



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA MATEMÁTICA

ANÁLISIS SOBRE MÉTODOS DE AJUSTE Y APRENDIZAJE DE MÁQUINAS
APLICADOS A LA EQUIVALENCIA Y REDUCCIÓN DE MODELOS DE
ELECTROFISIOLOGÍA CARDIACA

TESIS PARA OPTAR AL GRADO DE
MAGÍSTER EN CIENCIAS DE LA INGENIERÍA, MENCIÓN MATEMÁTICAS
APLICADAS

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL MATEMÁTICO

MANUEL IGNACIO SUIL JORQUERA

PROFESOR GUÍA:
AXEL OSSES ALVARADO

PROFESORES COGUÍA:
DANIEL HURTADO SEPÚLVEDA
FRANCISCO SAHLI COSTABAL

COMISIÓN:
FELIPE TOBAR HENRÍQUEZ

Este trabajo ha sido parcialmente financiado por Proyecto Fondecyt 1191903, Proyecto Apoyo Financiamiento Basal CMM ANID PIA AFB-170001 y Millennium Nucleus for Cardiovascular Magnetic Resonance.

Powered@NLHPC: Esta tesis fue parcialmente apoyada por la infraestructura de supercómputo del NLHPC (ECM-02)

SANTIAGO DE CHILE

2021

Resumen

Las cardiopatías siguen siendo una de las principales causas de decesos a pesar de los avances en tratamientos y campañas preventivas, muchas veces una detección temprana ha significado salvar vidas.

Entender la actividad cardiaca es esencial para la detección temprana de enfermedades asociadas al corazón. Por lo mismo, la modelación de la actividad eléctrica en este órgano, sumado a su simulación, son de gran importancia para la creación y mejoramiento de herramientas que permitan detectar cardiopatías sobre la base de la observación de señales provenientes de la difusión del voltaje.

Esta investigación consiste en introducir dos metodologías de trabajo que permiten buscar otra alternativa para reconstruir las soluciones provenientes de modelos asociados a esta temática, para esto, incluye, en primer lugar, una introducción a la electrofisiología, como también algunos modelos que han trabajado esta temática; luego, se analizan las dos metodologías propuestas, donde el primer método permite obtener un modelo cualitativo equivalente a uno de los estudiados, independiente si este último es cuantitativo o no; el segundo, permite obtener la solución de una reducción del modelo original mediante el uso de redes neuronales; finalmente, el análisis consiste en estudiar los resultados numéricos al aplicar los métodos sobre la solución de los modelos introducidos en este trabajo.

Dedicado a mi familia.

Agradecimientos

Agradezco en primer lugar a mi familia, a mis padres por hacer todo lo posible para que mi hermana y yo tengamos la mejor educación, como también por su apoyo incondicional en los momentos más difíciles. A mi hermana por alegrarme los días, ya que a pesar de no poder verla debido a la cuarentena, sus llamadas han sido muy importantes para mí especialmente en estos últimos meses. Agradezco a mis amigos por los lindos momentos vividos y que nunca olvidaré, desde las emocionantes partidas de “Uno” (y variaciones) en la oficina 435, hasta aquellos fines de semana de intenso y, casi siempre, productivo estudio grupal. Agradezco a cada una de las personas presentes en mi formación académica, en especial, está mi gratitud a mis profesores guías Axel Osses, Daniel Hurtado y Francisco Sahli, por acompañarme en la culminación con este proceso de aprendizaje y, por cada una de las ideas compartidas que han sido esenciales para concluir este trabajo.

Tabla de Contenido

Introducción	1
1. Presentación del problema y objetivos	2
1.1. Ritmo cardiaco y propagación de voltaje	3
1.2. Potencial de acción	4
1.3. Modelamiento matemático en electrofisiología cardiaca	5
1.4. Modelos a estudiar	6
1.4.1. Hodgkin-Huxley (1952)	6
1.4.2. Morris-Lecar (1981)	8
1.4.3. FitzHugh-Nagumo (1961)	9
1.4.4. ten Tusscher - Panfilov (2006)	11
1.5. Costo computacional y característica de los modelos estudiados	13
2. Equivalencia de modelos a través de métodos de ajuste de nulclinas	15
2.1. Ajuste de funciones de densidad de energía	15
2.2. Método de ajuste de nulclinas	18
2.3. Resultados de equivalencia a partir del método de ajuste de nulclinas	20
3. Método de reducción de variables usando aprendizaje de máquinas	32
3.1. Introducción al método de reducción de variables usando POD	32
3.2. Autoencoders como generalización de POD	34
3.2.1. Arquitectura de la red	35
3.2.2. Entrenamiento de la red	36
3.2.3. Medición de rendimiento	38
3.2.4. Recuperación del sistema de ecuaciones reducido	38
3.3. Resultados del método de reducción de variables usando Autoencoders	43
4. Discusión	58
Conclusión	61
Bibliografía	62
Anexo	63

Índice de Tablas

1.1.	Resumen de las características de los modelos estudiados.	13
1.2.	Características generales de los modelos de tipo Fenomenológico y Biofísico. .	14
2.1.	Valores aproximados de las funciones de ajuste en cada región.	23
2.2.	Valores aproximados de las funciones de ajuste en cada región, caso polinomial a trozos.	26
2.3.	Tiempo medido en la resolución del sistema. Condiciones iniciales: $V_0 = -65$, $\omega_0 = 0,4$	31
3.1.	Valores obtenidos para cada métrica de la distancia entre el valor de salida y de entrada.	44
4.1.	Valores de las constantes usadas en el modelo de Hodgkin-Huxley.	64
4.2.	Funciones α y β , utilizadas en 1.15, 1.16 y 1.17.	64
4.3.	Valores de las constantes usadas en el modelo de Morris-Lecar.	65
4.4.	Funciones m_∞ , ω_∞ y τ , usadas en (1.18) y (1.19).	65
4.5.	Valores de las constantes usadas en el modelo de FitzHugh-Nagumo.	65
4.6.	Corrientes totales usadas en (1.27).	66
4.7.	Otras corrientes usadas en el sistema y en la Tabla (4.6).	66
4.8.	Funciones usadas en el sistema de ecuaciones de ten Tusscher-Panfilov. . . .	68
4.9.	Valores de las constantes físicas usadas en el modelo de ten Tusscher-Panfilov. .	69
4.10.	Potenciales de membrana (o de <i>Nerst</i>) usadas en el modelo de ten Tusscher-Panfilov.	69
4.11.	Concentraciones externas usadas en el modelo de ten Tusscher-Panfilov. . . .	69
4.12.	Otras constantes usadas en el modelo de ten Tusscher-Panfilov.	70

Índice de Ilustraciones

1.1.	Estructura del corazón y trayecto de flujo a través de sus cavidades y válvulas ¹ .	2
1.2.	Red de propagación de potenciales de acción a través del corazón ² .	3
1.3.	Gráfico de ejemplo de potencial de acción sobre una célula con respecto al tiempo, especificando las corrientes asociadas a cada una de las 5 fases, el tiempo está medido en milisegundos. Fuente: <i>Mathematical Cardiac Electrophysiology</i> ([10]).	4
1.4.	Modelo de circuito (Hodgkin & Huxley,[3]), obtenido a partir del análisis de la estructura de la membrana celular.	5
1.5.	Solución obtenida al simular el modelo de Hodgkin-Huxley en MatLab. Los valores de constantes son las mismas que en la tabla (4.1). Mientras que la corriente aplicada es $I_{app} = 5$.	8
1.6.	Solución obtenida al simular el modelo de Morris-Lecar en MatLab. Los valores de constantes son las mismas que en la Tabla (4.3). La corriente aplicada es $I_{app} = 90$.	9
1.7.	Solución obtenida al simular el modelo de FitzHugh-Nagumo en MatLab. Los valores de constantes utilizadas se encuentran en la Tabla (4.5), en el <i>Anexo</i> . La corriente aplicada es $I_{app} = 0,02$.	11
1.8.	Voltaje y corrientes asociadas a las dinámicas de iones de Sodio, Potasio y Calcio, sobre una célula del <i>epicardio</i> . Resultados obtenidos al ejecutar simulaciones del modelo de ten Tusscher-Panfilov en <i>MyoKit</i> . Corriente aplicada $I_{app} = -52 [A/F]$ en el tiempo $t = 50 [ms]$.	13
2.1.	Valores de E con respecto a V y R . Caso en el cual se está cerca de un mínimo local con respecto a V , pero con respecto a R aún no se encuentra en un punto estable, provocando lo que se muestra en la Figura (2.2).	16
2.2.	Valores de E con respecto a V y R . Caso en el cual se está cerca de un máximo local con respecto a R , pero con respecto a V aún no se encuentra en un punto estable, provocando lo que se muestra en la Figura (2.3).	17
2.3.	Valores de E con respecto a V y R . Caso en el cual vuelve a estar cerca de un mínimo local con respecto a V , pero vuelve a alejarse de un punto estable con respecto a R , provocando la continuación del ciclo.	17
2.4.	Superficie que describe E con respecto al plano (V, R) .	18
2.5.	Nulclinas sobre un diagrama de fases de modelo de FitzHugh-Nagumo.	19
2.6.	Ajuste de parámetros de FitzHugh-Nagumo sobre modelo de Morris-Lecar.	21
2.7.	Nulclinas asociadas a cada modelo.	21

2.8.	Ajuste polinomial de nulclinas de <i>Morris-Lecar</i> . Izquierda: Ajuste de nulclina asociada a V . Derecha: Ajuste de nulclina asociada a ω	22
2.9.	Regiones utilizadas para cálculo de funciones de ajuste de velocidades.	23
2.10.	Comportamiento de Voltaje (izquierda) y variable de compuerta ω (derecha) para el modelo polinomial (izquierda) y el de <i>Morris-Lecar</i> (derecha). Condiciones iniciales: $V_0 = -65$, $\omega_0 = 0,4$	24
2.11.	Trayectoria de simulación sobre plano de fases para modelos de <i>Morris-Lecar</i> y Polinomial. Condiciones iniciales: $V_0 = -65$, $\omega_0 = 0,4$	24
2.12.	Ajuste polinomial por trozos de nulclinas de <i>Morris-Lecar</i> . Izquierda: Ajuste de nulclina asociada a V . Derecha: Ajuste de nulclina asociada a ω	26
2.13.	Comportamiento de Voltaje (izquierda) y variable de compuerta ω (derecha) para el modelo polinomial por partes (izquierda) y el de <i>Morris-Lecar</i> (derecha). Condiciones iniciales: $V_0 = -65$, $\omega_0 = 0,56$	27
2.14.	Trayectoria de simulación sobre plano de fases para modelos de <i>Morris-Lecar</i> y Polinomial por partes. Condiciones iniciales: $V_0 = -65$, $\omega_0 = 0,56$	27
2.15.	Comportamiento de Voltaje (izquierda) y variable de compuerta ω (derecha) para el modelo polinomial por partes (izquierda) y el de <i>Morris-Lecar</i> (derecha). Condiciones iniciales: $V_0 = -35$, $\omega_0 = 0,4$	28
2.16.	Trayectoria de simulación sobre plano de fases para modelos de <i>Morris-Lecar</i> y Polinomial por partes. Condiciones iniciales: $V_0 = -35$, $\omega_0 = 0,4$	28
2.17.	Comportamiento de Voltaje (izquierda) y variable de compuerta ω (derecha) para el modelo polinomial por partes (izquierda) y el de <i>Morris-Lecar</i> (derecha). Condiciones iniciales: $V_0 = 45$, $\omega_0 = 0,04$	29
2.18.	Trayectoria de simulación sobre plano de fases para modelos de <i>Morris-Lecar</i> y Polinomial por partes. Condiciones iniciales: $V_0 = 45$, $\omega_0 = 0,04$	29
2.19.	Comportamiento de Voltaje (izquierda) y variable de compuerta ω (derecha) para el modelo polinomial por partes (izquierda) y el de <i>Morris-Lecar</i> (derecha). Condiciones iniciales: $V_0 = -16,5$, $\omega_0 = 0,18$	30
2.20.	Trayectoria de simulación sobre plano de fases para modelos de <i>Morris-Lecar</i> y Polinomial por partes. Condiciones iniciales: $V_0 = -16,5$, $\omega_0 = 0,18$	30
3.1.	Recuperación de la variable de voltaje usando POD sobre los snapshots provenientes del modelo de <i>ten Tusscher-Panfilov</i>	34
3.2.	Caso de recuperación fallida usando POD sobre los snapshots provenientes del modelo de <i>ten Tusscher-Panfilov</i>	34
3.3.	Esquema básico de un autoencoder ³	35
3.4.	Primer esquema de reducción usando Autoencoders sobre modelo HH.	43
3.5.	Resultado de reconstrucción usando dos diferentes normas en un elemento del conjunto de test.	44
3.6.	Porcentaje de test que cumplen el criterio (3.6).	45
3.7.	Primera variable del representante latente.	45
3.8.	Nuevo esquema de reducción usando Autoencoders sobre modelo HH.	46
3.9.	Code obtenido a partir de input mostrado en las Figuras (3.10), (3.11), (3.12) y (3.13).	46
3.10.	Voltaje [mV] con respecto al tiempo en [ms]. Caso test cuyo error relativo es menor al valor indicado. 2 capas de Encoder y 8 de Decoder	47

3.11. m con respecto al tiempo en $[ms]$. Caso test cuyo error relativo es menor al valor indicado. 2 capas de Encoder y 8 de Decoder	47
3.12. h con respecto al tiempo en $[ms]$. Caso test cuyo error relativo es menor al valor indicado. 2 capas de Encoder y 8 de Decoder	47
3.13. n con respecto al tiempo en $[ms]$. Caso test cuyo error relativo es menor al valor indicado. 2 capas de Encoder y 8 de Decoder	48
3.14. Voltaje $[mV]$ con respecto al tiempo en $[ms]$. Caso test cuyo error relativo es mayor al valor indicado. 2 capas de Encoder y 8 de Decoder	48
3.15. m con respecto al tiempo en $[ms]$. Caso test cuyo error relativo es mayor al valor indicado. 2 capas de Encoder y 8 de Decoder	48
3.16. h con respecto al tiempo en $[ms]$. Caso test cuyo error relativo es mayor al valor indicado. 2 capas de Encoder y 8 de Decoder	49
3.17. n con respecto al tiempo en $[ms]$. Caso test cuyo error relativo es mayor al valor indicado. 2 capas de Encoder y 8 de Decoder	49
3.18. Valor de función de pérdida para distintas configuraciones de capas.	49
3.19. Tiempo de entrenamiento por época en promedio para distintas configuraciones de capas.	50
3.20. Caso de reconstrucción deficiente.	50
3.21. Reconstrucción de V	51
3.22. Reconstrucción de Ca_i	51
3.23. Reconstrucción de Na_i	51
3.24. Reconstrucción de K_i	52
3.25. Función de pérdida para diferentes tamaño de reducción. POD y Autoencoders en ten Tusscher-Panfilov.	52
3.26. Error relativo máximo para diferentes tamaño de reducción usando el conjunto test. POD y Autoencoders en ten Tusscher-Panfilov.	53
3.27. Voltaje original y reconstruido usando 4 dígitos.	53
3.28. Voltaje original y reconstruido usando 2 dígitos.	54
3.29. Ejemplo de m original y reconstruido.	56
3.30. Ejemplo de h original y reconstruido.	56
3.31. Ejemplo de n original y reconstruido.	57
4.1. Variable x_{r2}	59
4.2. Variable ω_6	60
4.3. Variable ω_1 de reducción. Usando POD.	70
4.4. Variable ω_2 de reducción. Usando POD.	70
4.5. Variable ω_3 de reducción. Usando POD.	71
4.6. Variable ω_4 de reducción. Usando POD.	71
4.7. Variable ω_5 de reducción. Usando POD.	71
4.8. Variable ω_6 de reducción. Usando POD.	72
4.9. Variable ω_7 de reducción. Usando POD.	72
4.10. Variable ω_8 de reducción. Usando POD.	72
4.11. Variable ω_9 de reducción. Usando POD.	73
4.12. Variable ω_{10} de reducción. Usando POD.	73
4.13. Variable ω_{11} de reducción. Usando POD.	73
4.14. Variable ω_{12} de reducción. Usando POD.	74
4.15. Variable ω_{13} de reducción. Usando POD.	74

4.16. Reconstrucción de V . Usando POD.	74
4.17. Reconstrucción de Ca_i . Usando POD.	75
4.18. Reconstrucción de Ca_{sr} . Usando POD.	75
4.19. Reconstrucción de Ca_{ss} . Usando POD.	75
4.20. Reconstrucción de R_{prime} . Usando POD.	76
4.21. Reconstrucción de Na_i . Usando POD.	76
4.22. Reconstrucción de K_i . Usando POD.	76
4.23. Reconstrucción de x_{r1} . Usando POD.	76
4.24. Reconstrucción de x_{r2} . Usando POD.	77
4.25. Reconstrucción de x_s . Usando POD.	77
4.26. Reconstrucción de m . Usando POD.	77
4.27. Reconstrucción de h . Usando POD.	77
4.28. Reconstrucción de j . Usando POD.	78
4.29. Reconstrucción de d . Usando POD.	78
4.30. Reconstrucción de f . Usando POD.	78
4.31. Reconstrucción de $f2$. Usando POD.	78
4.32. Reconstrucción de $fCass$. Usando POD.	79
4.33. Reconstrucción de r . Usando POD.	79
4.34. Reconstrucción de s . Usando POD.	79
4.35. Variable ω_1	80
4.36. Variable ω_2	80
4.37. Variable ω_3	80
4.38. Variable ω_4	81
4.39. Variable ω_5	81
4.40. Variable ω_6	81
4.41. Variable ω_7	81
4.42. Variable ω_8	82
4.43. Variable ω_9	82
4.44. Variable ω_{10}	82
4.45. Reconstrucción de Ca_{sr}	83
4.46. Reconstrucción de Ca_{ss}	83
4.47. Reconstrucción de R_{prime}	83
4.48. Reconstrucción de x_{r1}	84
4.49. Reconstrucción de x_{r2}	84
4.50. Reconstrucción de x_s	84
4.51. Reconstrucción de m	85
4.52. Reconstrucción de h	85
4.53. Reconstrucción de j	85
4.54. Reconstrucción de d	86
4.55. Reconstrucción de f	86
4.56. Reconstrucción de $f2$	86
4.57. Reconstrucción de $fCass$	87
4.58. Reconstrucción de r	87
4.59. Reconstrucción de s	87

Introducción

Entender la actividad cardiaca es esencial para la detección temprana de enfermedades asociadas al corazón (cardiopatías). Por lo mismo, la modelación de la actividad eléctrica en este órgano, sumado a su simulación son de gran importancia para la creación y mejoramiento de herramientas que permitan detectar cardiopatías en base a la observación de señales provenientes de la difusión del voltaje.

