especial para esto (DEAP), no hubo mayormente inconvenientes en la implementación base de CMA-ES integrándose con NEURON. La implementación de las restricciones resultó en una dificultad

encontrada ya que no se manejan mayormente por el framework y su integración con múltiples núcleos necesitó adaptaciones adicionales que se salen de lo sugerido por la documentación de DEAP. No obstante, el algoritmo se logró ejecutar sin problemas de forma local en un PC multinúcleo para tiempos de simulación cortos. La mayor dificultad surge de su implementación en el clúster de la Universidad de Santiago de Chile, donde fue requerida configuración adicional para lograr hacer funcionar la librería de multiprocesamiento en el clúster utilizando todos los nodos necesarios, así como también manejo de dependencias en cada nodo, y dado que existían errores al utilizar la versión de NEURON presente en el clúster, fue necesario utilizar un gestor de paquetes como Conda para poder tener la versión apropiada del software en cada computador. Todo esto se tuvo que manejar mientras se arreglaban múltiples errores que fueron surgiendo, lo cual finalmente atrasó la realización del experimento por aproximadamente tres semanas desde lo estimado.

Un punto importante respecto a NEURON y que afectó gran parte de la investigación son los tiempos de cómputo. Inicialmente se estimaba que para 5 segundos de simulación de fibra se necesitaría un tiempo de computo en un solo núcleo de 3 minutos, pero luego en la práctica esto fue mucho más incluso luego de reducir el coste computacional reduciendo el tiempo de simulación y otros parámetros, dando un tiempo de aproximadamente 12 minutos por fibra para solo 2,8s de simulación. Utilizando 104 núcleos del clúster, el tiempo de ejecución para una generación de 50 soluciones usando 3 rangos de amplitudes, cada una con 30 simulaciones de 2,8s es de aproximadamente 8 horas, o 3 generaciones por día. Dado esto y las demoras mencionadas anteriormente, el primer experimento realizado solo para parámetros electrofisiológicos se detuvo al momento que se observara que el resultado convergió a un punto, de esta forma dejando más tiempo para ejecutar el segundo experimento con todos los parámetros.

Por último, NEURON presentó problemas al momento de ejecutar el modelo MRG dado, particularmente al simular fibras usando impulsos con amplitud elevada, con lo que se arrojaban errores al momento de calcular el valor de una función exponencial con un exponente demasiado alto, truncando siempre este valor al de e^{700} lo cual afectó principalmente a las simulaciones con intervalos de intensidad altos, los cuales no fueron utilizados durante la optimización pero sí durante el análisis de resultados.

6.2 RESULTADOS

En cuanto a resultados, el proceso de optimización en ambos experimentos convergió a una solución que minimizaba la función objetivo, por lo que el algoritmo CMA-

ES como tal cumple con lo esperado de él. En ambos casos se converge a soluciones en gran parte similares salvo algunos parámetros. En el caso del experimento para solo parámetros electrofisiológicos el mayor cambio va en la conductancia de sodio, potasio, MYSA y en menor medida de sodio persistente, mientras que agregando los parámetros morfológicos, la conductancia de sodio persistente incrementa mucho más respecto al caso anterior. En el segundo experimento también los parámetros del diámetro axonal y uno del diámetro nodal divergieron de los parámetros base, mientras que el resto se mantuvo cercano a los iniciales.

Al momento de utilizar estos parámetros para simular las curvas objetivo, los resultados no se acercaron en gran medida a las curvas objetivo esperadas. El caso del primer experimento fue donde se tuvieron los mejores resultados, donde se observó un aumento gradual de la densidad de impulsos en cada rango hasta llegar al 4to, esto último posiblemente sesgado por el error del cálculo del exponencial mencionado anteriormente, u otros factores no vistos. En el primer y segundo rango particularmente se observó una concentración de impulsos mayormente al comienzo como se esperaría según las curvas objetivos, pero la forma de la curva sigue sin ser lo suficiente pronunciada al comienzo, y las magnitudes no llegan a los valores esperados en esos rangos. Los resultados obtenidos en el segundo experimento. muestran que para los tres primeros rangos nuevamente se aumentó de forma general la densidad de impulsos, pero no se muestra un crecimiento gradual de estos en cada rango, y solo una ligera concentración de impulsos al comienzo de la primera curva.

Es posible que estos resultados fuesen afectados, además del error de NEURON que pudo haberse presentado en ocasiones para intensidades menores, principalmente por la magnitud de las curvas esperadas pero también el intervalo de tiempo en el que se realiza la comparación, el cual fue de 2,8 s por motivos de tiempos de computo. Esto último pudo resultar en que no se considerara en mayor medida la forma de montículo de la curva, y se priorizara por ejemplo, una recta de mayor magnitud que aunque no tuviera la forma, se ajusta de mejor medida a la parte superior de la curva esperada que es lo que se está usando principalmente para comparar. Se propone utilizar un intervalo mayor de tiempo de comparación para considerar en mejor medida el aumento y disminución en la densidad de impulsos, pero sin ser demasiado grande, ya que esto podría resultar en que el algoritmo priorice por ejemplo, curvas cercanas a 0 que se ajustan de mejor medida a la sección inferior de la curva esperada

Otro problema que surge en menor medida y a considerar son la implementación de las restricciones. En el caso del primer experimento aproximadamente a partir de la 15va generación empezaron a encontrarse soluciones factibles, y llegando a la solución final la gran mayoría de las soluciones encontradas estaban dentro de lo esperado. Pero en el segundo experimento con todos los parámetros, todas las soluciones incluyendo la mejor encontrada presentan parámetros fuera de las restricciones, aunque esta excedió el límite por un solo

parámetro, fuera de la restricción por un 5% del valor de esta, y considerando que estos límites son valores esperados más que estrictos, esta discrepancia puede ignorarse o proyectarse sin problemas a la restricción correspondiente. Para el resto de las soluciones el promedio de parámetros fuera de los límites se mantuvo bajo al final, cercano a 2 parámetros promedio. De cualquier forma, es necesario considerar otras metodologías para implementar las restricciones, dado que la técnica de penalización y reparación pudo resultar en que muchas soluciones siempre cayeran en los bordes del espacio solución, y en su lugar se podrían utilizar técnicas de remuestreo, asignando un valor aleatorio dentro del espacio solución a los parámetros que se encontraran fuera de este, entre otras estrategias.

Respecto a las variables que cambiaron y el análisis de sensibilidad realizado, el hallazgo más notorio que se pudo observar es el incremento en la conductancia máxima de sodio, lo que resulta en un patrón de disparo persistente a alta frecuencia, lo cual no se observa para los parámetros base y las soluciones encontradas. En segundo lugar, un aumento considerable (868% respecto al valor base) de la conductancia máxima de potasio resulta en un bloqueo de conducción persistente, donde no se muestra conducción de ningún tipo de impulso, tanto antes como después de quitar el campo eléctrico correspondiente al SCS. Un aumento de la conductancia máxima de sodio persisten por sobre un 1000% resultó en un comportamiento similar al anterior, pero manteniendo un potencial eléctrico más alto. La modificación individual de otros parámetros, particularmente aquellos correspondientes a elementos morfológicos no presentaron cambios significativos, o los resultados que mostraban eran erráticos.

6.2.1 Objetivos e hipótesis

Sobre los objetivos que se buscaban completar, partiendo por los específicos descritos en la subsección 1.4.2 se considera lo siguiente:

- Se logró realizar la simulación del experimento de Crosby utilizando Python + NEURON para distintos parámetros asignados al modelo.
- Se logró implementar un algoritmo CMA-ES integrado con el software de simulación NEURON y Python, utilizando las simulaciones realizadas para comparar los resultados de esta con los esperados, obteniendo una métrica de ajuste utilizada como función objetivo a minimizar por el algoritmo de optimización.
- Se logró con dificultades la implementación del software de simulación y de optimización de forma distribuida utilizando el clúster de la Universidad de Santiago de Chile.

- Se pudo comparar los parámetros obtenidos mediante la ejecución del algoritmo de optimización con aquellos descritos por McIntyre et al. (2002). Si bien, los parámetros obtenidos minimizaban en mayor medida la función objetivo, solo los resultados obtenidos en el primer experimento lograban caracterizar en mejor medida los los resultados vistos en Crosby et al. (2017) en comparación a los parámetros base pero sin alcanzarlos particularmente en cuanto a las magnitudes esperadas. Los resultados obtenidos en el segundo experimento tampoco logran caracterizar en mejor medida los resultados esperados, ni representar una mejora significativa respecto a los parámetros base o los resultados vistos en el primer experimento.

Dado esto, el objetivo general mencionado en la sección 1.4.1 no se logró del todo, ya que si bien se logró realizar el experimento, al compararse los resultados obtenidos utilizando los parámetros encontrados con aquellos dados por los parámetros base solo se obtiene una mejora menor, especialmente para el primer experimento. Se considera que es necesario realizar repeticiones de los experimentos con distintos ajustes y mayores tiempos de ejecución antes de poder determinar si se acepta o rechaza la hipótesis descrita en la sección 1.3.1, debido a que es posible encontrar mejores resultados para una mayor cantidad de iteraciones, pero también, debido a la naturaleza estocástica del algoritmo, es necesario repetir la ejecución de este para determinar en mejor medida la consistencia en encontrar soluciones para este problema, y descartar que los resultados encontrados en este experimento sean netamente resultados de una mala ejecución en términos de aleatoriedad.

6.3 TRABAJO A FUTURO

Existen diversos puntos en los cuales se puede continuar con este trabajo, en base a la misma hipótesis planteada. Uno de estos es realizar ajustes a la función objetivo, ya sea aumentando el tiempo utilizado para la comparación de curvas, o encontrando formas de caracterizar en mejor manera la forma esperada de la curva dentro de la función. En este trabajo, se utilizó la función objetiva planteada bajo la suposición de la existencia de un conjunto de parámetros que lograra recrear con buen grado de exactitud los resultados vistos en el experimento de Crosby, incluyendo los valores observados por ejemplo, en las puntas de las curvas de densidad, y dado esto el algoritmo lograra encontrar un conjunto de parámetros que se acercasen a estos. Existe la posibilidad de que dada las limitaciones de NEURON, en conjunto a factores experimentales presentes en el experimento de Crosby que no se puedan recrear con la simulación, no sea factible encontrar un conjunto de parámetros que para simulaciones

del modelo recree exactamente las curvas experimentales o incluso se acerque a estos, pero puede ser posible que existan parámetros que mediante simulación generen curvas con el mismo comportamiento, pero magnitudes inferiores como las vistas en los resultados obtenidos. En otras palabras y resumiendo este punto, un objetivo a futuro sería obtener una función objetivo alternativa a la utilizada en este experimento, enfocada mayormente en el comportamiento visto en estas curvas más que los valores específicos mostrados en el experimento original, buscando al menos encontrar un conjunto de parámetros que recree el comportamiento de las curvas de Crosby aunque sea para una magnitud inferior. Existen también otros puntos a intentar recrear e investigar utilizando el modelo, como puede ser bajo que condiciones o parámetros se generan patrones de disparo persistentes u otros fenómenos que se deseen estudiar, para lo cual se puede realizar un procedimiento de optimización intentando buscar parámetros que consigan esto.

Otro punto importante y que es de gran importancia para lograr realizar avances en este trabajo, es lograr disminuir considerablemente el tiempo de computo utilizado por las simulaciones de NEURON. Observando los datos del potencial eléctrico generado durante la simulación de SCS aplicado a cada fibra, estos para promedian alrededor de los 100 Mb para 5 segundos de simulación en una sola fibra. Considerando estos datos son únicamente valores de dos ejes (tiempo y voltaje), es posible que dada la configuración de NEURON se estén realizando las simulaciones con demasiada precisión, generando una masa demasiado alta de datos. Una opción a intentar podría ser el disminuir la precisión utilizada en la simulación con tal de calcular una menor cantidad de puntos en el tiempo y por ende disminuyendo el coste computacional. No obstante, esto último podría traer errores en la simulación o sesgarla, alejándose de lo esperado experimentalmente. De realizar esto es importante encontrar un balance entre una reducción del coste computacional sin una pérdida significativa en la calidad de la simulación. Pueden existir otros puntos donde se pueda reducir el tiempo de computo, pero independiente de cuales se elijan, la reducción del tiempo de computo permitirá realizar los experimentos a una mayor velocidad, en lo posible realizando este para distintos parámetros de optimización o funciones objetivo, así permitiendo determinar o no si es realmente posible ajustar el modelo.

Además, y siguiendo en la línea de ajustes a NEURON, es necesario realizar correcciones a la forma en la que se calcula el modelo MRG o al mismo software NEURON para corregir los errores al momento de calcular la función exponencial, lo cual puede sesgar los resultados obtenidos y el análisis de estos.

Por último y particularmente después de lograr reducir los tiempos de computo, es necesario repetir el experimento múltiples veces, debido a la naturaleza estocástica de este. Como se menciona anteriormente, esto servirá para asegurar la consistencia del algoritmo en encontrar una solución, y evitar tomar conclusiones en base a una ejecución que pudo ser afectada por aleatoriedad desfavorable.

6.4 OBSERVACIONES FINALES

El trabajo no logró todo lo esperado y es necesario realizar más experimentos para poder concluir algo respecto a la hipótesis. No obstante, se obtuvieron avances y existen varios puntos a rescatar dentro del trabajo, primero que todo desde el lado del algoritmo CMA-ES se logró converger a soluciones para ambos experimentos, donde particularmente los resultados obtenidos en el primero muestran una mejora significativa respecto a los vistos con los parámetros base, mostrando un aumento gradual de la densidad de impulsos según aumenta la intensidad, pero sin ajustarse mayormente a la forma esperada de la curva ni mostrando los resultados esperados para el 4to intervalo de intensidad en adelante. El segundo experimento mostró una mejora menor solo en el aumento de la intensidad pero se encuentra lejos de ser una mejora significativa. Es posible que realizando ejecuciones con mayor cantidad de generaciones y un mayor tiempo de comparación para las curvas se obtengan resultados mejores, particularmente de realizarse sobre los parámetros electrofisiológicos.