Diversos modelos se han postulado y estudiado para explicar este fenómeno.

Estos modelos pueden ser cualitativos, también llamados *fenomenológicos*, o cuantitativos, denominados *biofísicos*.

La complejidad de estos modelos ha provocado la necesidad de buscar formas de simplificar la obtención de soluciones, de esta forma, en otros trabajos como *Quarteroni* [12], *Kerschen* [5] o *Lee* [8] se estudian modelos de orden reducido (ROM) o técnicas de reducción de orden. Este trabajo cuenta con cuatro capítulos, el primero consiste en presentar el problema y ser una introducción para quienes no han realizado investigación sobre electrofisiología cardiaca, previamente.

El segundo capítulo presenta un método para la obtención de un modelo equivalente a partir de las realizaciones de uno original mediante el ajuste de coeficientes polinomiales.

El tercer capítulo muestra un método de reducción en la cantidad de variables que presenta un sistema a través de una generalización de POD, usando redes neuronales.

Finalmente, el cuarto capítulo cuenta con una breve discusión de los resultados. El documento cuenta, además, con un Anexo donde se muestran los valores numéricos de las constantes usados en las simulaciones, algunos resultados numéricos, como también información sobre software y paquetes utilizados para la obtención de los resultados vistos en este trabajo.

Capítulo 1

Presentación del problema y objetivos

El corazón es el órgano principal del sistema circulatorio, su función consiste en recibir e impulsar la sangre a través de las arterias, tarea importante para el transporte de la sangre a lo largo del cuerpo. Para realizar esta tarea, el corazón está separado en dos regiones, una encargada de bombear la sangre recibida desde el resto del cuerpo hacia los pulmones, los cuales se encargan de oxigenar la sangre, y otra que bombea la sangre oxigenada hacia el resto del cuerpo.

Cada una de las dos partes del corazón se separan en una *aurícula* y un *ventrículo*. La aurícula actúa como receptor de la sangre, que luego impulsa la sangre hacia el ventrículo, siendo este último, el sector en el cual la sangre es bombeada a través de una arteria.

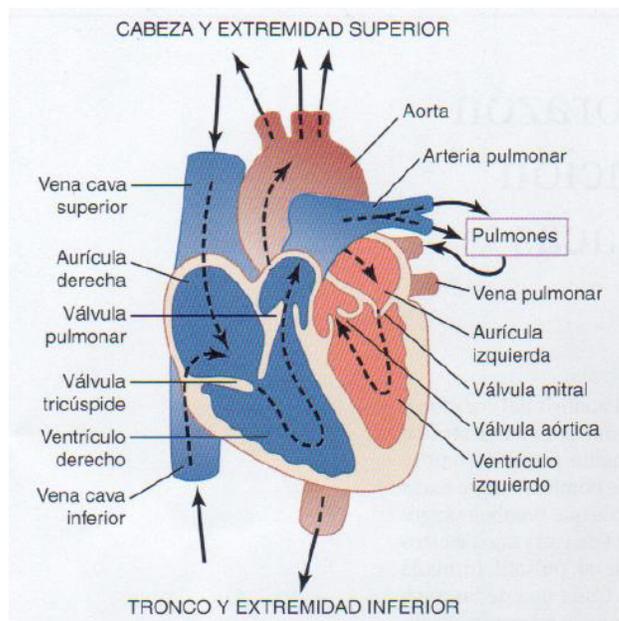


Figura 1.1: Estructura del corazón y trayecto de flujo a través de sus cavidades y válvulas¹.

Los mecanismos del corazón se producen gracias a una sucesión continuada de contracciones

¹Fuente: *Texas Heart Institute*.

denominada ritmicidad cardíaca, que transmite potenciales de acción (voltaje) a través de todo el músculo cardíaco.

1.1. Ritmo cardíaco y propagación de voltaje

Los potenciales de acción, se generan a partir de tejido especializado de excitación. En el corazón, el impulso se genera en el *nódulo sinoauricular* (o *nódulo sinusal*), el cual es una banda de músculo cardíaco especializado, localizada en la pared posterolateral superior de la aurícula derecha. Las fibras de este nódulo tienen la capacidad de *autoexcitación*, que es un proceso que puede producir descargas y contracciones rítmicas automáticas, siendo de suma importancia para controlar la frecuencia cardíaca. Las fibras del nódulo se conectan directamente con las fibras musculares auriculares, propagando de forma inmediata el potencial de acción a través de las vías internodulares, localizadas en las paredes de la aurícula derecha hasta llegar al *nódulo auriculoventricular* o *nódulo AV*. La función del nódulo AV es retrasar la transmisión del potencial de acción, este retraso da tiempo para que las aurículas vacíen su sangre hacia los ventrículos, actuando como un marcapasos natural.

El nódulo AV, a su vez, está conectado a otros conductos denominados como *Haz AV*, cuya función es la de propagar el potencial de acción a la aurícula izquierda y a través de los ventrículos, en este último caso, se logra a través de dos ramas del *Haz AV*, que a su vez, se ramifican formando el sistema de *Purkinje*, cuya conducción alcanza una mayor velocidad que en el nódulo AV y que se localiza en la pared cardíaca, bajo en *Endocardio* (capa de la pared que delimita esta con el interior del corazón). Finalmente, la contracción se produce en el *Miocardio*, capa interior de la pared cardíaca. Un resumen se puede observar en la Figura (1.2). Este trabajo, se centra en estudiar modelos que explican la propagación de un

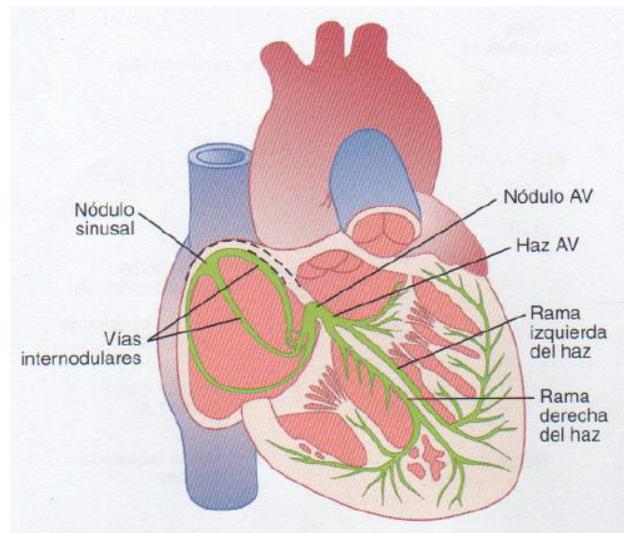


Figura 1.2: Red de propagación de potenciales de acción a través del corazón².

potencial de acción a través de las *redes de Purkinje*.

²Fuente: *Texas Heart Institute*.

1.2. Potencial de acción

El potencial de acción es provocado por reiterados estímulos a través de la membrana celular. Esta pared, separa dos ambientes con diferentes concentraciones iónicas, lo que provoca que haya una diferencia de potencial, denominada como *potencial de membrana*. El voltaje medido en estado de reposo (sin un estímulo exterior), generalmente tiene valores negativos, medidos en un orden de decenas de mV .

La excitación provocada por un estímulo externo, genera una variación abrupta del potencial de membrana, a esta excitación se le denomina *potencial transmembrana*, la cual provoca un proceso de varias fases.

- Fase 0: En esta etapa, ocurre la despolarización rápida gracias al ingreso de sodio a través de los canales de la membrana. Esto provoca un aumento en la diferencia de voltaje, llegando a valores positivos. La despolarización rápida ocurre si la diferencia de voltaje llega a un cierto valor umbral. En caso de no llegar a este valor, la diferencia de voltaje se estabilizará y volverá a su valor de reposo.
- Fase 1: Se llega al *peak* del voltaje, en esta fase se desactivan los canales de sodio y comienza a escapar el potasio y cloro.
- Fase 2: El potencial alcanza una etapa de "meseta", esto se debe que coincide ingreso de calcio y salida de potasio, provocando un "equilibrio" en el potencial. Este paso es más común en modelos cardiacos.
- Fase 3: Es la etapa de *repolarización rápida*, provocada por el cierre de los canales de calcio. También puede ocurrir una hiperrepolarización (llegada a valores por debajo del reposo).
- Fase 4: Consiste en la etapa de reposo.

Las 5 fases se muestran en la Figura (1.3). Para observar la actividad eléctrica cardiaca, se

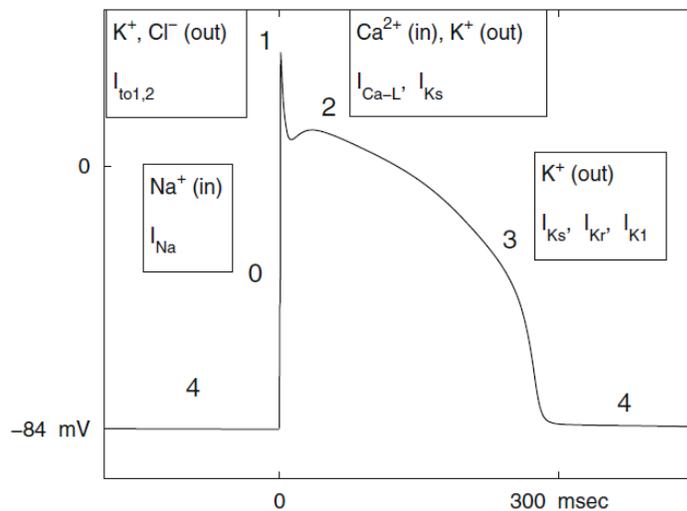


Figura 1.3: Gráfico de ejemplo de potencial de acción sobre una célula con respecto al tiempo, especificando las corrientes asociadas a cada una de las 5 fases, el tiempo está medido en milisegundos. Fuente: *Mathematical Cardiac Electrophysiology* ([10]).

realizan mediciones no invasivas a través de electrodos situados a lo largo de la superficie del cuerpo. Una representación gráfica de la diferencia de voltaje obtenida a partir de los datos entregados por estos electrodos se denomina *Electrocardiograma*. Obtener este tipo de mediciones es de vital importancia, pues, ayuda a facilitar el diagnóstico de enfermedades cardiacas.

1.3. Modelamiento matemático en electrofisiología cardiaca

Para modelar este fenómeno, se tratan dos enfoques de modelación, uno cualitativo, llamado *Fenomenológico* y otro cuantitativo, también denominado como *Biofísico*. Como la membrana celular separa las cargas, acumulando distintas polaridades en la superficie exterior e interior, se puede considerar como un condensador electrolítico. Por otro lado, los canales iónicos, los cuales permiten el intercambio iónico actúan como conductores, los cuales tienen una *resistencia* asociada. La *capacitancia* o capacidad de almacenar energía de

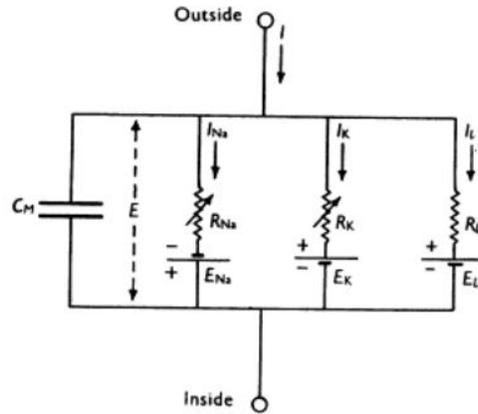


Figura 1.4: Modelo de circuito (Hodgkin & Huxley,[3]), obtenido a partir del análisis de la estructura de la membrana celular.

este condensador (C_m), se puede calcular en función a la diferencia de voltaje entre los dos medios (V) y la carga eléctrica almacenada en la capa exterior de la membrana (Q).

$$C_m = \frac{Q}{V} \quad (1.1)$$

Considerando la capacitancia constante, se obtiene a partir de (1.1) que la corriente capacitiva se obtiene a partir de la variación de la diferencia de voltaje con respecto al tiempo (1.2).

$$C_m \frac{dV}{dt} = \frac{dQ}{dt} = I_{cap} \quad (1.2)$$

La ecuación (1.3) se obtiene gracias a la primera ley de Kirchhoff, donde I_{ion} es la suma de todas las corrientes iónicas que intervienen en el sistema.

$$I_{cap} = C_m \frac{dV}{dt} + I_{ion} \quad (1.3)$$

Cada una de las corrientes iónicas (I_r), depende de la cantidad de canales iónicos activados por unidad de área de la superficie de la membrana (g), la conductancia del canal iónico (g_c), el potencial de reposo del canal (V_r).

$$I_r = g(V, t) g_c(V - V_r) \quad (1.4)$$

A su vez, dada la definición de g , (1.4) se puede escribir como (1.5).

$$I_r = g_c \frac{N_{tot}^r}{S} \omega(V, t) (V - V_r) = G_c \omega(V, t) (V - V_r) \quad (1.5)$$

Donde N_{tot}^r es el número total de canales del ión r , S es el área de la superficie de la membrana celular y $G_c = g_c \frac{N_{tot}^r}{S}$, término que representa la conductancia maximal. Por otro lado, el término $\omega(V, t) = \omega_1^{n_1}(V, t) \cdot \omega_2^{n_2}(V, t) \cdot \dots \cdot \omega_s^{n_s}(V, t)$ posee valores porcentuales y representa el porcentaje de ciertos canales iónicos activados, donde $s, N_i \in \mathbb{N}$ y ω_i son variables llamadas *Gating Variables* (o variables de compuerta). En otras palabras, la conductancia sobre la superficie estudiada es la conductancia maximal multiplicada por un factor de escala, con valores entre 0 y 1, formado por las variables de compuerta, tal como se muestra en (1.6).

$$\bar{g} = g \cdot g_c = G_c \omega(V, t) \quad (1.6)$$

De esta forma, (1.7) es una representación general de una corriente iónica.

$$I_r = \bar{g}_r(V - V_r) \quad (1.7)$$

Los modelos estudiados son de distintas características, ya sea porque son biofísicos o no, o si están especializados en tejido cardíaco o no.

Es importante notar que el tipo de modelos descritos son provenientes de una sola célula. Para sistemas más complejos, como por ejemplo tejidos, hay que considerar la corriente proveniente de de otras células, provocando un fenómeno de difusión. Por lo mismo, se añade el término $D_x(\sigma(x)D_x(V))$, donde x es la posición y σ una función (escalar o tensor) que modela la conductividad del tejido, pues, cuando toma valores cercanos a 0, disminuye la difusión. Como el voltaje depende de la posición y se añaden derivadas espaciales, el sistema de ecuaciones diferenciales ordinarias que se considera en los modelos, pasan a ser sistemas de ecuaciones en derivadas parciales.

1.4. Modelos a estudiar

1.4.1. Hodgkin-Huxley (1952)

El modelo de Hodgkin-Huxley (HH) explica los mecanismos asociados a la propagación de potenciales de acción en un axón de un calamar gigante. El modelo es de tipo *Biofísico*, por lo cual, los parámetros y ecuaciones fueron ajustadas en base a mediciones reales.

En este modelo, I_{ion} corresponde a la suma de las corrientes iónicas asociadas al *potasio* (I_K), *Sodio* (I_{Na}) y una *corriente de fuga* que encapsula el aporte de otros iones, como el cloro, (I_l).

Los potenciales de reposo y los términos G_c de cada corriente iónica son constantes y sus valores fueron calculados empíricamente, al igual que la capacitancia C_m .

A partir de las observaciones de la conductancia de canales iónicos sobre la superficie analizada, en (Hodgkin-Huxley, [3]) se concluye que para obtener un comportamiento similar al crecimiento del potasio al inicio de la despolarización, se necesita un término de orden 4. Por lo cual, usando una *gating variable* asociada a la apertura de canales de potasio (llamada n), que cumpla una dinámica de orden 1, entonces, (1.8) representa el término de corriente asociado al intercambio de iones de potasio.

$$I_K = G_K n^4 (V - V_K) \quad (1.8)$$

Donde n cumple la ecuación (1.9).

$$\frac{dn}{dt} = \alpha_n(V) (1 - n) - \beta_n(V) n \quad (1.9)$$

Se escoge la ecuación (1.9) debido a que n es un porcentaje, por lo cual, n representa un porcentaje de iones en el interior y $(1 - n)$ en el exterior, α_n es la tasa de transferencia desde el exterior hacia el interior y β_n desde el interior hacia el exterior. Estas tasas varían con respecto al voltaje, ya que, la evolución de las corrientes iónicas afectan el voltaje y viceversa. Las funciones $\alpha_n(\cdot)$ y $\beta_n(\cdot)$, también se determinan a partir de datos empíricos.

Por otro lado, siguiendo el mismo esquema, se puede concluir que la apertura de canales de Sodio se comportan como un término de orden 3, sin embargo, el proceso de cierre no es igual, siendo de orden 1. Esto se modela agregando dos *Gating Variables*, m (apertura) y h (cerradura), ambas que cumplen una ecuación de primer orden. Así, el término de corriente iónica está representado en (1.10), mientras que (1.11) y (1.12) son las ecuaciones que cumplen m y h , respectivamente.

$$I_{Na} = G_{Na} m^3 h (V - V_{Na}) \quad (1.10)$$

$$\frac{dm}{dt} = \alpha_m(V) (1 - m) - \beta_m(V) m \quad (1.11)$$

$$\frac{dh}{dt} = \alpha_h(V) (1 - h) - \beta_h(V) h \quad (1.12)$$

Donde α_m , β_m , α_h y β_h cumplen las mismas suposiciones que α_n y β_n .

Finalmente, como el término I_l aporta menos en la dinámica, sus valores son todos obtenidos de forma empírica. siendo un término de orden 1 que depende de V .

$$I_l = \bar{g}_l(V - V_l) \quad (1.13)$$

Denominando I_{app} la corriente aplicada a la región estudiada, se obtiene el sistema de ecuaciones (1.14).