Otro punto a rescatar, proveniente del mismo análisis de las soluciones encontradas es el efecto de los distintos parámetros del modelo sobre la curva de densidad, permitiendo observar distintos efectos producto de la modificación de unos cuantos parámetros principalmente. Experimentos futuros pueden realizarse variando únicamente algunos de estos parámetros, ahorrando capacidad de computo.

Finalmente, todo el trabajo realizado queda a disposición de quienes prosigan con este trabajo, particularmente en lo que software desarrollado respecta, el cual ya habiendo resuelto distintos problemas producto de la integración de las distintas herramientas y el uso de estas en un clúster, permitirán continuar con nuevos experimentos, ahorrando tiempo en lo que respecta a desarrollo y configuración de software.

GLOSARIO

Axón: Fibra nerviosa.

Canal iónico: Estructuras proteicas presentes en la cubierta de la fibra nerviosa mediante la cual se hace intercambio de iones entre el interior y exterior de la fibra, produciendo cambios en la diferencia de potencial y otras propiedades electrofisiológicas.

CMA-ES: *Covariance matrix adaptation evolution strategy* o Estrategia evolutiva de adaptación de matrices de covarianza.

Columna Dorsal: Zona torácica de la columna

Impulso antidrómico: Impulso propagado en dirección contraria a la normal (ortodrómica), lejos de los terminales de las fibras nerviosas (Carron et al., 2013)

Mielina (capa): Cobertura que envuelve la fibra nerviosa (axón) y entregando aislamiento eléctrico.

Modelo MRG: Modelo de fibra nerviosa descrito por McIntyre et al. (2002)

Nodo de Ranvier: Zona en el axón que se presenta de forma periódica con ausencia de la capa de mielina, concentrando una alta cantidad de canales iónicos facilitando la conducción de impulsos (of Encyclopaedia Britannica (2018)).

Parámetro electrofisiológico: Parámetros referentes a la naturaleza eléctrica de la fibra nerviosa, como son conductancias o resistencias eléctricas, entre otros.

Parámetro morfológico: Parámetros referentes a la estructura geométrica y forma de la fibra nerviosa, como son los diámetros o largos de la fibra.

Parestesia: Sensación de hormigueo y piquetes en una zona del cuerpo.

SCS, Spinal Cord Stimulation: Estimulación de la médula espinal. Técnica para el tratamiento del dolor en base a un dispositivo insertado bajo la piel para enviar señales eléctricas directo a la médula espinal.

Segmento paranodal: Corresponde a la sección contigua al nodo de Ranvier.

Segmento internodal: Corresponde a la porción entre dos nodos de Ranvier.

Umbral motor: Estimulación mínima necesaria para inducir actividad motora.

REFERENCIAS BIBLIOGRÁFICAS

- Amirdelfan, K., Yu, C., Doust, M. W., Gliner, B. E., Morgan, D. M., Kapural, L., Vallejo, R., Sitzman, B. T., Yearwood, T. L., Bundschu, R., et al. (2018). Long-term quality of life improvement for chronic intractable back and leg pain patients using spinal cord stimulation: 12-month results from the senza-rct. *Quality of Life Research*, 27(8), 2035–2044.
- Biedrzycki, R. (2020). Handling bound constraints in cma-es: An experimental study. *Swarm and Evolutionary Computation*, 52, 100627.
- Bostock, H., & Bergmans, J. (1994). Post-tetanic excitability changes and ectopic discharges in a human motor axon. *Brain*, 117(5), 913–928.
- Bretschneider, F., & De Weille, J. R. (2018). *Introduction to electrophysiological methods and instrumentation*. Academic Press.
- Burke, E. K., Hyde, M. R., Kendall, G., Ochoa, G., Özcan, E., & Woodward, J. R. (2019). A classification of hyper-heuristic approaches: revisited. In *Handbook of metaheuristics*, (pp. 453–477). Springer.
- Carron, R., Chaillet, A., Filipchuk, A., Pasillas-Lépine, W., & Hammond, C. (2013). Closing the loop of deep brain stimulation. *Frontiers in systems neuroscience*, 7, 112.
- Caylor, J., Reddy, R., Yin, S., Cui, C., Huang, M., Huang, C., Rao, R., Baker, D. G., Simmons, A., Souza, D., et al. (2019). Spinal cord stimulation in chronic pain: evidence and theory for mechanisms of action. *Bioelectronic medicine*, 5(1), 1–41.
- Chambers, D., Huang, C., & Matthews, G. (2019). *Basic physiology for anaesthetists*. Cambridge University Press.
- Crosby, N. D., Janik, J. J., & Grill, W. M. (2017). Modulation of activity and conduction in single dorsal column axons by kilohertz-frequency spinal cord stimulation. *Journal of neurophysiology*, 117(1), 136–147.
- Dafny, N. (2020). Anatomy of the spinal cord.
URL <https://nba.uth.tmc.edu/neuroscience/m/s2/chapter03.html>
- Durá, J. L., Solanes, C., De Andrés, J., & Saiz, J. (2019). Computational study of the effect of electrode polarity on neural activation related to paresthesia coverage in spinal cord stimulation therapy. *Neuromodulation: Technology at the Neural Interface*, 22(3), 269–279.
- Forbes, C. S. (2011). *Statistical distributions*. Wiley.
- Fortin, F.-A., De Rainville, F.-M., Gardner, M.-A., Parizeau, M., & Gagné, C. (2012). DEAP: Evolutionary algorithms made easy. *Journal of Machine Learning Research*, 13, 2171–2175.
- Gaines, J. L., Finn, K. E., Slopsema, J. P., Heyboer, L. A., & Polasek, K. H. (2018). A model of motor and sensory axon activation in the median nerve using surface electrical stimulation. *Journal of computational neuroscience*, 45(1), 29–43.
- Gramacki, A. (2018). *Nonparametric Kernel Density Estimation and Its Computational Aspects*. Springer International Publishing.
- Hansen, N. (2006). The cma evolution strategy: a comparing review. *Towards a new evolutionary computation*, (pp. 75–102).
- Hansen, N. (2016). The cma evolution strategy: A tutorial. *arXiv preprint arXiv:1604.00772*.
- Hansen, N., & Ostermeier, A. (2001). Completely derandomized self-adaptation in evolution strategies. *Evolutionary computation*, 9(2), 159–195.