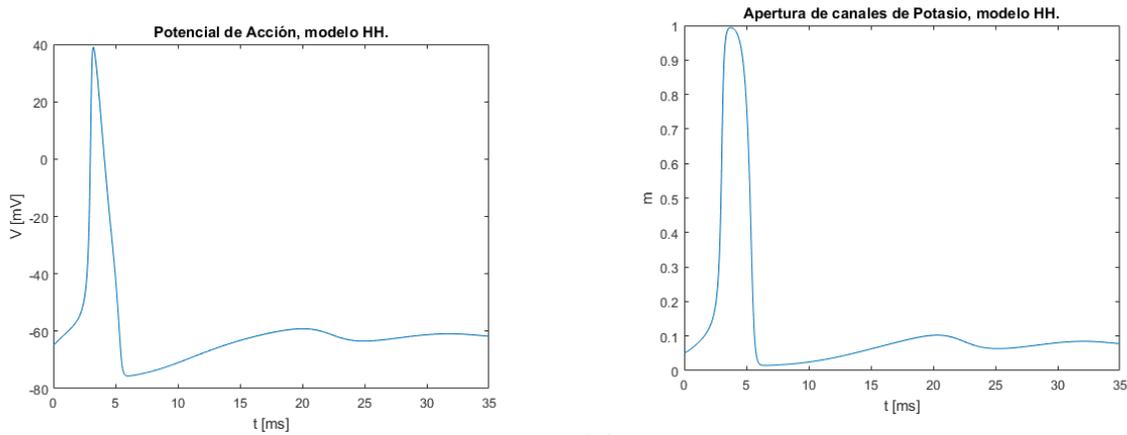
$$I_{app} = C_m \frac{dV}{dt} + G_{Na} m^3 h (V - V_{Na}) + G_K n^4 (V - V_K) + \bar{g}_l (V - V_l) \quad (1.14)$$

$$\frac{dm}{dt} = \alpha_m(V) (1 - m) - \beta_m(V) m \quad (1.15)$$

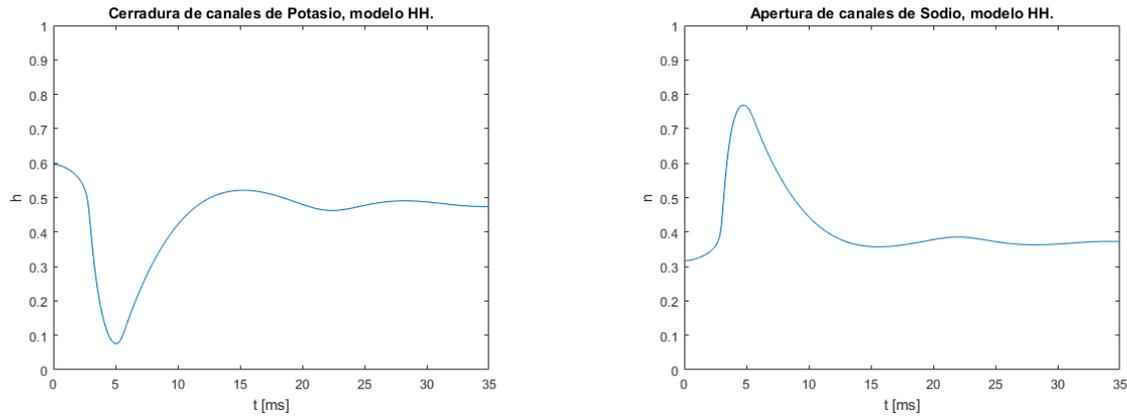
$$\frac{dh}{dt} = \alpha_h(V) (1 - h) - \beta_h(V) h \quad (1.16)$$

$$\frac{dn}{dt} = \alpha_n(V) (1 - n) - \beta_n(V) n \quad (1.17)$$

Los valores de constantes utilizados, como también la descripción de las funciones α y β se pueden ver en las Tablas (4.1) y (4.2), respectivamente, en el *Anexo*.



(a) Voltaje [mV] con respecto al tiempo [ms]. (b) Apertura de canales de Potasio con respecto al tiempo [ms].



(c) Cerradura de canales de Potasio con respecto al tiempo [ms]. (d) Apertura de canales de Sodio con respecto al tiempo [ms].

Figura 1.5: Solución obtenida al simular el modelo de Hodgkin-Huxley en MatLab. Los valores de constantes son las mismas que en la tabla (4.1). Mientras que la corriente aplicada es $I_{app} = 5$.

Como se trata de un modelo obtenido de un axón, se pierde la forma de meseta que aparece en la Fase 2 del Potencial de Acción. Esto se debe a que no es un modelo cardíaco.

1.4.2. Morris-Lecar (1981)

Modelo neuronal que estudia el potencial de acción sobre las fibras musculares de un percebe gigante. En este caso, se trata de un modelo fenomenológico, por lo cual, se centra en obtener la forma de un potencial de acción con bajo costo computacional a cambio de encontrar una forma simplificada de trabajar el intercambio iónico asociado.

El modelo de Morris-Lecar, propuesto en (Morris & Lecar, [9]), considera dos corrientes asociadas a canales iónicos, uno de Calcio y otro de Potasio. Por otro lado, se simplifica el

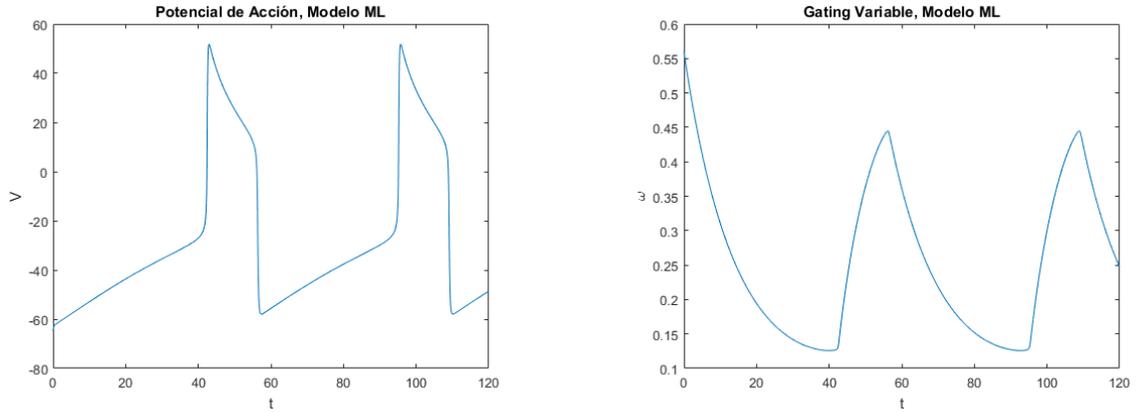
modelo con respecto a HH, ya que se considera la *gating variable* asociada al Calcio como una función dependiente del voltaje (m), pero que se ajusta a su estado estacionario (*steady state*) m_∞ , reduciendo el número de ecuaciones diferenciales en el sistema. Por otro lado, la variable de compuerta asociada al potasio (ω), se considera de orden 1, cumpliendo una ecuación diferencial del mismo orden. De esta manera, el sistema de ecuaciones se simplifica al visto en (1.18) y (1.19).

$$I_{app} = C_m \frac{dV}{dt} + G_{Ca} m_\infty (V - V_{Ca}) + G_K \omega (V - V_K) + \bar{g}_l (V - V_l) \quad (1.18)$$

$$\frac{d\omega}{dt} = \varepsilon \frac{(\omega_\infty(V) - \omega)}{\tau(V)} \quad (1.19)$$

Donde ε corresponde a un ajuste de frecuencias, ω_∞ es el estado estacionario y τ una función dependiente de V .

Los valores de constantes utilizados, como también la descripción de los estados estacionarios y τ se pueden ver en las Tablas (4.3) y (4.4), respectivamente, en el *Anexo*.



(a) Voltaje con respecto al tiempo.

(b) Gating variable, ω , con respecto al tiempo.

Figura 1.6: Solución obtenida al simular el modelo de Morris-Lecar en MatLab. Los valores de constantes son las mismas que en la Tabla (4.3). La corriente aplicada es $I_{app} = 90$.

1.4.3. FitzHugh-Nagumo (1961)

Modelo polinomial, que ajusta la dinámica de un oscilador para obtener una descripción fenomenológica de un sistema excitable (como neuronas o células cardiacas). La primera aproximación (FitzHugh, [1]), consiste en ajustar una ecuación diferencial, en particular la del oscilador de Van der Pol (1.20) para explicar el fenómeno.

$$\frac{d^2x}{dt^2} - c(1 - x^2) \frac{dx}{dt} + x = 0 \quad (1.20)$$

Considerando el cambio de variable $y = \frac{dx}{dt}/c + \frac{x^3}{3} - x$, reemplazando en 1.20, se obtiene (1.21).

$$\frac{dx}{dt} = c \left(x - \frac{x^3}{3} + y \right) \quad (1.21)$$

$$\frac{dy}{dt} = -\frac{x}{c} \quad (1.22)$$

La variable x representa la diferencia de voltaje e y la *restitución*. La restitución es una variable *lenta* que ajusta el voltaje a lo largo del tiempo, si el voltaje es bajo, la restitución crece, si esta crece, el voltaje tiende a aumentar, sin embargo, cuando el voltaje es muy alto, la restitución tiende a decrecer. Esto no se logra con el sistema en (1.21), ya que, la restitución no tiene el efecto retardado que se desea. Para esto, se corrige el la segunda ecuación añadiendo un término lineal que depende de y , obteniendo (1.23).

$$\frac{dV}{dt} = c \left(V - \frac{V^3}{3} + R + I_{app} \right) \quad (1.23)$$

$$\frac{dR}{dt} = -\frac{(V - a - bR)}{c} \quad (1.24)$$

Donde se reemplaza x por V e y por R . I_{app} es la corriente externa aplicada sobre el sistema. Finalmente, para añadir a la dinámica la condición de valor umbral que se requiere para lograr la despolarización, se agrega el término α , que representa el valor del umbral, obteniendo (1.25).

$$\frac{\partial V}{\partial t} = \nu V(V - \alpha)(V_l - V) - R + I_{app} \quad (1.25)$$

$$\frac{\partial R}{\partial t} = \delta(V - a - bR) \quad (1.26)$$

Donde ν es la velocidad de excitación o despolarización ($0 < \nu$), α es el umbral de excitación o despolarización ($0 < \alpha < 1/2$), δ es la tasa de decaimiento del voltaje ($0 < \delta$), d es la tasa de decaimiento de la restitución ($d > 0$) y V_l es un valor dado para cambiar el signo de la derivada y provocar un descenso una vez que el voltaje alcanza ese valor, comúnmente $V_l = 1$.

Esta forma del término cúbico, permite obtener 3 puntos de equilibrio (0 , α y V_l), siendo $V = 0$ el punto de equilibrio estable, pasando a ser este valor el estable, a diferencia de los modelos Biofísicos, donde este es negativo.

Gracias a la forma que tiene la ecuación, no sólo R funciona como retroalimentador de V , sino también el término cúbico $V(V - \alpha)(V_l - V)$. Si el voltaje, luego de una excitación previa, toma valores menores a α (umbral de excitación), el término se tornará negativo ($V(t, x)$ será positivo, $(\alpha - V(t, x))$ positivo y $(V(t, x) - 1)$ negativo), esto se traduce en que la dinámica tiende a tomar valores negativos, llegando a un comportamiento esperable desde el punto de vista teórico, si V no supera el umbral α luego de una excitación previa, no ocurrirá el proceso de despolarización (Fase 0), esto se debe a que el término cúbico será negativo. Por otro lado, si el valor de V es muy alto, en este caso se modela como $V \gg V_l$, el término cúbico nuevamente tomará valores negativos (V será positivo, $(\alpha - V)$ negativo y $(V - V_l)$ positivo), luego, esto ayuda a lograr la transición de la fase 0 a las siguientes. En el caso de hiperpolarización (en este caso, que V sea negativo), el término no lineal, otra vez, funciona

como "retroalimentador", siendo positivo en ese caso ($V(t, x)$ será negativo, $(\alpha - V(t, x))$ positivo y $(V(t, x) - 1)$ negativo).

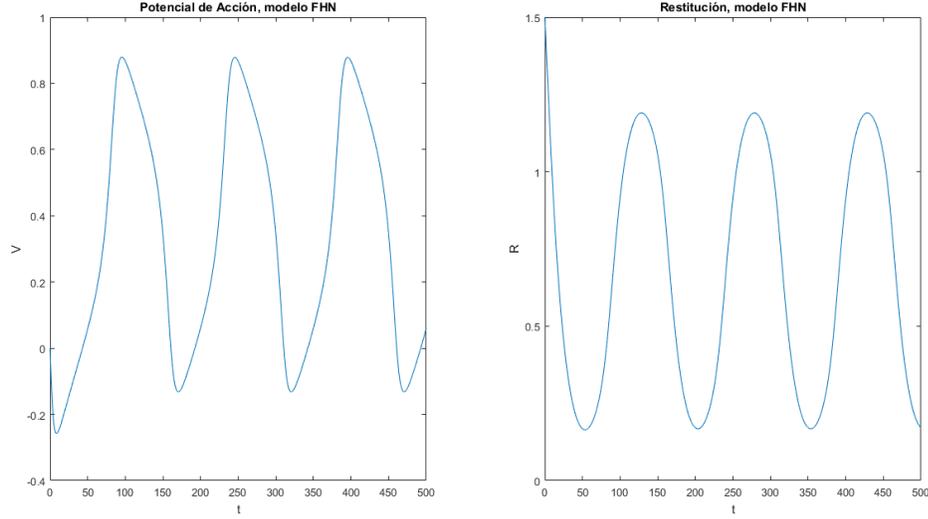


Figura 1.7: Solución obtenida al simular el modelo de FitzHugh-Nagumo en MatLab. Los valores de constantes utilizadas se encuentran en la Tabla (4.5), en el *Anexo*. La corriente aplicada es $I_{app} = 0,02$.

1.4.4. ten Tusscher - Panfilov (2006)

Modelo propuesto en (ten Tusscher et al, [14]), que consiste en una mejora a un modelo propuesto el año 2004. El modelo de ten Tusscher - Panfilov es de tipo biofísico y modela el Potencial de Acción sobre tejido ventricular, teniendo variaciones para Miocardio (medio), Endocardio y Epicardio. Considera las siguientes variables:

- Dinámica de Calcio Ca_i : Depende de 3 variables (Ca_{sr} , Ca_{ss} y R_{prime}) y una corriente total de calcio (I_{tCa}), la cual, a su vez, está conformada por distintos tipos de corrientes asociadas a este ión. Una es la corriente de calcio tipo-L (I_{CaL}), producida por el intercambio de iones de calcio a través de canales especializados (*Canal DHP*), su activación es más larga. En su modelo participan 4 *gating variables*, las cuales se denominan como \bar{d} , f , $f2$ y $fCass$. Otras dos, pero que no cuentan con variables de compuerta son la corriente de fondo o *Background Current* de Calcio (I_{CaB}) y la corriente de bombeo de calcio (I_{pCa}). Además, también hay bombas encargadas de la dinámica de intercambio de distintos iones, uno que afecta el calcio es la bomba *Sodio-Calcio* (I_{NaCa}), que también depende de Ca_i . Obteniendo así, $I_{tCa} = I_{CaL} + I_{CaB} + I_{pCa} - 2I_{NaCa}$.
- Dinámica de Sodio Na_i : Obtenida a partir de la corriente total de Sodio (I_{tNa}), la cual depende de otras corrientes. La primera es la corriente rápida de sodio (I_{Na}), que está sujeta a 3 variables de compuerta m , h y j . Otra es la corriente, pero que no cuenta con estas variables, es la de fondo de Sodio (I_{NaB}). Sumando lo aportado por las bombas de *Sodio-Potasio* (I_{NaK}) y *Sodio-Calcio* (I_{NaCa}), se obtiene $I_{tNa} = I_{Na} + I_{NaB} + 3I_{NaK} + 3I_{NaCa}$.

- Dinámica de Potasio K_i : Dependiente de la corriente externa y la corriente total de Potasio (I_{tK}). Esta última corriente depende de la corriente rápida de potasio (I_{Kr}), proveniente de dos variables de compuerta: $xr1$ y $xr2$; corriente lenta de potasio (I_{Ks}), con una variable de compuerta xs ; corriente de rectificación interna de potasio (I_{K1}) y la corriente de bombeo de potasio (I_{pK}). La corriente externa transitoria (I_{to}), la más importante en durante la Fase 1, con dos variables de compuerta r y s . Finalmente, agregando lo aportado por la bomba *sodio-potasio*, se obtiene $I_{tK} = I_{Kr} + I_{Ks} + I_{K1} + I_{pK} + I_{to} - 2I_{NaK}$.

En resumen, es un sistema de ecuaciones (1.27) de 19 variables (4 asociadas a corrientes y 15 *gating variables*). Valores de constantes y variables se encuentran en el Anexo.

$$\frac{dV}{dt} = -(I_{tK} + I_{tCa} + I_{tNa} + I_{app}) \quad (1.27)$$

$$\frac{dNa_i}{dt} = \frac{-I_{tNa}}{(V_c F)} C_m \quad (1.28)$$

$$\frac{dK_i}{dt} = \frac{-(I_{tK} + I_{app})}{(V_c F)} C_m \quad (1.29)$$

$$\frac{dCa_i}{dt} = Ca_i^{bufc} ((i_{leak} - i_{up}) V_{sr} / V_c + i_{xfer} - (I_{tCa} - I_{CaL}) C_m / (2V_c F)) \quad (1.30)$$

$$\frac{d\bar{d}}{dt} = \frac{(\bar{d}_\infty - \bar{d})}{\tau_d} \quad (1.31)$$

$$\frac{df}{dt} = \frac{(f_\infty - f)}{\tau_f} \quad (1.32)$$

$$\frac{df_2}{dt} = \frac{(f_{2\infty} - f_2)}{\tau_{f2}} \quad (1.33)$$

$$\frac{dfCass}{dt} = \frac{(fCass_\infty - fCass)}{\tau_{fCass}} \quad (1.34)$$

$$\frac{dh}{dt} = \frac{(h_\infty - h)}{\tau_h} \quad (1.35)$$

$$\frac{dj}{dt} = \frac{(j_\infty - j)}{\tau_j} \quad (1.36)$$

$$\frac{dm}{dt} = \frac{(m_\infty - m)}{\tau_m} \quad (1.37)$$

$$\frac{dx_{r1}}{dt} = \frac{(x_{r1\infty} - x_{r1})}{\tau_{x_{r1}}} \quad (1.38)$$

$$\frac{dx_{r2}}{dt} = \frac{(x_{r2\infty} - x_{r2})}{\tau_{x_{r2}}} \quad (1.39)$$

$$\frac{dx_s}{dt} = \frac{(x_{s\infty} - x_s)}{\tau_{x_s}} \quad (1.40)$$

$$\frac{dr}{dt} = \frac{(r_\infty - r)}{\tau_r} \quad (1.41)$$

$$\frac{ds}{dt} = \frac{(s_\infty - s)}{\tau_s} \quad (1.42)$$

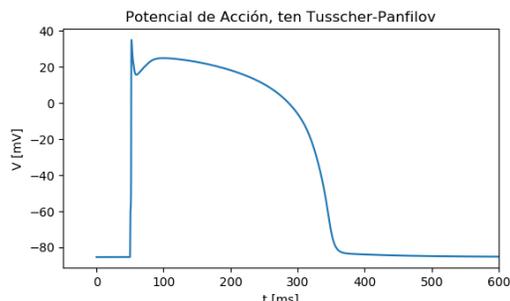
$$\frac{dCa_{ss}}{dt} = Ca_{ss}^{bufss} (-I_{CaL} C_m / (2V_{ss} F) + i_{rel} V_{sr} / V_{ss} - i_{xfer} V_c / V_{ss}) \quad (1.43)$$

$$\frac{dCa_{sr}}{dt} = Ca_{sr}^{bufsr} (i_{up} - (i_{rel} + i_{leak})) \quad (1.44)$$

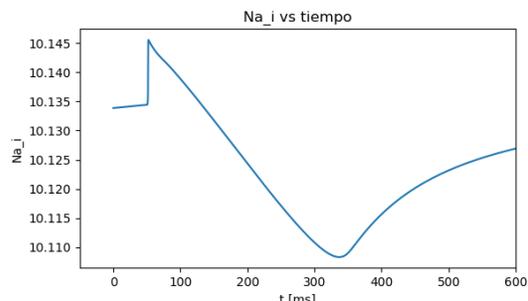
$$\frac{dR_{prime}}{dt} = -k_2 Ca_{ss} R_{prime} + k_4 (1 - R_{prime}) \quad (1.45)$$

$$(1.46)$$

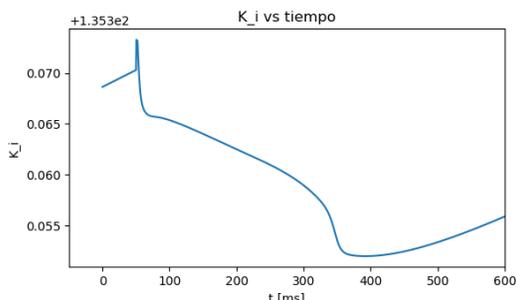
Resultados numéricos del sistema se muestran en la Figura (1.8). Para la obtención de estos, se usó *MyoKit*, paquete diseñado para simplificar el uso de modelos biofísicos asociados al funcionamiento de los miocitos. El lenguaje de programación de *MyoKit* está basado en *Python*. Más detalles sobre *MyoKit* se encuentran en la sección de *Software Utilizado* en *Anexo*.



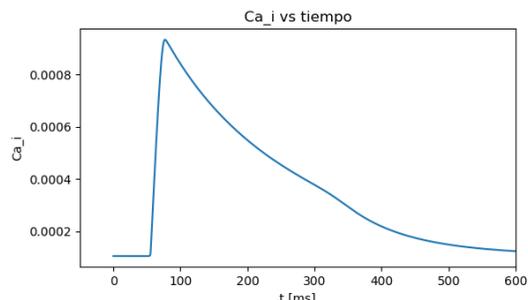
(a) Voltaje [mV] con respecto al tiempo en [ms].



(b) Na_i con respecto al tiempo en [ms].



(c) K_i con respecto al tiempo en [ms].



(d) Ca_i con respecto al tiempo en [ms].

Figura 1.8: Voltaje y corrientes asociadas a las dinámicas de iones de Sodio, Potasio y Calcio, sobre una célula del *epicardio*. Resultados obtenidos al ejecutar simulaciones del modelo de ten Tusscher-Panfilov en *MyoKit*. Corriente aplicada $I_{app} = -52 [A/F]$ en el tiempo $t = 50 [ms]$.

1.5. Costo computacional y característica de los modelos estudiados

A partir de lo visto anteriormente, la Tabla (1.1) resume algunas características que distinguen a los modelos a trabajar. Por otro lado, la Tabla (1.2) indica las características de cada tipo de modelo.

Modelo presentado	Tipo de Modelo	Tipo de Célula	Cantidad de variables
Hodgkin-Huxley	Biofísico	Neuronal	4
Morris-Lecar	Fenomenológico	Muscular	2
FitzHugh-Nagumo	Fenomenológico	Cualquier sistema excitable	2
ten Tusscher-Panfilov	Biofísico	Cardiaca	19

Tabla 1.1: Resumen de las características de los modelos estudiados.

Característica \ Tipo de modelo	Fenomenológico	Biofísico
Cuantitativo	No	Sí
VARIABLES complementarias predominantes	Restitución	Compuerta
Tiempo de cómputo	Usualmente menor	Mayor

Tabla 1.2: Características generales de los modelos de tipo Fenomenológico y Biofísico.

Una explicación más precisa, suele requerir un mayor número de variables, por lo tanto, también de ecuaciones. Si bien, todos los modelos vistos permiten explicar la evolución del potencial de acción en una célula, los de tipo biofísico tratan con otros mecanismos que participan en la generación de las diferencias de voltaje a lo largo de la membrana. Conocer más a fondo el mecanismo de intercambio iónico permite entender situaciones algunas anómalas como, por ejemplo, la acción de ciertas drogas que puedan afectar el comportamiento de uno de estos procesos, lo cual provoca comportamientos distintos en el potencial de acción. Sin embargo, a medida que más información se requiere, mayor es el trabajo computacional que se realiza.

La búsqueda de un *modelo equivalente*, permitiría considerar un modelo fenomenológico y ajustar sus variables a otro modelo ya conocido, independiente de su naturaleza, de esta forma, se podría reconstruir un resultado proveniente de un sistema estudiado usando uno completamente nuevo, permitiendo rastrear el efecto provocado por una anomalía en algún modelo, que tenga un mayor grado explicativo, en las constantes que conforman un modelo equivalente, el cual, se busca que sea simple. Por otro lado, la *reducción de variables*, busca explicar el comportamiento de un modelo con una menor cantidad de ecuaciones con el menor error posible. Esto se logra a través de un operador que proyecta los datos sobre un subespacio de menor dimensión, de esta manera, cualquier variación en el sistema original, se puede rastrear en el sistema reducido al evaluar en la proyección.

De esta forma, el objetivo de este trabajo es estudiar los métodos de equivalencia de modelos a través de un método de ajuste de nulclinas, tal que, permita considerar un modelo equivalente a uno de los estudiados; y un método de reducción de variables, que consiste en la obtención de un operador de proyección usando redes neuronales.

Capítulo 2

Equivalencia de modelos a través de métodos de ajuste de nulclinas

Consideremos el sistema de n_f ecuaciones (2.1), (2.2).

$$\dot{V} = f_v(V, \omega; \mu) \quad (2.1)$$

$$\dot{\omega} = f_\omega(V, \omega; \mu) \quad (2.2)$$

Donde $V \in \mathbb{R}$ representa, en el caso de una dinámica de potencial de acción, la diferencia de voltaje. $\omega \in \mathbb{R}^{n_f-1}$ representa las $(n_f - 1)$ *gating variables* del sistema y μ son los parámetros del modelo, $\mu \in \mathcal{D} \subset \mathcal{R}^d$.

De forma general, también se puede describir tal como se muestra en (2.3).

$$\dot{x} = f(x; \mu) \quad (2.3)$$

Donde $x = (V, \omega)^t \in \mathbb{R}^{n_f}$.

2.1. Ajuste de funciones de densidad de energía

El primer enfoque consiste en encontrar modelos que posean una equivalencia con otros modelos, sin la necesidad de tener un trasfondo físico. Una forma sería observar como funciona el sistema desde un punto de vista energético y lograr reproducir ese comportamiento con otras variables cuyo cálculo de sus valores demanda un menor costo, de esta manera, al considerar un sistema que conforman esas variables, se estaría determinando un modelo fenomenológico equivalente. Primero se analizó el término de *densidad de potencial electroquímico* (Hurtado & Henao, ([4])), función multivariada a valores reales cuya derivación permite obtener los términos que constituyen la ecuación original. En el caso de *FitzHugh-Nagumo* sobre una célula, la densidad de potencial electroquímico es la función en (2.4), la cual, al derivar con respecto a V y R se obtiene el sistema conformado por (2.5) y (2.6).

$$E(V, R) = \nu \left(\frac{1}{4}V^4 - \frac{(V_l + \alpha)}{3}V^3 + \frac{\alpha V_l}{2}V^2 \right) + \delta \left((V - a)R - \frac{b}{2}R^2 \right) \quad (2.4)$$

$$\frac{\partial V}{\partial t} = -\frac{\partial E}{\partial V} = \nu V(V - \alpha)(V_l - V) - \delta R \quad (2.5)$$

$$\frac{\partial R}{\partial t} = \frac{\partial E}{\partial R} = \delta(V - a - bR) \quad (2.6)$$

$$(2.7)$$

A partir del sistema conformado por (2.5) y (2.6), se concluye que el sentido de crecimiento de energía con respecto al voltaje es contrario al crecimiento del voltaje con respecto al tiempo, caso distinto al de la restitución, en el cual los sentidos son iguales. Esto provoca una dinámica en la cual se busca disminuir E en la dirección dada por el voltaje, pero en el caso de la restitución, se busca maximizar E . Las Figuras (2.1), (2.2) y (2.3) muestra el comportamiento de la función E con respecto a distintos valores de voltaje y restitución. Por otro lado, la Figura (2.4), muestra la función sobre el plano (V, R) .

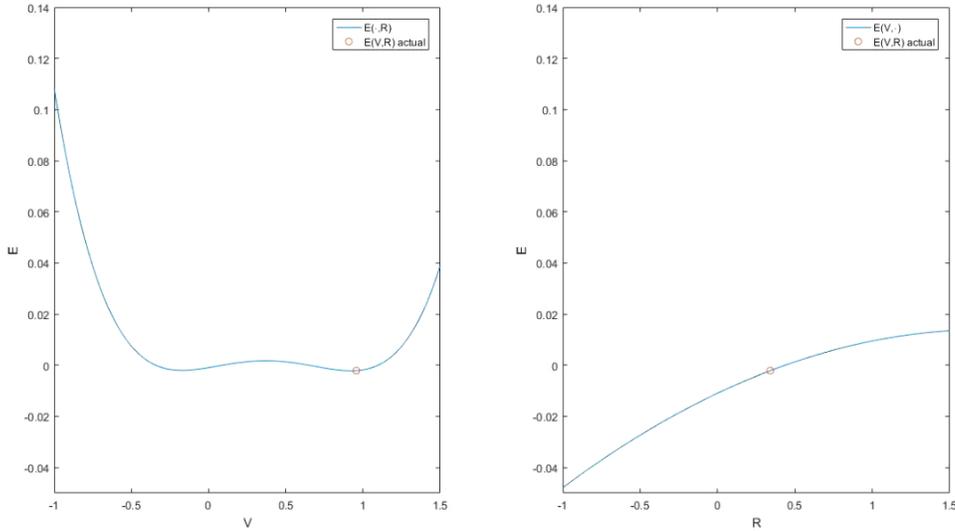


Figura 2.1: Valores de E con respecto a V y R . Caso en el cual se está cerca de un mínimo local con respecto a V , pero con respecto a R aún no se encuentra en un punto estable, provocando lo que se muestra en la Figura (2.2).

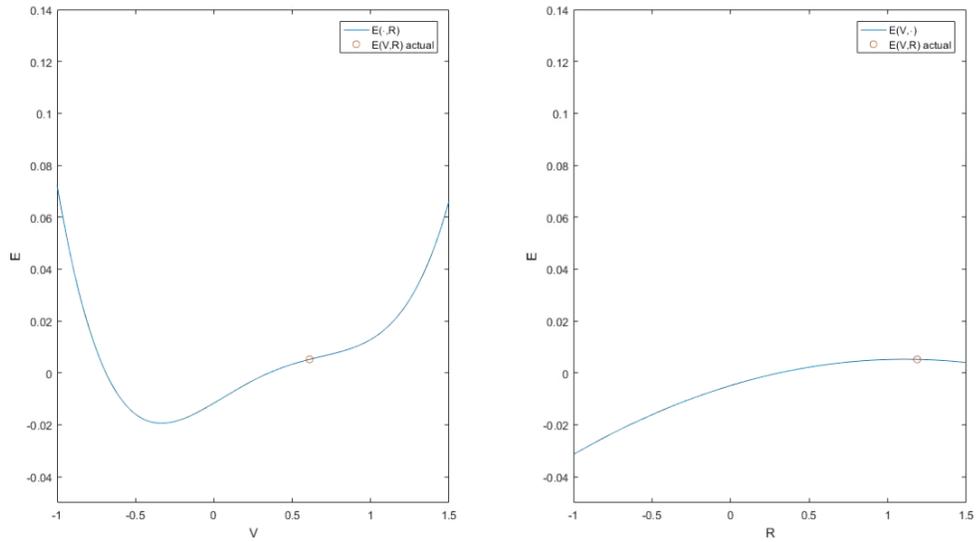


Figura 2.2: Valores de E con respecto a V y R . Caso en el cual se está cerca de un máximo local con respecto a R , pero con respecto a V aún no se encuentra en un punto estable, provocando lo que se muestra en la Figura (2.3).

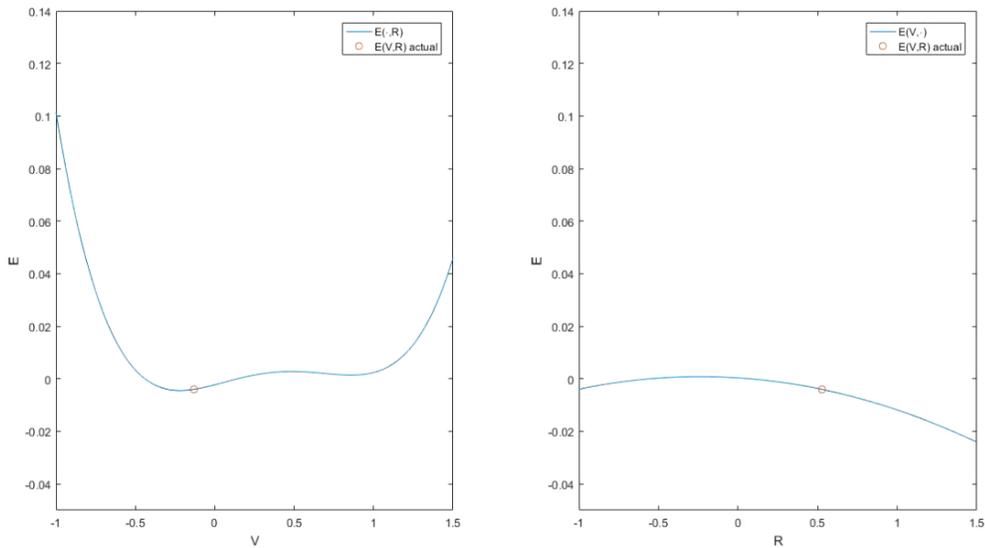


Figura 2.3: Valores de E con respecto a V y R . Caso en el cual vuelve a estar cerca de un mínimo local con respecto a V , pero vuelve a alejarse de un punto estable con respecto a R , provocando la continuación del ciclo.

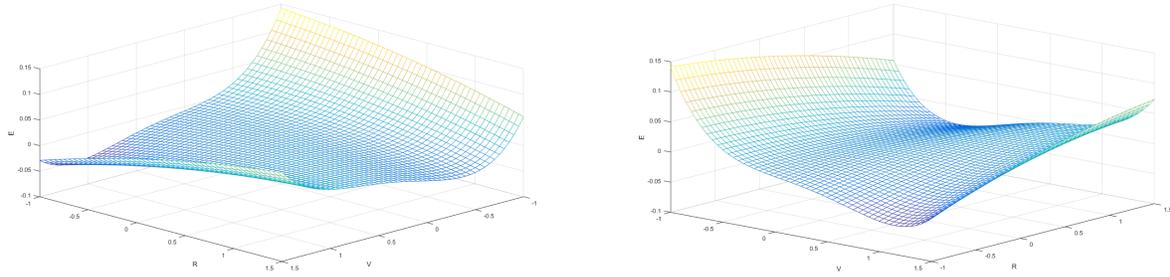


Figura 2.4: Superficie que describe E con respecto al plano (V, R) .

Para lograr el ajuste a modelos más complejos, se debe encontrar una función que dependa de las variables del sistema original y permita explicar la energía asociada al sistema a menor costo. Esto permitiría recuperar un sistema de ecuaciones que obtenga valores similares al original. Esto motiva a estudiar el lado derecho de la ecuación, ya que, este proviene de la función de densidad. Para ajustar el lado derecho de las ecuaciones, primero es necesario explicar cómo este afecta a la dinámica a través del estudio del diagrama de fases.

2.2. Método de ajuste de nulclinas

El diagrama de fases es una descripción de las trayectorias sobre un plano conformado por las variables del sistema. En el caso de *FitzHugh-Nagumo*, por ejemplo, nos permite saber qué valores tomará el par (V, R) luego de una condición inicial. Para lograr que dos diagramas de fase sean similares es necesario lograr que los puntos sean redirigidos en la misma dirección, sentido y velocidad.

Una nulclina es la representación de los puntos en los cuales la derivada temporal de una de las variables es nula, en otras palabras, cuando el lado derecho se anula, debido a esto, cuando una trayectoria atraviesa una de estas curvas, lo hace de forma totalmente vertical u horizontal dependiendo si la variable de la cual proviene la nulclina es representada en la abscisa o la ordenada, respectivamente. Debido a esta definición, las intersecciones de nulclinas son de gran importancia, ya que, representan los puntos de equilibrio del sistema en un diagrama de fases y determinan cuando ocurre un cambio de dirección en el diagrama de fases.