- Hold-Geoffroy, Y., Gagnon, O., & Parizeau, M. (2014). Once you scoop, no need to fork. In *Proceedings of the 2014 Annual Conference on Extreme Science and Engineering Discovery Environment*, (p. 60). ACM.
- Howells, J., Trevillion, L., Bostock, H., & Burke, D. (2012). The voltage dependence of I_h in human myelinated axons. *The Journal of physiology*, 590(7), 1625–1640.
- Huxley, A., & Stämpfli, R. (1949). Evidence for saltatory conduction in peripheral myelinated nerve fibres. *The Journal of physiology*, 108(3), 315.
- Levenstein, D., Alvarez, V. A., Amarasingham, A., Azab, H., Gerkin, R. C., Hasenstaub, A., Iyer, R., Jolivet, R. B., Marzen, S., Monaco, J. D., et al. (2020). On the role of theory and modeling in neuroscience. *arXiv preprint arXiv:2003.13825*.
- Loewe, A., Wilhelms, M., Schmid, J., Krause, M. J., Fischer, F., Thomas, D., Scholz, E. P., Dössel, O., & Seemann, G. (2016). Parameter estimation of ion current formulations requires hybrid optimization approach to be both accurate and reliable. *Frontiers in Bioengineering and Biotechnology*, 3, 209.
URL <https://www.frontiersin.org/article/10.3389/fbioe.2015.00209>
- Mayfield Clinic (2021). Spinal cord stimulation. Retrieved February 20, 2021.
URL <https://mayfieldclinic.com/pe-stim.htm>
- McConnell, K. A., Nahas, Z., Shastri, A., Lorberbaum, J. P., Kozel, F. A., Bohning, D. E., & George, M. S. (2001). The transcranial magnetic stimulation motor threshold depends on the distance from coil to underlying cortex: a replication in healthy adults comparing two methods of assessing the distance to cortex. *Biological psychiatry*, 49(5), 454–459.
- McIntyre, C. C., Richardson, A. G., & Grill, W. M. (2002). Modeling the excitability of mammalian nerve fibers: influence of afterpotentials on the recovery cycle. *Journal of neurophysiology*, 87(2), 995–1006.
- Medina, L. (2020). Comunicación personal.
URL <https://leomedina.diinf.usach.cl/>
- Melzack, R., Wall, P. D., et al. (1965). Pain mechanisms: a new theory. *Science*, 150(3699), 971–979.
- Micheli-Tzanakou, E. (2014). Nervous system. In J. D. Bronzino, & D. R. Peterson (Eds.) *Biomedical Engineering Fundamentals*. CRC Press, 2nd ed.
- Molga, M., & Smutnicki, C. (2005). Test functions for optimization needs. *Test functions for optimization needs*, 101, 48.
- of Encyclopaedia Britannica, T. E. (2018). Node of ranvier. Retrieved February 20, 2021.
URL <https://www.britannica.com/science/node-of-Ranvier>
- Owusu, S., Huynh, A., Gruenthal, E., Prusik, J., Owusu-Sarpong, S., Cherala, R., Peng, S., Pilitsis, J. G., & McCallum, S. E. (2017). Prospective evaluation of patient usage of above and below threshold waveforms with traditional spinal cord stimulation devices. *Neuromodulation: Technology at the Neural Interface*, 20(6), 567–574.
- Rea, P. (2015). *Essential clinical anatomy of the nervous system*. Academic Press.
- Rumbell, T. H., Draguljić, D., Yadav, A., Hof, P. R., Luebke, J. I., & Weaver, C. M. (2016). Automated evolutionary optimization of ion channel conductances and kinetics in models of young and aged rhesus monkey pyramidal neurons. *Journal of Computational Neuroscience*, 41(1), 65–90.
- Rydmark, M. (1981). Nodal axon diameter correlates linearly with internodal axon diameter in spinal roots of the cat. *Neuroscience letters*, 24(3), 247–250.

- Scholz, A., Reid, G., Vogel, W., & Bostock, H. (1993). Ion channels in human axons. *Journal of neurophysiology*, 70(3), 1274–1279.
- Scott, D. W. (2015). *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons.
- Sdrulla, A. D., Guan, Y., & Raja, S. N. (2018). Spinal cord stimulation: clinical efficacy and potential mechanisms. *Pain Practice*, 18(8), 1048–1067.
- Sterratt, D., Graham, B., Gillies, A., & Willshaw, D. (2011). *Principles of Computational Modelling in Neuroscience*. Cambridge University Press.
- Szynkiewicz, P. (2018). Comparative study of pso and cma-es algorithms on black-box optimization benchmarks. *Journal of Telecommunications and Information Technology*.
- Talbi, E.-G. (2009). *Metaheuristics: from design to implementation*, vol. 74. John Wiley & Sons.
- Thomson, S. (2019). Spinal cord stimulation. Retrieved February 20, 2021.
URL <https://www.neuromodulation.com/spinal-cord-stimulation>
- Varelas, K., Auger, A., Brockhoff, D., Hansen, N., ElHara, O. A., Semet, Y., Kassab, R., & Barbaresco, F. (2018). A comparative study of large-scale variants of cma-es. In *International Conference on Parallel Problem Solving from Nature*, (pp. 3–15). Springer.
- Witte, R. S., & Witte, J. S. (2019). *Statistics*. Wiley.
- Xu, H., Shinozaki, T., & Kobayashi, R. (2019). Effective and stable neuron model optimization based on aggregated cma-es. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (pp. 1264–1268). IEEE.
- Yang, X.-S., & Koziel, S. (2011). Computational optimization: An overview. *Computational Optimization, Methods and Algorithms*, (pp. 1–11).
- Zhang, T. C., Janik, J. J., & Grill, W. M. (2014). Mechanisms and models of spinal cord stimulation for the treatment of neuropathic pain. *Brain research*, 1569, 19–31.

APÉNDICE A. GRÁFICOS DE RESPUESTA ELÉCTRICA ADICIONALES

Acá se registran gráficos adicionales para observar el comportamiento eléctrico del modelo de fibra para distintos parámetros.