Si las nulclinas asociadas a dos sistemas distintos son iguales, no necesariamente son sistemas parecidos, el ajuste de nulclinas sólo permite obtener información sobre el cambio de direcciones, pero no del sentido de la trayectoria ni de su velocidad. Por lo mismo, es importante determinar qué ocurre en las regiones que delimitan las nulclinas.

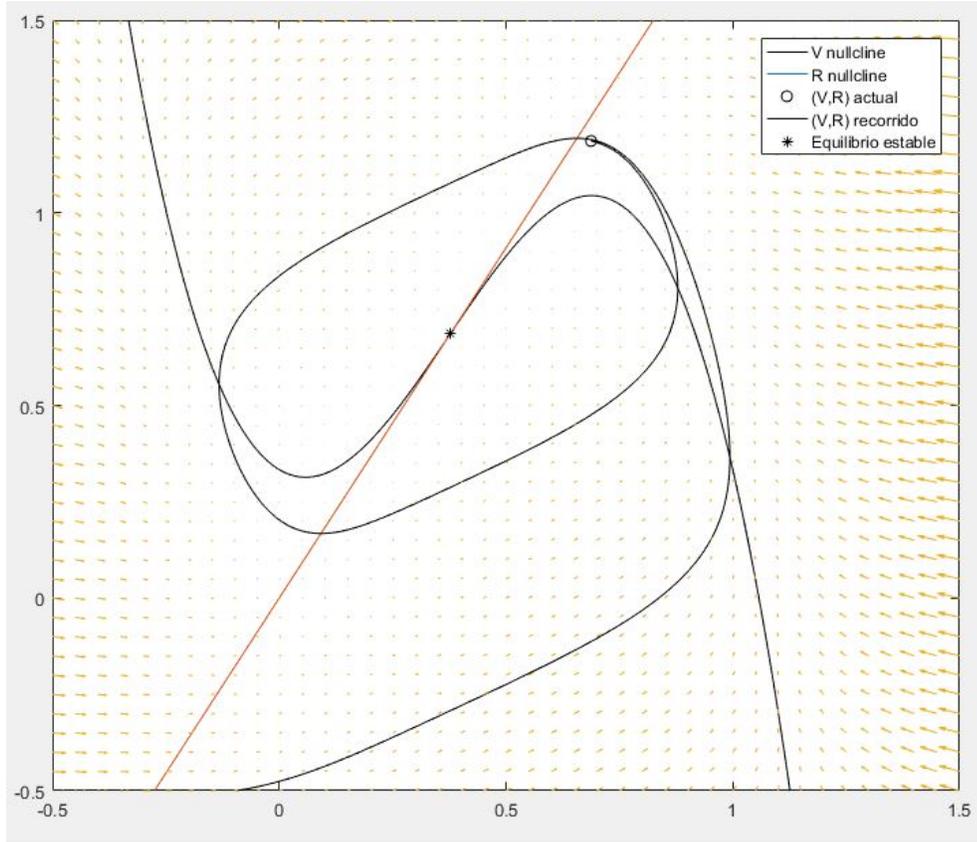


Figura 2.5: Nulclinas sobre un diagrama de fases de modelo de FitzHugh-Nagumo.

El caso analizado corresponde a tomar un sistema de ecuaciones tipo HH, en este caso *Morris-Lecar* y lograr representarlo mediante un sistema equivalente, el cual se considerará de términos polinomiales. En caso de no lograr una aproximación polinomial como tal, podría ser polinomial por pedazos. Para el ajuste de velocidades, se introduce un factor de ajuste, el cual, dependerá del valor de las variables del sistema. El orden en el cual se efectúa el método es el siguiente:

1. Determinar explícitamente las nulclinas del sistema original.
2. Realizar un ajuste polinomial para las nulclinas.
3. Definir un nuevo sistema a partir de las nulclinas polinomiales.
4. Ajustar velocidades sobre las regiones delimitadas por nulclinas multiplicando por un factor no constante.

A partir del sistema de ecuaciones conformado por (2.1) y (2.2), al igualar $f_v(V, \omega) = 0$ y $f_\omega(V, \omega) = 0$, se obtienen las nulclinas al despejar una variable con respecto a la otra, en este caso, escogemos ω , definiendo las nulclinas \mathcal{N}_v y \mathcal{N}_ω , respectivamente, donde

$$\mathcal{N}_v : \omega = g_v(V) \quad (2.8)$$

$$\mathcal{N}_\omega : \omega = g_\omega(V) \quad (2.9)$$

De esta forma, concluye el paso 1.

Para el paso 2, se realiza un ajuste polinomial, esto consiste en encontrar una aproximación

polinomial, $p_v(V)$ y $p_\omega(V)$, de $g_v(V)$ y $g_\omega(V)$, respectivamente. Obteniendo las nulclinas aproximadas

$$\mathcal{N}_v^p : \omega = p_v(V) \quad (2.10)$$

$$\mathcal{N}_\omega^p : \omega = p_\omega(V) \quad (2.11)$$

Luego, el tercer paso, consiste en retomar la forma de un sistema de ecuaciones. Dejando un lado de la igualdad en 0 en las nulclinas obtenidas en (2.10) y (2.11), en otras palabras, restando en ambos lados ω . obtenemos el sistema:

$$f_v^p(V, \omega) = p_v(V) - \omega = 0 \quad (2.12)$$

$$f_\omega^p(V, \omega) = p_\omega(V) - \omega = 0 \quad (2.13)$$

De esta manera, un sistema de la forma:

$$\dot{V} = \varepsilon_v(V, \omega) \cdot f_v^p(V, \omega) \quad (2.14)$$

$$\dot{\omega} = \varepsilon_\omega(V, \omega) \cdot f_\omega^p(V, \omega) \quad (2.15)$$

Posee las nulclinas calculadas en (2.10) y (2.11). Las funciones $\varepsilon_v(V, \omega)$ y $\varepsilon_\omega(V, \omega)$, permiten ajustar la velocidad en los sectores no analizados previamente (regiones delimitadas por nulclinas). El último paso corresponde a determinar estas funciones.

2.3. Resultados de equivalencia a partir del método de ajuste de nulclinas

El ajuste polinomial se realiza sobre el modelo de *Morris-Lecar* (1.18). En trabajos previos, como Postnikov [11], se logra un ajuste de un modelo como *FitzHugh-Nagumo* en un modelo reducido de HH llamado *Rinzel*. Sin embargo, el ajuste es un escalamiento de los valores, lo cual no ajusta completamente la forma de la curva. En la Figura (2.6) se observa este tipo de ajuste sobre el modelo de *Morris Lecar*, obteniéndose una curva a una escala similar, sin embargo, tanto la Fase 0 como la Fase 3, no ocurren a la misma velocidad, sumado a un retraso de la despolarización en *FitzHugh-Nagumo* que aumenta con el tiempo.

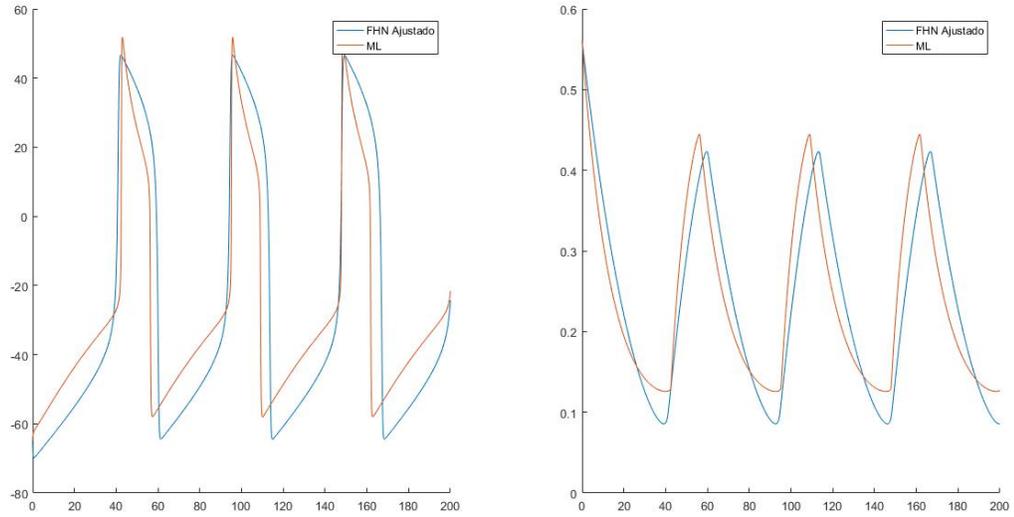
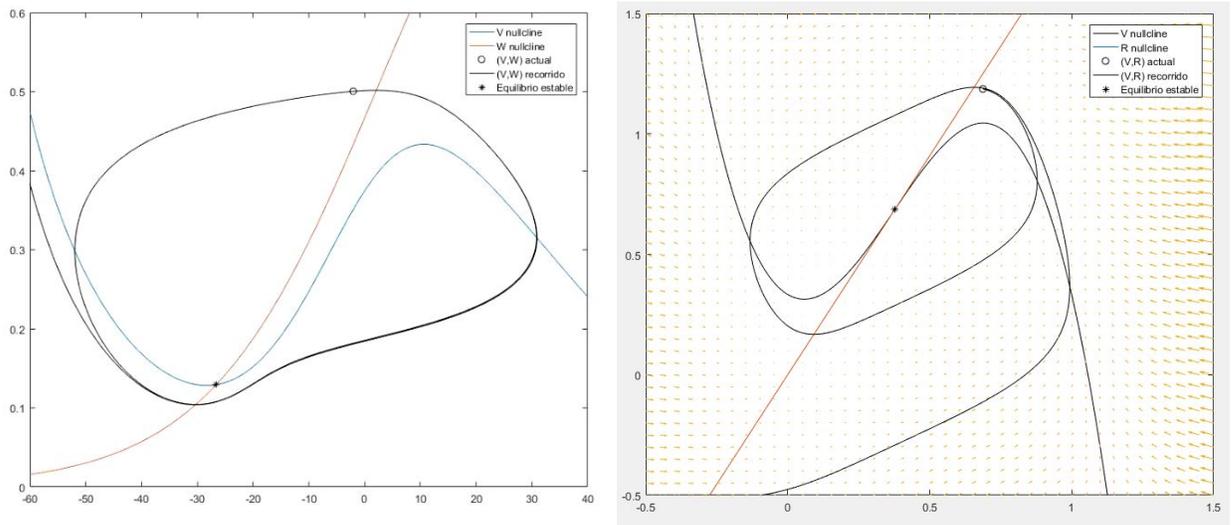


Figura 2.6: Ajuste de parámetros de FitzHugh-Nagumo sobre modelo de Morris-Lecar.

Por lo mismo, el paso a seguir es realizar un ajuste, tomando un modelo polinomial de forma general, de *Morris-Lecar*, pero igualando los diagramas de Fase de ambos modelos. En la Figura (2.7a) Y (2.7b), se observan las nulclinas de los modelos de *Morris-Lecar* y *FitzHugh-Nagumo*, respectivamente. A partir de esto se comprueba que el comportamiento de orden 3 de la nulclina de la dinámica del Voltaje en *FitzHugh-Nagumo*, se asemeja a la del caso de *Morris-Lecar*. Sin embargo, la nulclina asociada a la Restitución en FHN es lineal, caso contrario al visto en ML, el cual presenta una sección convexa y luego pasa a ser cóncava. Para obtener de forma analítica las nulclinas de ML, se iguala a 0 las ecuaciones del sistema (1.18), obteniéndose (2.16) y (2.17).



(a) Caso Morris Lecar.

(b) Caso Fitzhugh Nagumo.

Figura 2.7: Nulclinas asociadas a cada modelo.

$$\omega(V) = \frac{I_{app} - G_{Ca}m_{\infty}(V)(V - V_{Ca}) - \bar{g}_l(V - V_l)}{G_K(V - V_K)} \quad (2.16)$$

$$\omega(V) = \omega_{\infty}(V) \quad (2.17)$$

Debido al comportamiento de ambas nulclinas, se realizó un ajuste polinomial de orden 3, vía Matlab, que permite obtener las candidatas a nulclinas polinomiales (2.18) y (2.19).

$$p_V(V) = p_0(V) + p_1(V)V + p_2(V)V^2 + p_3(V)V^3 \quad (2.18)$$

$$p_{\omega}(V) = q_0(V) + q_1(V)V + q_2(V)V^2 + q_3(V)V^3 \quad (2.19)$$

Donde:

$$p_0(V) \approx 0,3212$$

$$p_1(V) \approx 0,0079$$

$$p_2(V) \approx -6,4129e - 05$$

$$p_3(V) \approx -4,1120e - 06$$

$$q_0(V) \approx 0,4778$$

$$q_1(V) \approx 0,0138$$

$$q_2(V) \approx 4,1360e - 06$$

$$q_3(V) \approx -1,5663e - 06$$

De esta forma, se ajustan las nulclinas, la Figura (2.8) muestra una comparación entre ambas curvas, tanto para la nulclina asociada a V como ω .

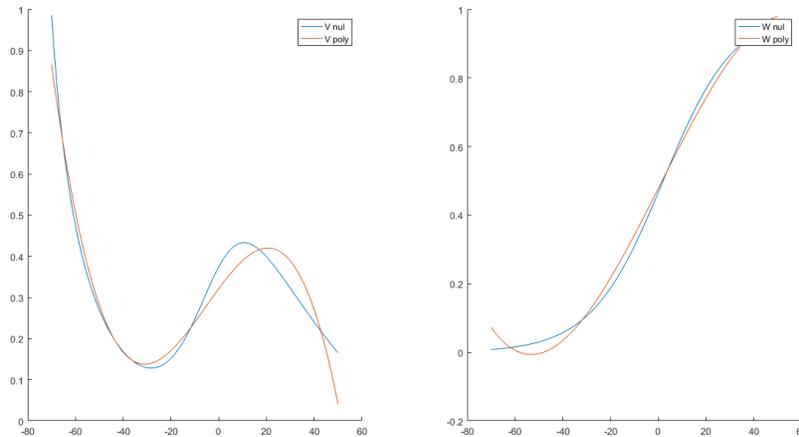


Figura 2.8: Ajuste polinomial de nulclinas de *Morris-Lecar*. Izquierda: Ajuste de nulclina asociada a V . Derecha: Ajuste de nulclina asociada a ω .

Tal como se explica en el Capítulo 2, el ajuste del diagrama de fases no consiste sólo en el de nulclinas, también son importantes las velocidades manejadas en el resto del plano. Para esto, se agrega un término que ajusta estas velocidades en la nueva ecuación. De esta forma,

(2.20) y (2.21) conforman un sistema de ecuaciones que describe el modelo aproximado, donde ε_V y ε_ω son los ajustes de velocidad.

$$\frac{dV}{dt} = \varepsilon_V(V, \omega)(p_V(V) - \omega) \quad (2.20)$$

$$\frac{d\omega}{dt} = \varepsilon_\omega(V, \omega)(p_\omega(V) - \omega) \quad (2.21)$$

Para facilitar el cálculo, las funciones ε_V y ε_ω se consideran constantes por pedazos, en este caso, constantes por regiones, las cuales están delimitadas por las nulclinas. Obteniendo, así, 4 valores distintos para cada función. La Figura (2.9) detalla las regiones en cuestión.

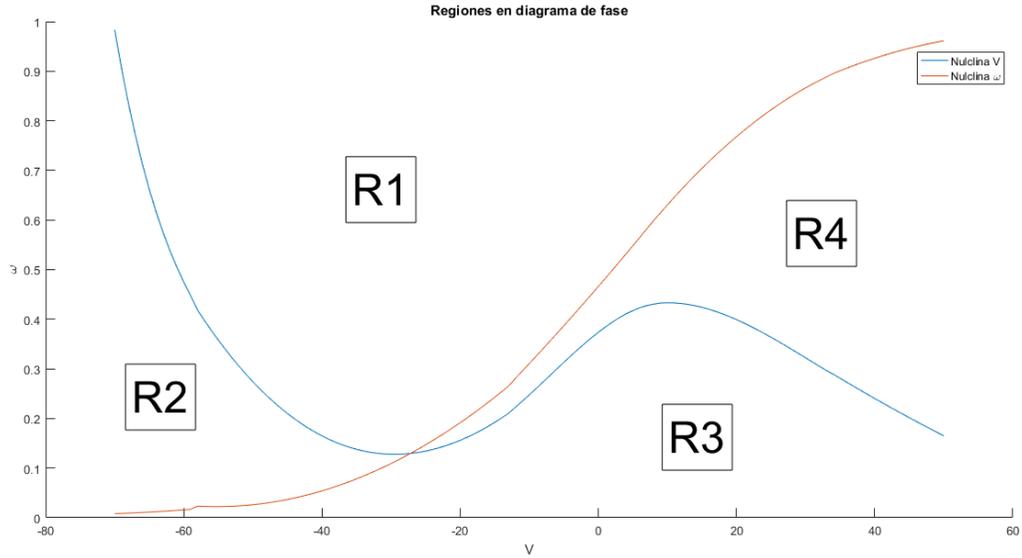


Figura 2.9: Regiones utilizadas para cálculo de funciones de ajuste de velocidades.

La obtención de los valores de cada función se realiza con un método de Newton de primer orden. La Tabla (2.1) muestra los valores obtenidos.

Función	Región	Valor (Aproximado)
ε_V	R1	292.1
ε_V	R2	80.3
ε_V	R3	200.7
ε_V	R4	1015.2
ε_ω	R1	0.0500
ε_ω	R2	0.0435
ε_ω	R3	0.0483
ε_ω	R4	0.0950

Tabla 2.1: Valores aproximados de las funciones de ajuste en cada región.

De esta manera, se completa el sistema de ecuaciones, las Figura (2.10) muestra un resultado comparativo en las soluciones. Por otro lado, la Figura (2.11) muestra las nulclinas en

el plano de fase, añadiendo la trayectoria de las mismas simulaciones.

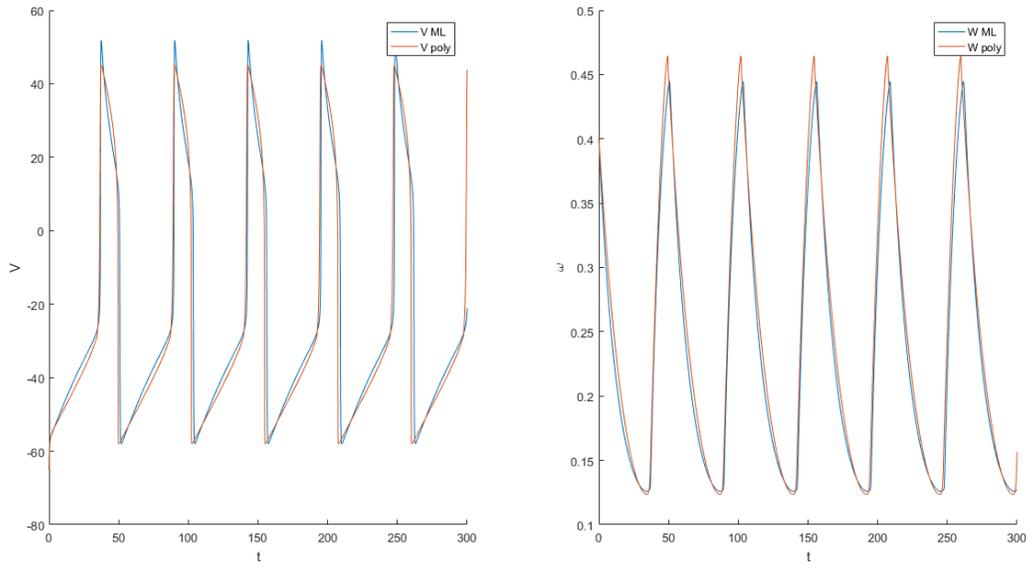


Figura 2.10: Comportamiento de Voltaje (izquierda) y variable de compuerta ω (derecha) para el modelo polinomial (izquierda) y el de *Morris-Lecar* (derecha). Condiciones iniciales: $V_0 = -65$, $\omega_0 = 0,4$.

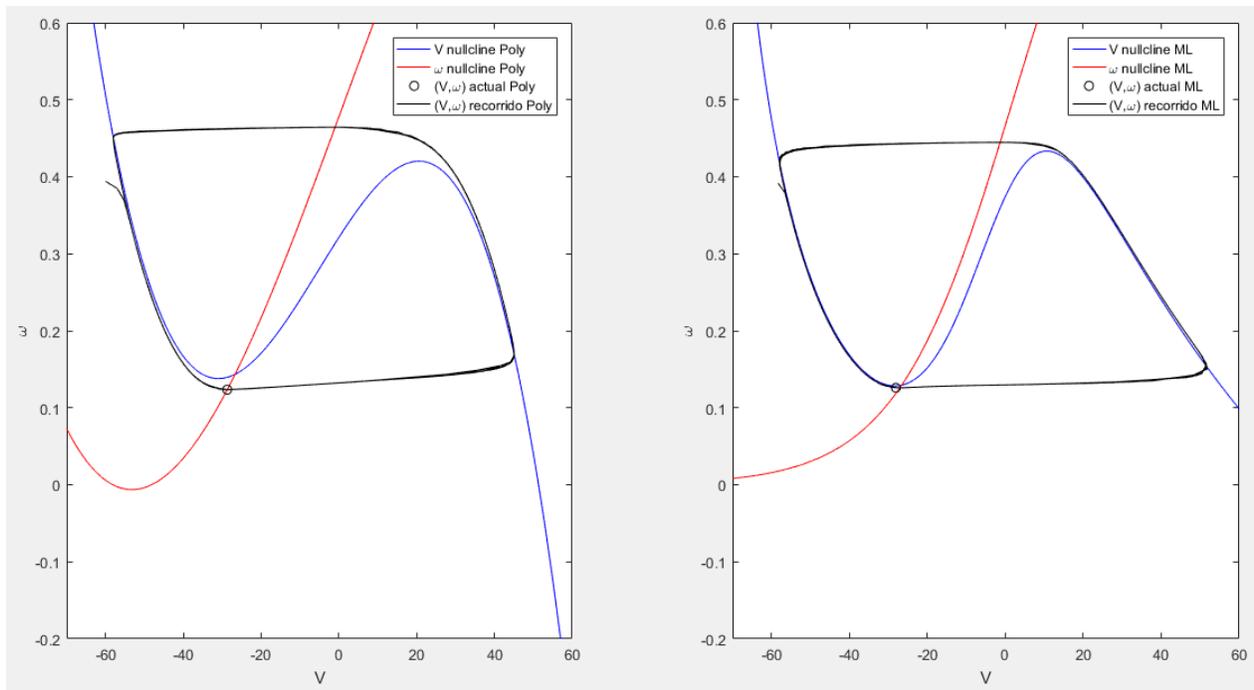


Figura 2.11: Trayectoria de simulación sobre plano de fases para modelos de *Morris-Lecar* y Polinomial. Condiciones iniciales: $V_0 = -65$, $\omega_0 = 0,4$.

Otra propuesta es un ajuste polinomial (por trozos). Como el ajuste es por trozos, se

puede separar en las regiones que cuentan con distintas convexidades, de esta manera, se puede reducir el grado del segundo polinomio a dos. Vía Matlab, se obtienen las nulclinas (2.22) y (2.23).

$$\bar{p}_V(V) = \bar{p}_0(V) + \bar{p}_1(V)V + \bar{p}_2(V)V^2 + \bar{p}_3(V)V^3 \quad (2.22)$$

$$\bar{p}_\omega(V) = \bar{q}_0(V) + \bar{q}_1(V)V + \bar{q}_2(V)V^2 \quad (2.23)$$

Donde:

$$\bar{p}_0(V) \approx \begin{cases} -17,0246 & \text{si } V < -58 \\ 0,3784 & \text{si } -58 \leq V < -12 \\ 0,3738 & \text{si } -12 \leq V < 9 \\ 0,3844 & \text{si } 9 \leq V < 34 \\ 0,6010 & \text{si } 34 \leq V \end{cases}$$

$$\bar{p}_1(V) \approx \begin{cases} -0,8939 & \text{si } V < -58 \\ 0,0159 & \text{si } -58 \leq V < -12 \\ 0,0109 & \text{si } -12 \leq V < 9 \\ 0,0103 & \text{si } 9 \leq V < 34 \\ -0,0096 & \text{si } 34 \leq V \end{cases}$$

$$\bar{p}_2(V) \approx \begin{cases} -0,0157 & \text{si } V < -58 \\ 2,1741e-04 & \text{si } -58 \leq V < -12 \\ -3,3999e-04 & \text{si } -12 \leq V < 9 \\ -6,1106e-04 & \text{si } 9 \leq V < 34 \\ 2,8208e-07 & \text{si } 34 \leq V \end{cases}$$

$$\bar{p}_3(V) \approx \begin{cases} -9,4247e-05 & \text{si } V < -58 \\ -1,1966e-06 & \text{si } -58 \leq V < -12 \\ -1,6299e-05 & \text{si } -12 \leq V < 9 \\ 6,5970e-06 & \text{si } 9 \leq V < 34 \\ 2,8207e-07 & \text{si } 34 \leq V \end{cases}$$

$$\bar{q}_0(V) \approx \begin{cases} 0,1833 & \text{si } V < -58 \\ 0,4376 & \text{si } -58 \leq V < -12 \\ 0,4661 & \text{si } -12 \leq V < 9 \\ 0,4535 & \text{si } 9 \leq V < 34 \\ 0,5811 & \text{si } 34 \leq V \end{cases}$$

$$\bar{q}_1(V) \approx \begin{cases} 0,0045 & \text{si } V < -58 \\ 0,0150 & \text{si } -58 \leq V < -12 \\ 0,0161 & \text{si } -12 \leq V < 9 \\ 0,0197 & \text{si } 9 \leq V < 34 \\ 0,0127 & \text{si } 34 \leq V \end{cases}$$

$$\bar{q}_2(V) \approx \begin{cases} 2,9046e-05 & \text{si } V < -58 \\ 1,3588e-04 & \text{si } -58 \leq V < -12 \\ 5,8379e-05 & \text{si } -12 \leq V < 9 \\ -1,9671e-04 & \text{si } 9 \leq V < 34 \\ -1,0230e-04 & \text{si } 34 \leq V \end{cases}$$

La Figura (2.12) muestra una comparación entre curvas.

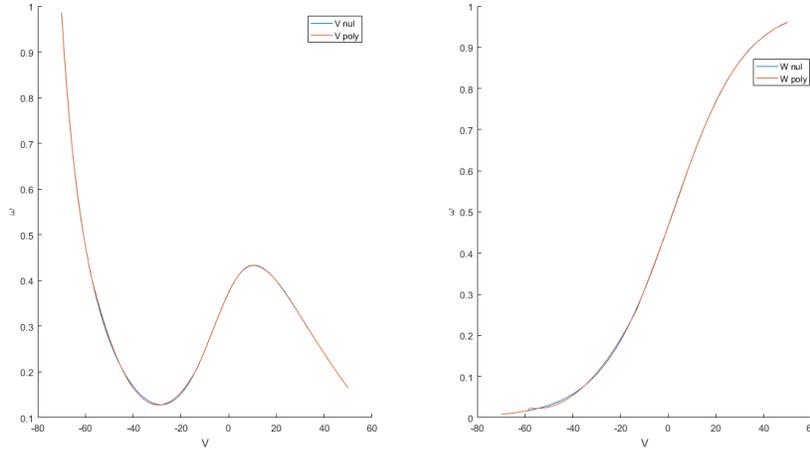


Figura 2.12: Ajuste polinomial por trozos de nulclinas de *Morris-Lecar*. Izquierda: Ajuste de nulclina asociada a V . Derecha: Ajuste de nulclina asociada a ω .

Además, (2.24) y (2.25) conforman un sistema de ecuaciones que describe el modelo equivalente, donde $\bar{\varepsilon}_V$ y $\bar{\varepsilon}_\omega$ son los ajustes de velocidad.

$$\frac{dV}{dt} = \bar{\varepsilon}_V(V, \omega)(\bar{p}_V(V) - \omega) \quad (2.24)$$

$$\frac{d\omega}{dt} = \bar{\varepsilon}_\omega(V, \omega)(\bar{p}_\omega(V) - \omega) \quad (2.25)$$

La Tabla (2.2) muestra los valores obtenidos, usando las mismas regiones especificadas en la Figura (2.9).

Función	Región	Valor (Aproximado)
$\bar{\varepsilon}_V$	R1	334.9
$\bar{\varepsilon}_V$	R2	135.1
$\bar{\varepsilon}_V$	R3	403.0
$\bar{\varepsilon}_V$	R4	231.6
$\bar{\varepsilon}_\omega$	R1	0.055
$\bar{\varepsilon}_\omega$	R2	0.055
$\bar{\varepsilon}_\omega$	R3	0.050
$\bar{\varepsilon}_\omega$	R4	0.048

Tabla 2.2: Valores aproximados de las funciones de ajuste en cada región, caso polinomial a trozos.

Las Figuras (2.13), (2.15), (2.17) y (2.19) detallan los resultados comparativos de las soluciones para distintas condiciones iniciales. Por otro lado, las Figuras (2.14), (2.16), (2.18) y (2.20), muestran las nulclinas en el plano de fase, añadiendo la trayectoria de las mismas simulaciones.

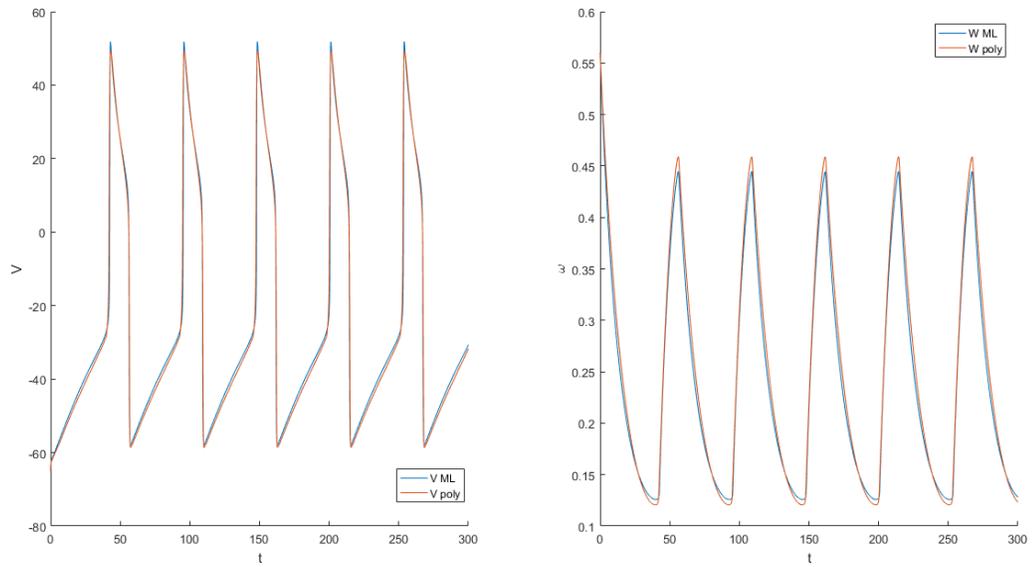


Figura 2.13: Comportamiento de Voltaje (izquierda) y variable de compuerta ω (derecha) para el modelo polinomial por partes (izquierda) y el de *Morris-Lecar* (derecha). Condiciones iniciales: $V_0 = -65$, $\omega_0 = 0,56$.

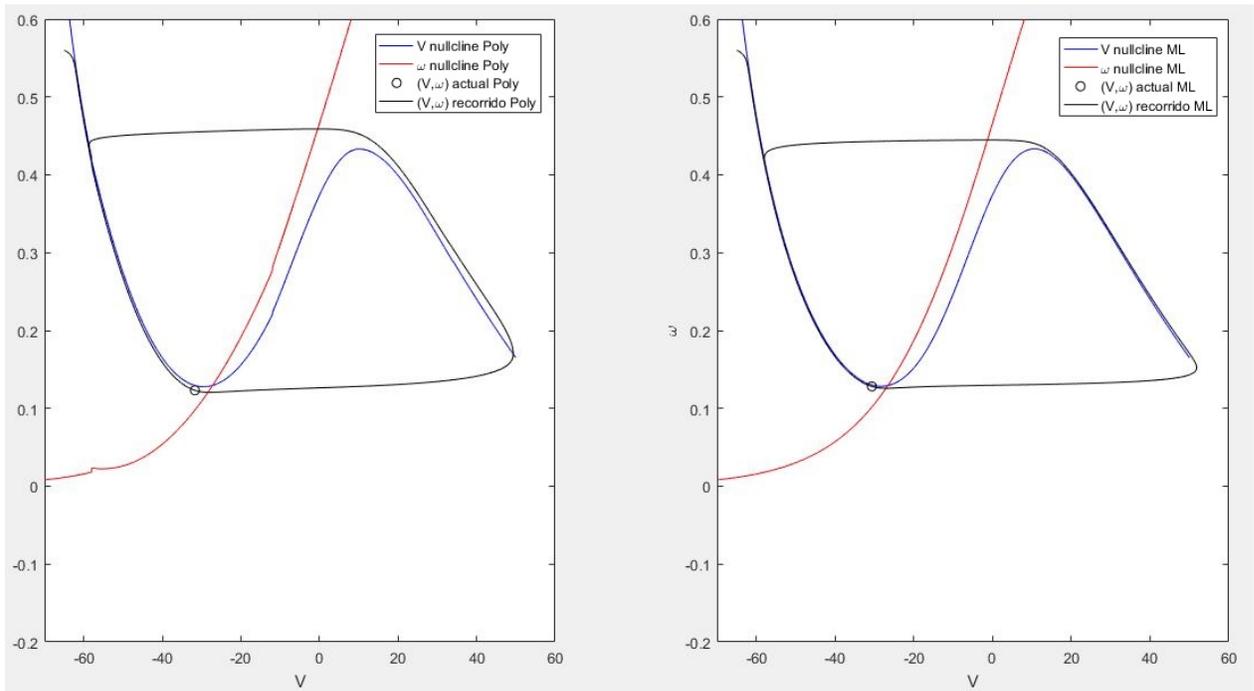


Figura 2.14: Trayectoria de simulación sobre plano de fases para modelos de *Morris-Lecar* y Polinomial por partes. Condiciones iniciales: $V_0 = -65$, $\omega_0 = 0,56$.

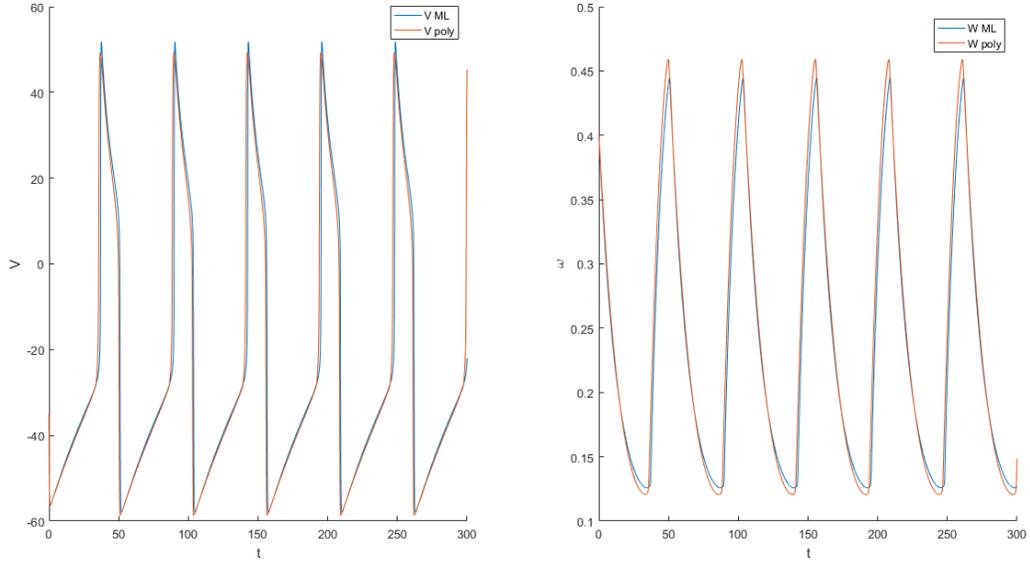


Figura 2.15: Comportamiento de Voltaje (izquierda) y variable de compuerta ω (derecha) para el modelo polinomial por partes (izquierda) y el de *Morris-Lecar* (derecha). Condiciones iniciales: $V_0 = -35$, $\omega_0 = 0,4$.