A.1 PARÁMETROS BASE

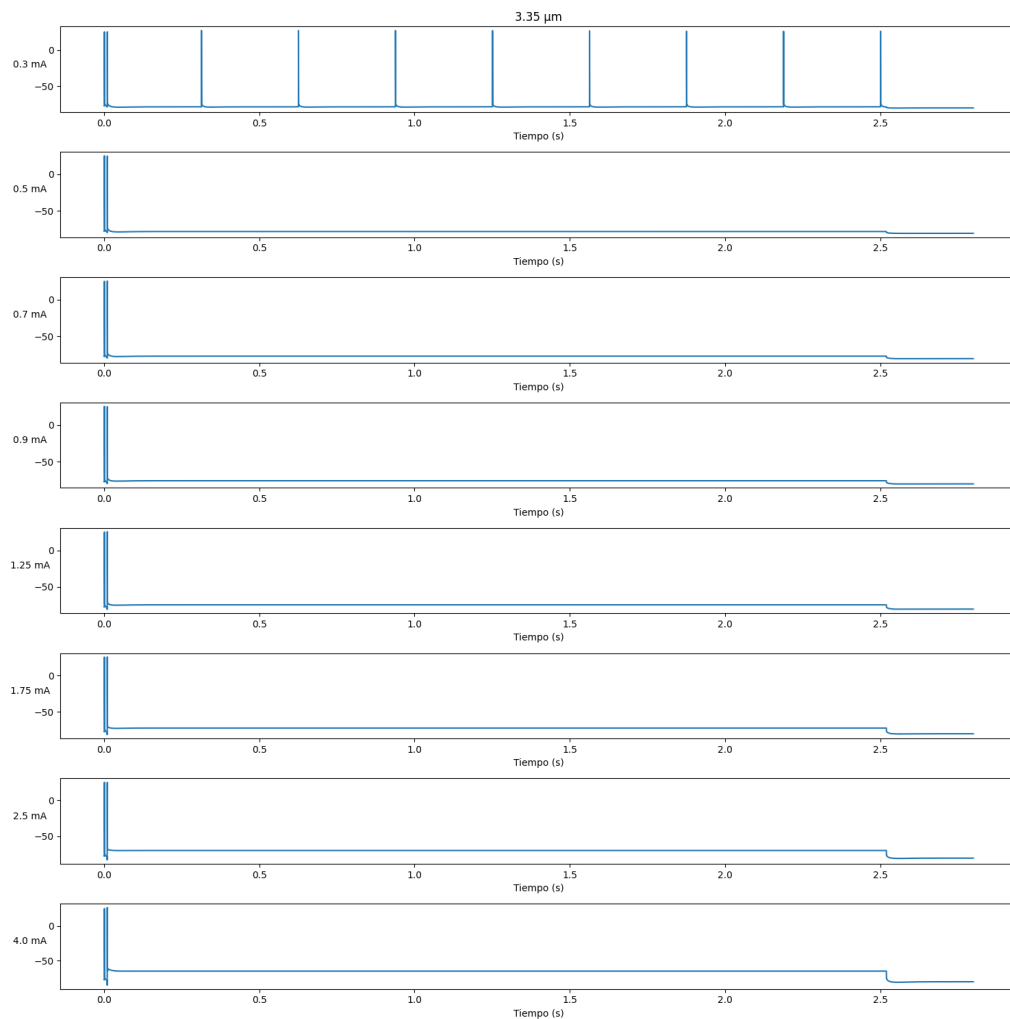


Figura A.1: Respuesta eléctrica para fibras de 3.35 micrómetros para cada rango de amplitud durante 2.8 segundos, usando parámetros base.

Fuente: Elaboración propia (2021)

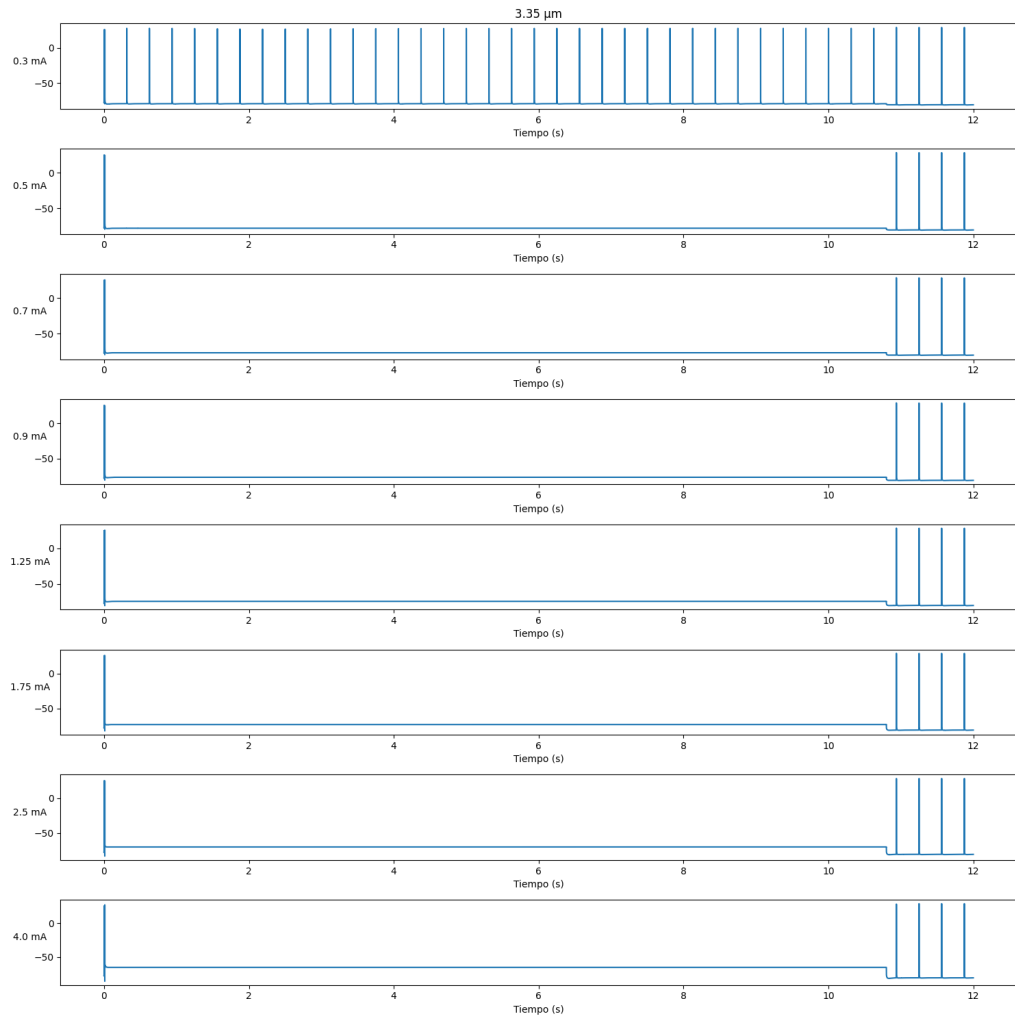


Figura A.2: Respuesta eléctrica para fibras de 3.35 micrómetros para cada rango de amplitud durante 12 segundos, usando parámetros base.
Fuente: Elaboración propia (2021)

A.2 PARÁMETROS OBTENIDOS MEDIANTE CMA-ES APLICADO SOLO A PARÁMETROS ELECTROFISIOLÓGICOS

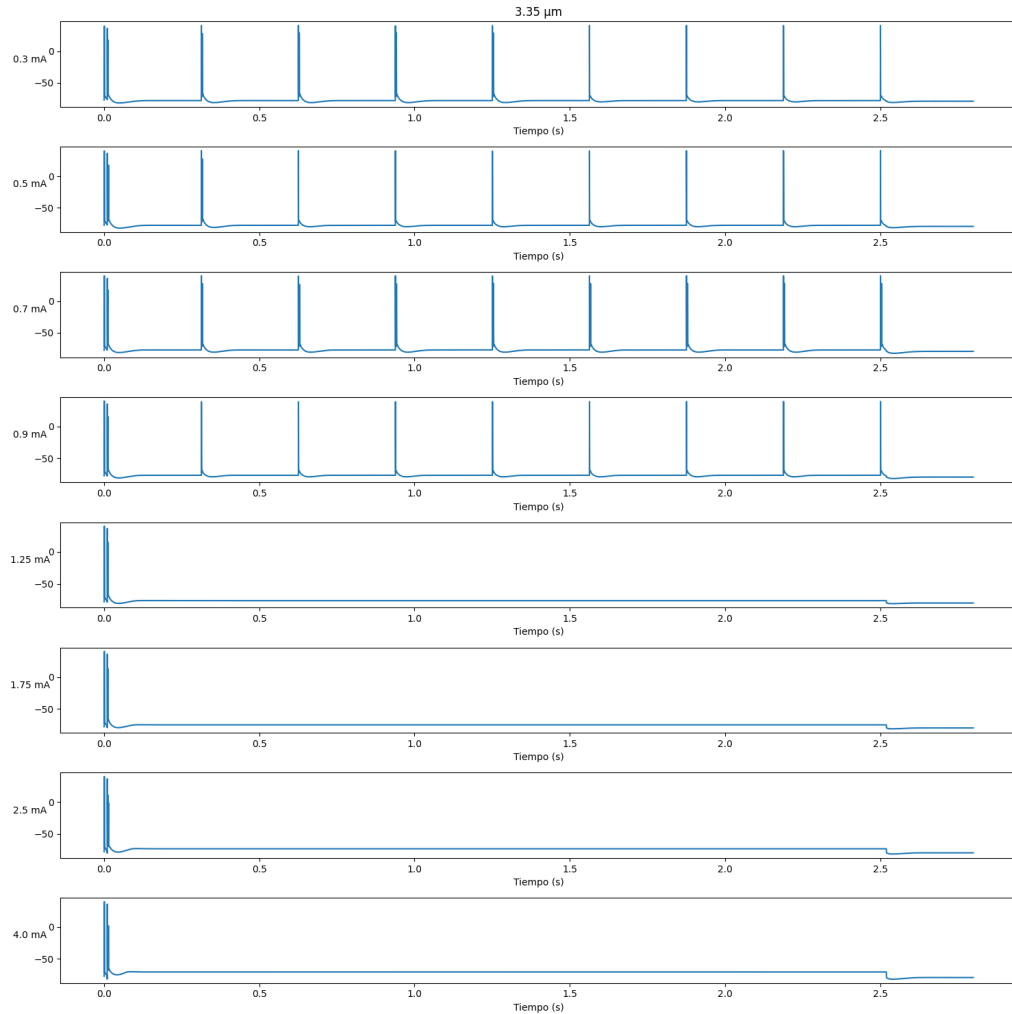


Figura A.3: Respuesta eléctrica para fibras de 3.35 micrómetros para cada rango de amplitud durante 2.8 segundos, usando parámetros obtenidos mediante CMA-ES aplicado únicamente a parámetros electrofisiológicos.

Fuente: Elaboración propia (2021)

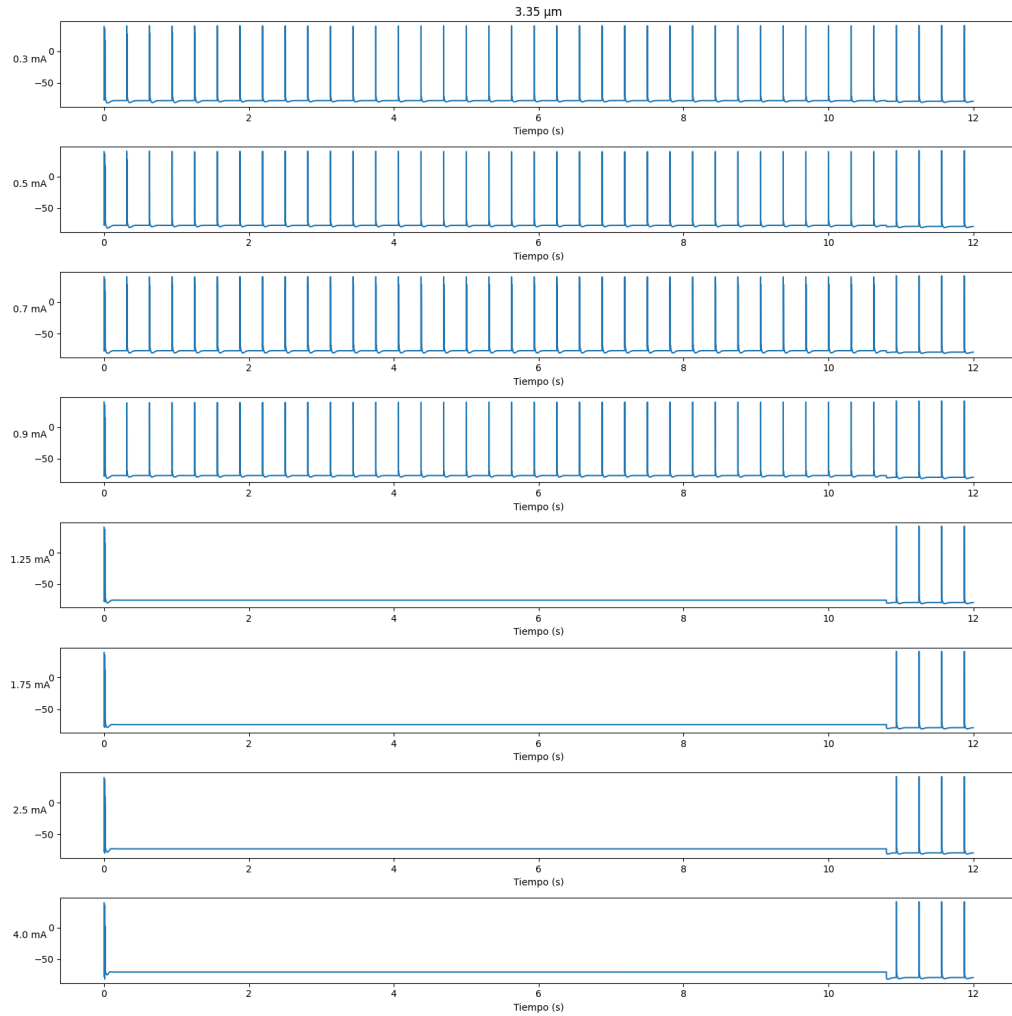


Figura A.4: Respuesta eléctrica para fibras de 3.35 micrómetros para cada rango de amplitud durante 12 segundos, usando parámetros obtenidos mediante CMA-ES aplicado únicamente a parámetros electrofisiológicos.

Fuente: Elaboración propia (2021)