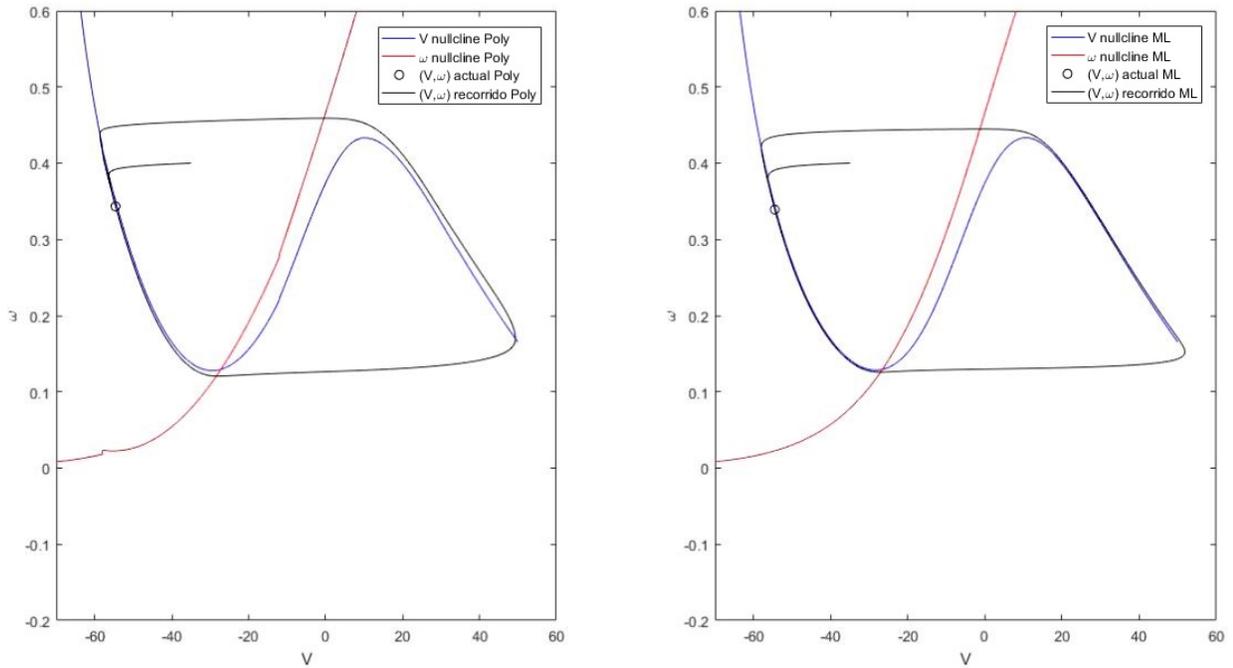


Figura 2.16: Trayectoria de simulación sobre plano de fases para modelos de *Morris-Lecar* y Polinomial por partes. Condiciones iniciales: $V_0 = -35$, $\omega_0 = 0,4$.

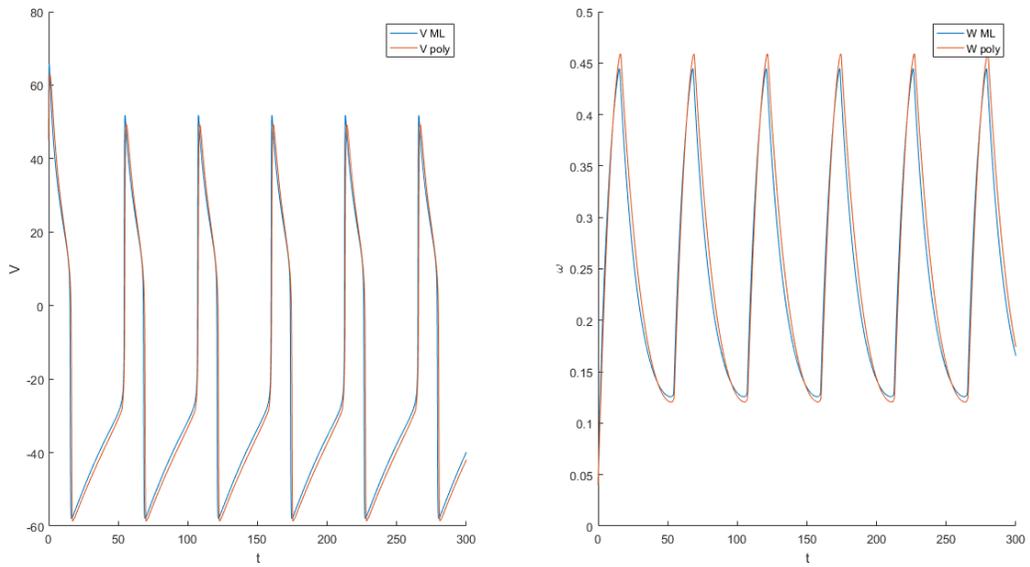


Figura 2.17: Comportamiento de Voltaje (izquierda) y variable de compuerta ω (derecha) para el modelo polinomial por partes (izquierda) y el de *Morris-Lecar* (derecha). Condiciones iniciales: $V_0 = 45$, $\omega_0 = 0,04$.

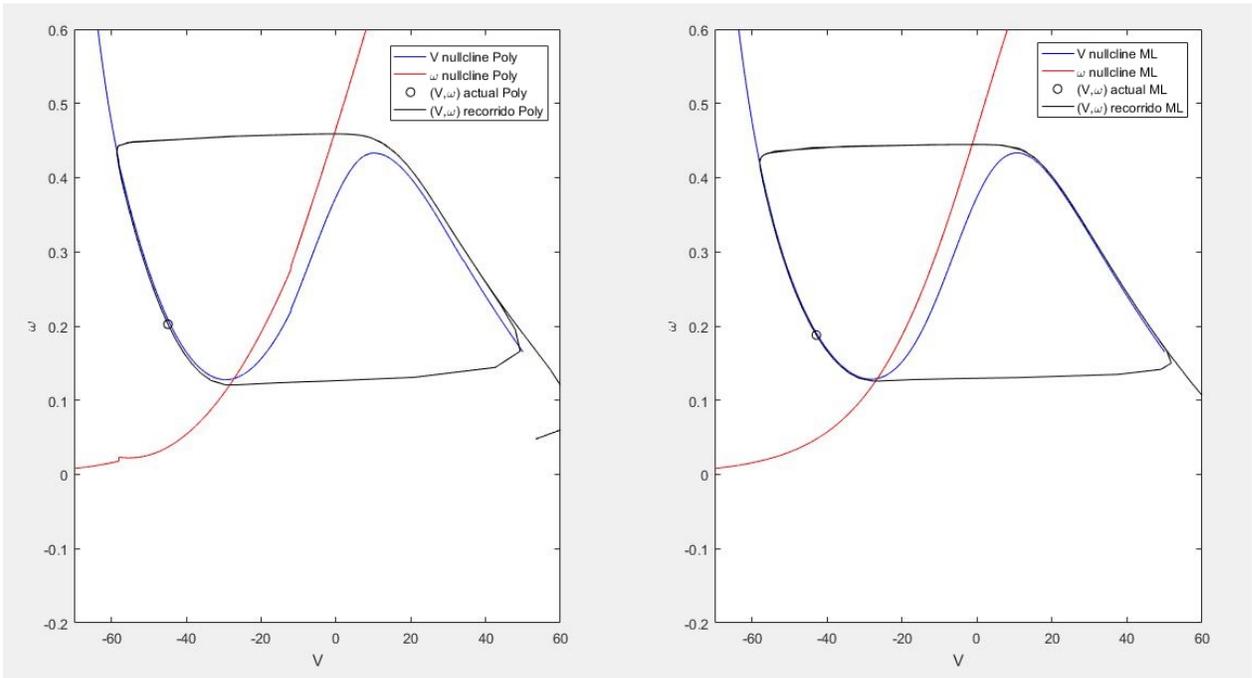


Figura 2.18: Trayectoria de simulación sobre plano de fases para modelos de *Morris-Lecar* y Polinomial por partes. Condiciones iniciales: $V_0 = 45$, $\omega_0 = 0,04$.

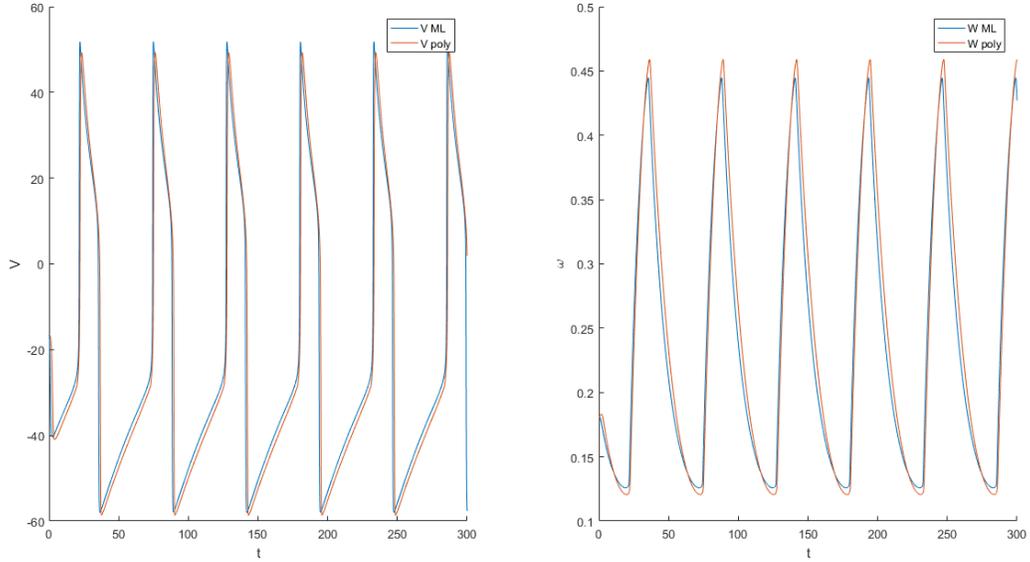


Figura 2.19: Comportamiento de Voltaje (izquierda) y variable de compuerta ω (derecha) para el modelo polinomial por partes (izquierda) y el de *Morris-Lecar* (derecha). Condiciones iniciales: $V_0 = -16,5$, $\omega_0 = 0,18$.

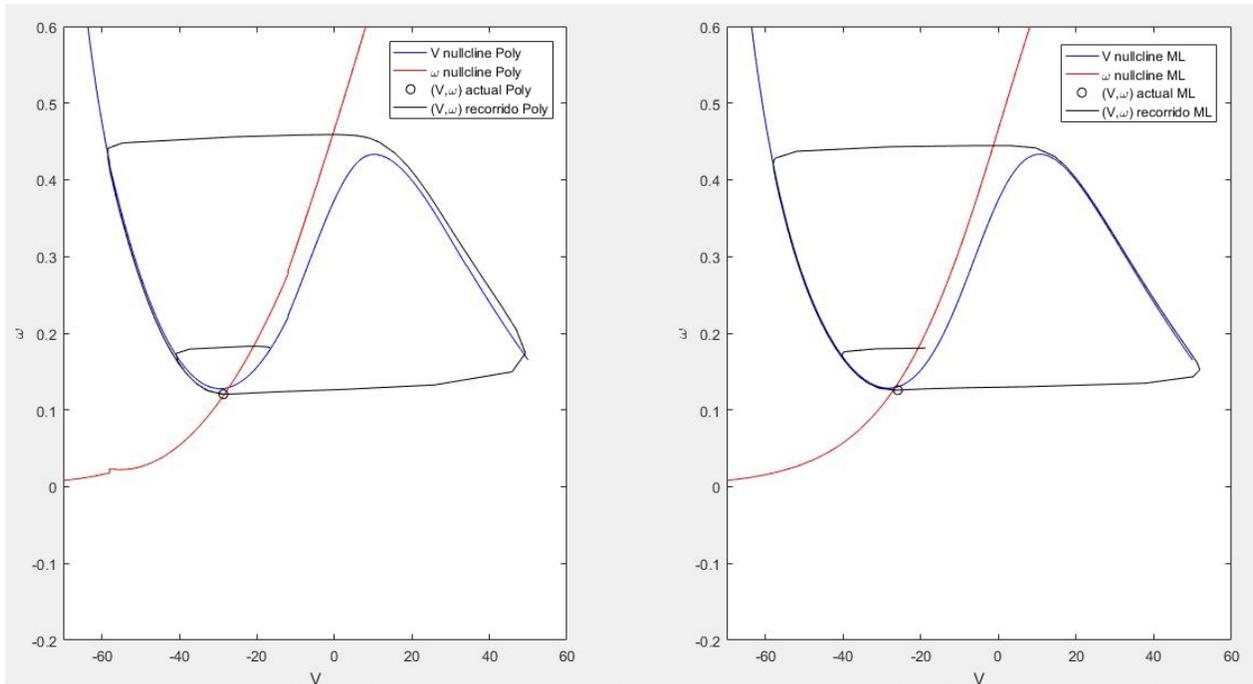


Figura 2.20: Trayectoria de simulación sobre plano de fases para modelos de *Morris-Lecar* y Polinomial por partes. Condiciones iniciales: $V_0 = -16,5$, $\omega_0 = 0,18$.

Los tiempos medidos en la resolución del sistema en Matlab, se resumen en la Tabla (2.3).

Modelo	Tiempo medido
Morris-Lecar	0.1030
Polinomial	0.4343
Polinomial por trozos	0.4021

Tabla 2.3: Tiempo medido en la resolución del sistema. Condiciones iniciales: $V_0 = -65$, $\omega_0 = 0,4$.

Capítulo 3

Método de reducción de variables usando aprendizaje de máquinas

El siguiente método consiste en la reducción de la cantidad de variables (o de dimensionalidad) que posee el sistema de ecuaciones que describe el modelo. Para esto, se introduce el método a partir de un caso específico como lo es la Descomposición Ortogonal Propia (POD), para luego realizar una generalización que consiste en entrenar una red neuronal.

3.1. Introducción al método de reducción de variables usando POD

La reducción de la cantidad de variables o de ecuaciones que conforman el sistema consiste en considerar el espacio al cual pertenecen las soluciones y encontrar una proyección en un subespacio de dimensión menor.

La Descomposición Ortogonal Propia (POD, sus siglas en inglés), es un método de reducción de dimensionalidad que se logra mediante una proyección el sistema original sobre un subespacio estrictamente menor, generado por una base de funciones obtenidas a través de datos empíricos. Debido a esto último, es utilizado para explicar información con una cantidad menor de datos con un bajo margen de error. En otras áreas, como la estadística, es conocido como *Análisis de Componentes Principales* (PCA). El método consta de dos fases, la primera consiste en resolver el sistema original y la segunda en calcular las bases del nuevo subespacio y obtener la proyección del sistema original sobre este. Se pueden encontrar aplicaciones del método para reducción de modelos realizados por Kerschen [5], Rowley [13], Lall [7] y una descripción más a fondo de Quarteroni y Rozza [12].

Dado un conjunto de datos que viven en un espacio \mathcal{R}^{n_f} , mediante POD se busca un subespacio V_r de dimensión $r < n_f$ tal que el error al proyectar esos datos sobre ese subespacio es minimizado. Esto es, dado $x(t) \in \mathcal{R}^{n_f}$, con $0 \leq t \leq T$, se busca la proyección ortogonal $P_r : \mathcal{R}^{n_f} \mapsto \mathcal{R}^{n_f}$ que minimiza el error

$$\int_0^T \|x(t) - P_r x(t)\|^2 dt \tag{3.1}$$

Como se trata de una proyección ortogonal, se cumple la igualdad

$$\operatorname{argmín}_{P_r} \|x(t) - P_r x(t)\|^2 = \operatorname{argmáx}_{P_r} \|P_r x(t)\|^2 \quad (3.2)$$

Luego, resolver el problema de minimización de (3.1) corresponde aumentar un término de energía asociado con la proyección, la cual, se puede escribir en función de los valores propios:

$$\int_0^T \|P_r x(t)\|^2 dt = \sum_{k=1}^r \lambda_k \quad (3.3)$$

Donde $\{\lambda_i\}_{i=1}^r$ es un subconjunto de tamaño r , de los n_f valores propios asociados al sistema original, que maximiza la suma de estos.

Como la matriz no necesariamente es cuadrada, no posee vectores propios, debido a esto, para resolver el problema, se buscan los valores y vectores propios (escogidos de forma que sean ortonormales) de la matriz de covarianza (3.4), la cual es de dimensión $(n_f \times n_f)$.

$$R = \int_0^T x(t)x(t)^t dt \quad (3.4)$$

En otras palabras, $\varphi_k, \forall k \in \{1, \dots, n_f\}$, ortonormales tales que cumplan que $R\varphi_k = \lambda_k\varphi_k$, $\lambda_1 \geq \lambda_2 > \dots \geq \lambda_n \geq 0$.

De esta forma, se puede obtener un subespacio óptimo de dimensión r (V_r), el cual está generado por los primeros r vectores propios $\{\varphi_1, \dots, \varphi_r\}$ y, luego, $P_r = \sum_{k=1}^r \varphi_k \varphi_k^t$. La

obtención de un modelo se logra mediante una proyección de Galerkin del sistema original, el cual es de la forma (2.3), sobre el nuevo subespacio. En otras palabras, el nuevo sistema se muestra en (3.5).

$$\dot{x}_r(t) = P_r f(x_r(t)) := g(x_r(t)) \quad (3.5)$$

Donde $x_r(t) \in V_r$. Una forma de obtener resultados numéricos es con el método de snapshots, que consiste en discretizar los saltos temporales y tomar capturas (snapshots) de x en esos tiempos. Una matriz de *snapshots* está conformada por la solución del sistema con respecto al tiempo, representando cada columna un instante de tiempo y cada fila una variable del sistema. Usando el sistema conformado por (2.1) y (2.2), la matriz de *snapshots* W es de la forma:

$$W = \begin{pmatrix} x(t_0) & x(t_1) & \dots & x(t_{N_t}) \\ \omega(t_0) & \omega(t_1) & \dots & \omega(t_{N_t}) \end{pmatrix} \in \mathbb{R}^{n_f \times (N_t+1)}$$

Donde t_0 es el tiempo inicial, t_{N_t} es el tiempo final, $(N_t + 1)$ es la cantidad de snapshots, los cuales son tomados en los tiempos $\{t_i\}_{i=0}^{N_t}$ usando las soluciones x y ω del sistema definido por (2.1) y (2.2). Obtener los vectores propios se logra vía SVD (*Singular Value Decomposition*). Si bien, al obtener una proyección lineal garantiza construir un sistema de ecuaciones con una menor cantidad de variables, no permite reconstruir el sistema original debido a que la proyección es estricta, la Figura (3.1), muestra como se logra una reconstrucción del voltaje, sin embargo, no todas las variables recrean los mismos resultados (como lo visto en la Figura (3.2)), donde la recuperación muestra un comportamiento similar pero en torno a otros valores. Esto motiva a usar una generalización de POD que, a través de funciones no lineales, pueda enviar los valores obtenidos del sistema a una variedad de dimensión menor

(no necesariamente un subespacio) y a través de funciones no lineales se pueda recuperar la información original. Obteniendo una mejor reconstrucción.