A.3 PARÁMETROS OBTENIDOS MEDIANTE CMA-ES APLICADO A PARÁMETROS ELECTROFISIOLÓGICOS Y MORFOLÓGICOS

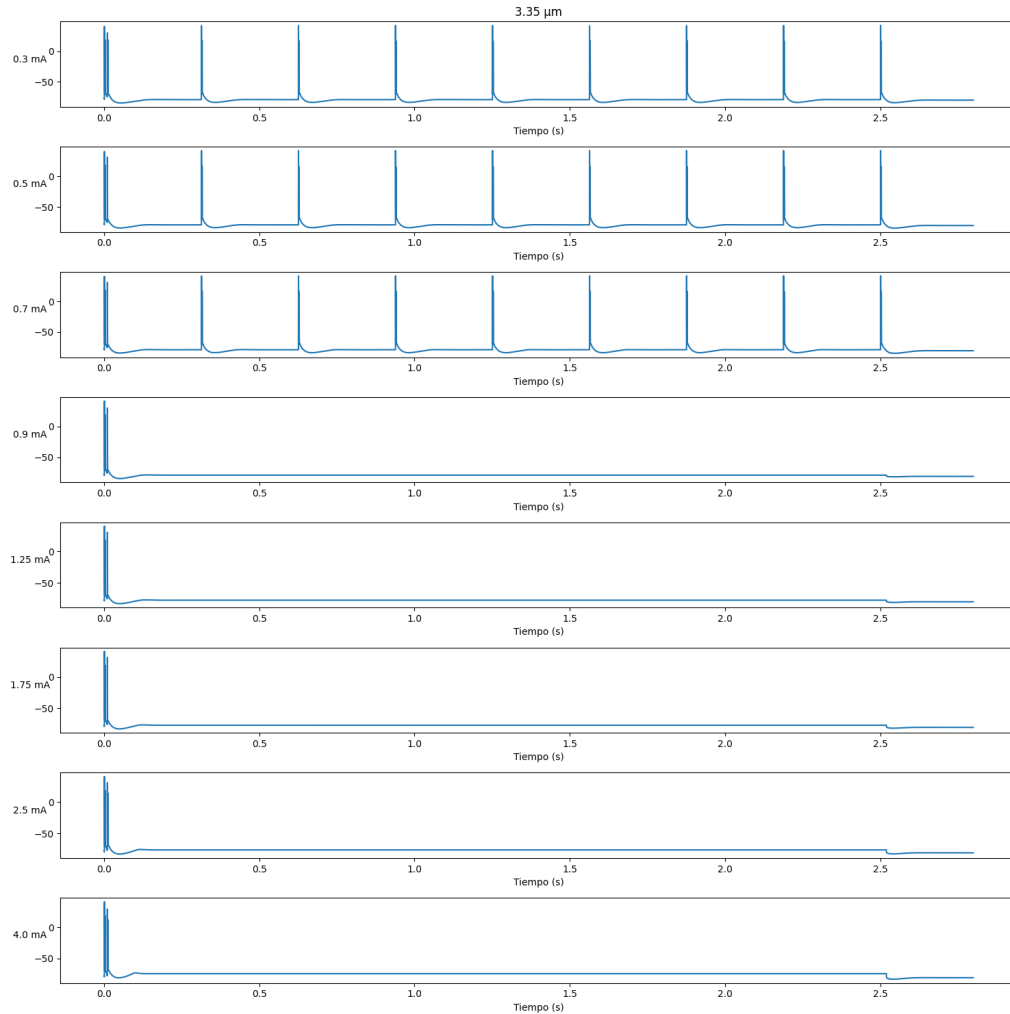


Figura A.5: Respuesta eléctrica para fibras de 3.35 micrómetros para cada rango de amplitud durante 2.8 segundos, usando parámetros obtenidos mediante CMA-ES aplicado a todo el conjunto de parámetros.

Fuente: Elaboración propia (2021)

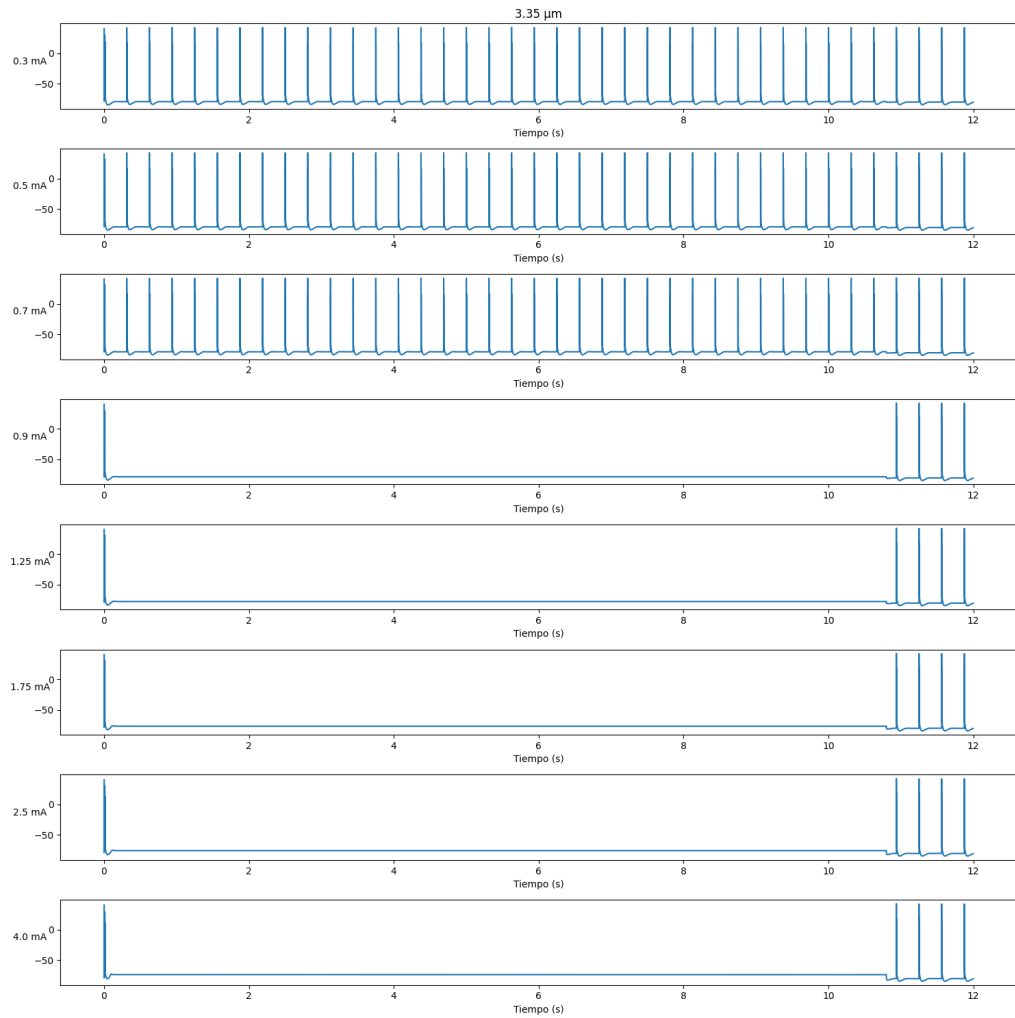


Figura A.6: Respuesta eléctrica para fibras de 3.35 micrómetros para cada rango de amplitud durante 12 segundos, usando parámetros obtenidos mediante CMA-ES aplicado únicamente a todo el conjunto de parámetros.

Fuente: Elaboración propia (2021)

APÉNDICE B. GRÁFICOS DE ANÁLISIS DE SENSIBILIDAD DE RESPUESTA ELÉCTRICA ADICIONALES

Acá se describen gráficos de respuesta eléctrica adicionales para analizar la sensibilidad de modificar distintos parámetros.

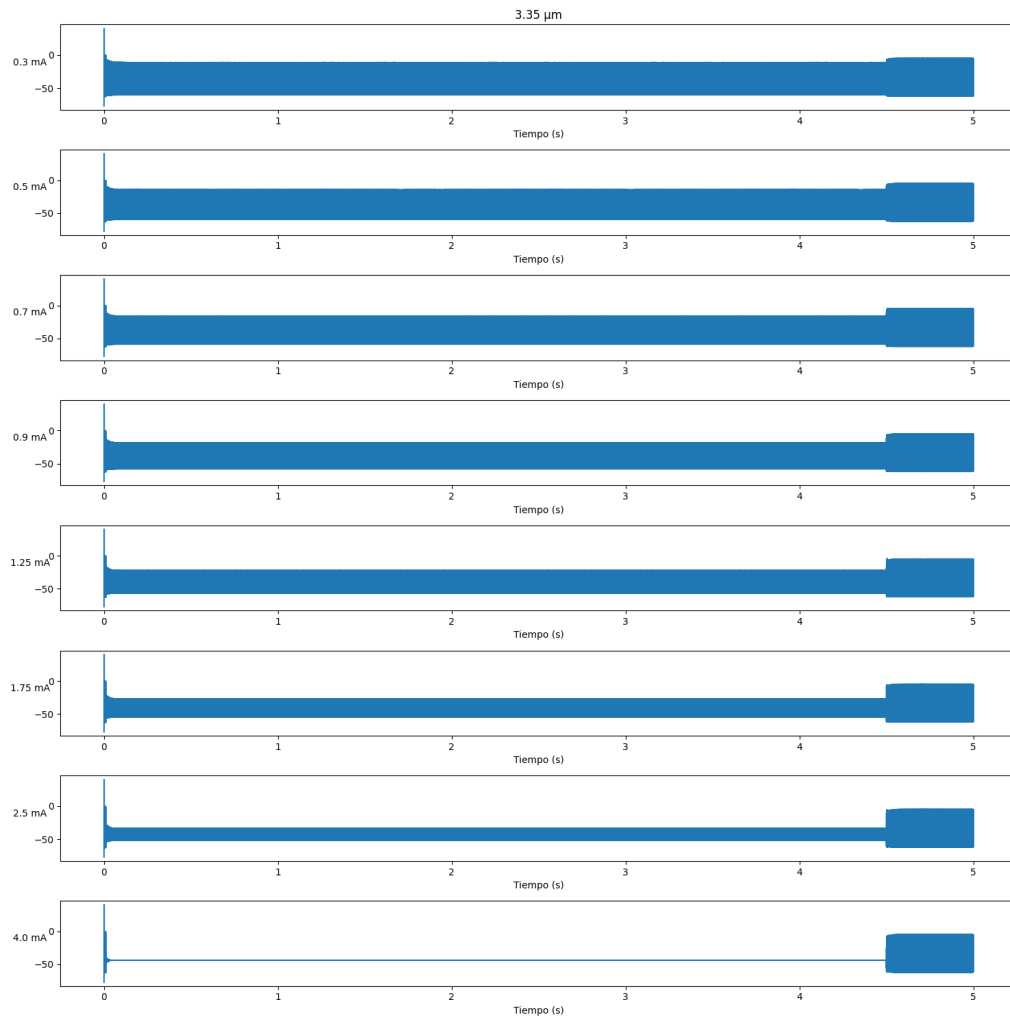


Figura B.1: Respuesta eléctrica para una fibra de 3.35 micrómetros, modificando únicamente la conductancia máxima de sodio según el valor obtenido utilizando CMA-ES únicamente sobre los parámetros electrofisiológicos.

Fuente: Elaboración propia (2021)

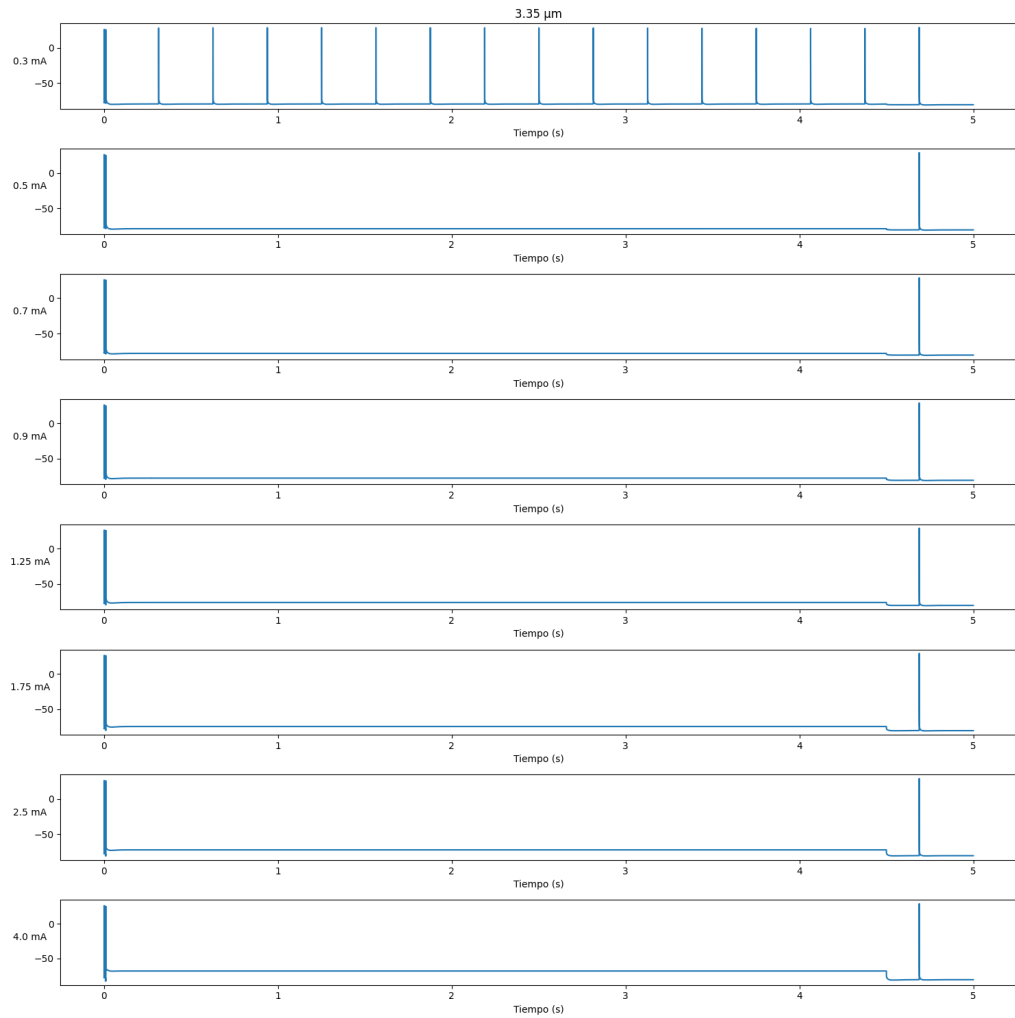


Figura B.2: Respuesta eléctrica para una fibra de 3.35 micrómetros, modificando únicamente las dos componentes del diámetro axonal según el valor obtenido utilizando CMA-ES para todos los parámetros.

Fuente: Elaboración propia (2021)