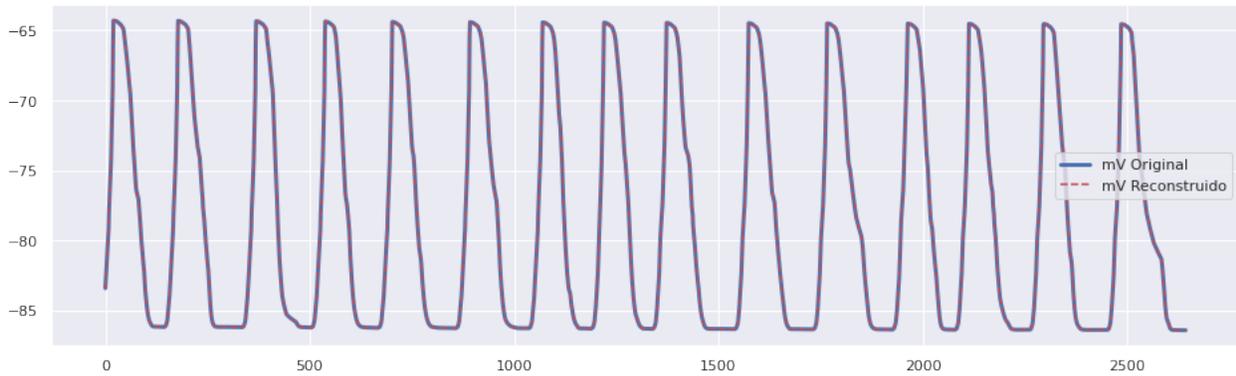


Figura 3.1: Recuperación de la variable de voltaje usando POD sobre los snapshots provenientes del modelo de *ten Tusscher-Panfilov*

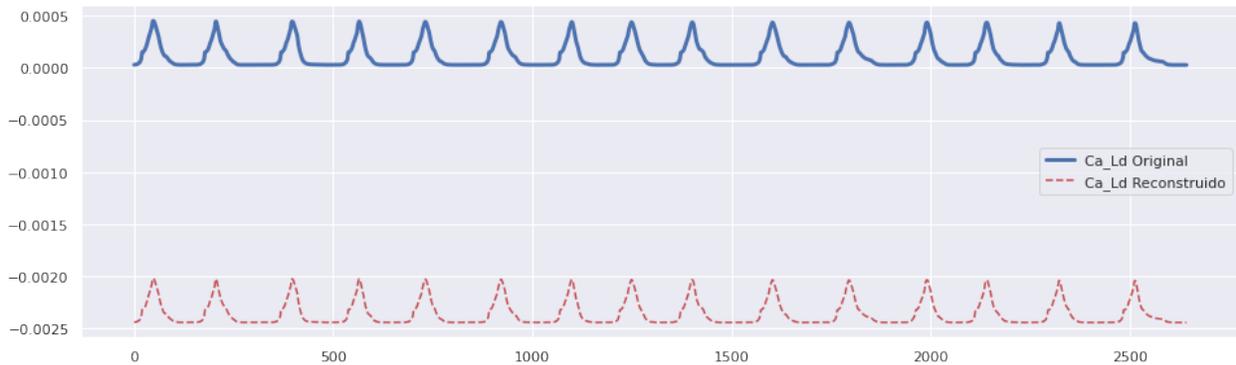


Figura 3.2: Caso de recuperación fallida usando POD sobre los snapshots provenientes del modelo de *ten Tusscher-Panfilov*

Otros resultados de reducción, como también de reconstrucción se encuentran en el Anexo.

3.2. Autoencoders como generalización de POD

Una generalización de POD a un caso no lineal, es la arquitectura de red denominada como *Autoencoder*, la cual es un tipo de Red Neuronal que aprende una representación (codificación) para un conjunto de datos de forma eficiente y no supervisada. En otras palabras, trata sobre una forma de comprimir datos. De la misma forma que ocurre con POD, normalmente se aplica en reducción de dimensionalidad. Está constituido por dos etapas principales: el Codificador que toma los datos de entrada y pasa por un *representante latente*, y el decodificador, que busca tomar este representante y recuperar los datos de entrada. Este representante latente, también llamado *code*, puede estar conformado por una o más capas. Una representación gráfica se muestra en la Figura (3.3).

¹Fuente: <https://towardsdatascience.com/generating-images-with-autoencoders-77fd3a8dd368>

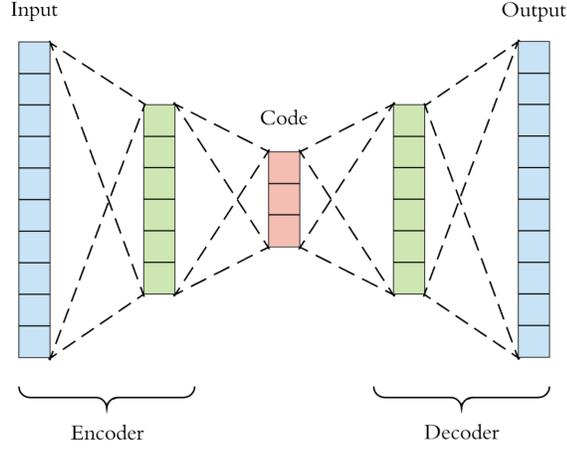


Figura 3.3: Esquema básico de un autoencoder¹.

Esto nos permite encontrar datos de menor dimensión que permiten recuperar los originales con el menor error posible, para efectos de este trabajo, el *input* corresponde a los *snapshots* provenientes del sistema definido por (2.1) y (2.2), a los cuales se les añade un ruido gaussiano. Por otro lado, el *code* representa los posibles *snapshots* de un *sistema reducido*. Sobre el uso de Autoencoders para reducción de modelos, se encuentran los trabajos de Hartman [2] y Lee [8].

3.2.1. Arquitectura de la red

Llamaremos \mathcal{X} al espacio donde se encuentran tanto el *input* como el *output*. Por otro lado, llamaremos \mathcal{F} al espacio del representante latente. Si \mathcal{F} posee una dimensión menor a \mathcal{X} , estamos ante un caso de compresión. Para efectos de este trabajo, $\mathcal{X} = \mathbb{R}^{n_f}$ y $\mathcal{F} = \mathbb{R}^d$, con $d < n_f$.

Definimos las funciones de codificación (*Encoder*) y decodificación (*Decoder*) como $\Phi : \mathcal{X} \rightarrow \mathcal{F}$ y $\Psi : \mathcal{F} \rightarrow \mathcal{X}$, respectivamente.

La red de codificación está conformada por una o más capas, cada salto de una capa a la otra es representada mediante una función de paso dada por un peso ω y un sesgo b , pasando por una función de activación no lineal σ . De esta forma, una función de paso es de la forma:

$$\phi_i(x; \theta) = \sigma(\omega_i x + b_i)$$

Donde $i \in \{1, 2, \dots, m_e\}$ representa la i -ésima capa del *Encoder*, $\phi_i(x)$ es la función de paso, ω_i el peso y b_i el sesgo del salto de la capa $(i - 1)$ a i . θ representa todos los parámetros de entrenamiento, en este caso $\theta = [(\omega_1, b_1), \dots, (\omega_i, b_i), \dots, (\omega_{m_e}, b_{m_e})]^T$, con m_e la cantidad de capas que conforman la red de codificación

De esta forma, dado un *input* x , el representante latente se define como:

$$z = \Phi(x; \theta) = \phi_{m_e} \circ \phi_{m_e-1} \circ \dots \circ \phi_1(x; \theta)$$

De la misma forma, se puede definir la decodificación:

$$\hat{x} = \Psi(z; \theta') = \psi_{m_d} \circ \psi_{m_d-1} \circ \dots \circ \psi_1(z; \theta')$$

Donde $\psi_i(z; \theta') = \sigma(\omega'_i z + b'_i)$, con $i \in \{1, 2, \dots, m_d\}$ representando la i -ésima capa del *Decoder*. Además, $\psi_i(x)$ es la función de paso, ω'_i el peso, b'_i el sesgo del salto de la capa $(i - 1)$ a i

de la decodificación. θ' representa todos los parámetros de entrenamiento, en este caso $\theta' = [(\omega'_1, b'_1), \dots, (\omega'_i, b'_i), \dots, (\omega'_{m_d}, b'_{m_d})]^T$, con m_d la cantidad de capas que conforman la red de decodificación

Como se busca obtener un *output* similar al *input* al aplicar ambas funciones, estas deben resolver el problema de minimización

$$\min_{\theta, \theta'} \|x - (\Psi \circ \Phi)(x; \theta, \theta')\|^2$$

En el cual, las variables utilizadas para la minimización son los pesos y sesgos de cada una de las capas.

Los pesos, usualmente, se aplican mediante una multiplicación matricial. Sin embargo, también es común aplicarlos mediante una convolución matricial, cuando esto ocurre, a estas redes se les denomina *Autoencoders Convolucionales* (*Convolutional Autoencoders*).

La función de paso σ suele ser no lineal, en este caso, se utiliza la tangente hiperbólica, $\tanh(\cdot)$, aplicada en cada posición (en caso de ser matricial). Como el conjunto imagen de esta función es $(-1, 1)$, es necesario normalizar los datos previamente, ya que, el *output* no podrá tener valores, en módulo, mayores o iguales a 1.

3.2.2. Entrenamiento de la red

El esquema de entrenamiento de la red es el siguiente:

1. Se define como parámetros de entrenamiento $\mu \in \mathbb{D}_{train} \equiv \{\mu_{train}^i\}_{i=1}^{n_{train}} \subset \mathbb{D}$. De forma equivalente, se define un set de parámetros para Test.
2. Se resuelve el sistema (2.3) para cada μ_{train}^i , luego, se construye la matriz de snapshots para cada set de parámetros:

$$W(\mu) := [x(t_0; \mu) \dots x(t_{N_t}; \mu)] \in \mathbb{R}^{n_f \times (N_t + 1)}$$

Donde $x(\cdot; \mu)$ es la solución del sistema para μ , t_0 el tiempo inicial y N_t el número de saltos temporales, obteniendo $(N_t + 1)$ columnas.

De esta forma, se toma en total de $|\mu|$ datos, de los cuales se extraen los necesarios para entrenamiento y los usados como test.

Durante el entrenamiento, el tratamiento de los datos será similar al de una imagen. La diferencia estará en como se distribuyen los datos. En el caso de una imagen, esta consiste en un arreglo de dos dimensiones, donde cada elemento consta de 1 canal (si la imagen está en blanco y negro) o de 3 canales (si es a colores compuestos por *RGB*). En el caso de snapshots provenientes de un sistema de ecuaciones diferenciales, se trabajan como un arreglo de n_x dimensiones (donde n_x es la dimensión espacial sobre la cual se rigen las variables físicas del sistema), donde el largo es $(N_t + 1)$ y cada elemento se conforma de n_f canales. En el caso de una célula, como se considera sólo una dirección espacial, los datos se trabajan como un arreglo de una dimensión.

3. Se normalizan los datos. Se define el operador de normalización $n : \mathbb{R}^{n_f} \mapsto \mathbb{R}^{n_f}$ como:

$$n(W^x(\mu)) = (W^x(\mu) - W_{min}^x) / (W_{max}^x - W_{min}^x)$$

Donde $W^x(\mu)$ es la columna de la matriz de snapshots dado un μ asociada a la variable

x del sistema original, W_{min}^x y W_{max}^x son los valores mínimos y máximos que toma la variable x sobre todos los datos, incluyendo set de entrenamiento y de test. La operación $(\cdot)/(\cdot)$ entrega el cociente elemento a elemento.

De la misma forma, su inversa $n^{-1} : \mathbb{R}^{n_f} \mapsto \mathbb{R}^{n_f}$ se define como:

$$n^{-1}(Z^x) = Z^x \circ (W_{max}^x - W_{min}^x) + W_{min}^x$$

Donde la operación $(\cdot) \circ (\cdot)$ representa un producto elemento a elemento (producto de Hadamard).

4. Comienza el entrenamiento:

- i. El primer paso es determinar el número de pasos de entrenamiento (N_{steps}), como también el tamaño del conjunto de datos usados para entrenar por cada paso (N_b). Se inicializan θ y θ' , debido a que la derivada suele tener una mayor variación cerca de 0 para la función de paso escogida, se toman valores aleatorios provenientes de una distribución normal centrada en 0. Esta inicialización se denomina *Xavier initialization*.
- ii. Se agrupa el conjunto de entrenamiento, X , utilizado para el paso actual.
- iii. Se calcula el representante latente:

$$Z = \Phi(X; \theta)$$

- iv. Se obtiene la reconstrucción:

$$X' = \Psi(Z; \theta')$$

- v. Se calcula el error:

$$E(\theta, \theta') = \|X'(\theta, \theta') - X\|$$

Donde $\|\cdot\| : \mathbb{R}^{n_f \times (N_t+1)} \mapsto \mathbb{R}$ es una norma.

- vi. Se realiza una iteración de un algoritmo de minimización de la función E a través de *Backpropagation*, actualizando los valores de θ y θ' . La actualización de estos valores se realiza a partir de un algoritmo (como *ADAM*, método estocástico de primer orden) bajo un *learning rate* (tasa de aprendizaje) otorgado por el usuario previamente.
 - vii. Volver a realizar los pasos (ii.) a (vi.) con los nuevos valores de θ y θ' , hasta iterar N_{steps} veces.
5. De esta forma, se recupera el *encoder* Φ y *decoder* Ψ , pudiendo calcular $z(\mu) = \Phi(n(W(\mu)))$, representando $z(\mu)$ una dinámica dependiente de los parámetros originales μ , pero con una cantidad menor de variables que la original.
 6. A partir de $Z(\mu)$, determinar el sistema de ecuaciones que satisface. Determinar si es posible recuperar $W(\mu)$ al resolver este sistema de ecuaciones de menor dimensión y aplicar a su solución el *decoder* y el inverso de la normalización, N^{-1} .

En el presente trabajo, la aplicación del método fue realizado tanto para el modelo de *Hodgkin-Huxley* como para el modelo de *ten Tusscher-Panfilov*. En ambos casos, se ha utilizado *Tensorflow*, librería de código abierto especializada en *Machine Learning*, a través de *notebooks* de *Jupyter*.

3.2.3. Medición de rendimiento

Para medir el rendimiento de la red después del entrenamiento, se calcula la distancia relativa entre el elemento del conjunto test y la salida obtenida al pasar el elemento por las funciones *Encoder* y *Decoder*, de esta forma, se le asocia un valor porcentual por cada elemento, tomando un criterio (3.6), el cual llamaremos *error relativo* y se mide por cada elemento del conjunto de las soluciones test.

$$\frac{\|x - \hat{x}\|}{\|x\|} < \varepsilon \quad (3.6)$$

En otras palabras, se espera que el error porcentual sea menor a un cierto nivel ε . De esta forma, es posible asociar un porcentaje de aciertos del conjunto de los elementos test, obtenido de la forma (3.7).

$$\frac{\text{Cantidad de Elementos test que cumplen el criterio}}{\text{Cantidad Total de Elementos Test}} \quad (3.7)$$

También es importante tomar en cuenta que el error comienza a contarse a partir de $t = 50$ (en el caso HH), que es cuando se estabilizan las curvas.

3.2.4. Recuperación del sistema de ecuaciones reducido

Es posible determinar de forma explícita los términos que conforman las funciones *Encoder* (Φ) y *Decoder* (Ψ). A partir del uso de la regla de la cadena, se obtiene (3.8), donde $x = n(y)$ son las variables originales normalizadas, f es el lado derecho del sistema original, u_b es la cota superior de cada componente, l_b la inferior y $//$ es una operación que consiste en la división, componente a componente, de los vectores involucrados.

$$\begin{aligned} \dot{z} &= \partial_t \Phi(x) \\ &= D\Phi(x) \cdot \dot{x} \\ &= D\Phi(\Phi^{-1}(z)) \cdot f(n^{-1}(x)) // (u_b - l_b) \\ &= D\Phi(\Phi^{-1}(z)) \cdot f(n^{-1}(\Phi^{-1}(z))) // (u_b - l_b) \end{aligned} \quad (3.8)$$

Denominando $\bar{g}(\cdot) := D\Phi(\Phi^{-1}(z)) \cdot f(n^{-1}(\Phi^{-1}(z))) // (u_b - l_b)$, de (3.8), se obtiene el sistema reducido (3.9).

$$\dot{z} = \bar{g}(z) \quad (3.9)$$

Sin embargo, debido a que Φ^{-1} no es conocido, se trabaja con una función que la aproxime, en este caso, Ψ . De esta manera, definiendo $g(\cdot) := D\Phi(\Psi(z)) \cdot f(n^{-1}(\Psi(z))) // (u_b - l_b)$, se obtiene (3.10).

$$\dot{z} \approx g(z) \quad (3.10)$$

Un problema discreto equivalente, es trabajar una recurrencia usando un esquema numérico, como por ejemplo, Euler. Los primeros pasos a seguir son:

- Considerar $y_0 = y(0)$, valor inicial del problema original.
- Normalizar: $x_0 = n(y_0)$
- Obtener $z_0 = \Phi(x_0)$
- Definir $y_{n+1} = F_\mu(y_n) := y_n + \Delta t f(y_n; \mu)$, con $n \in \mathbb{N}$

- Normalizar $x_{n+1} = n_{y_{n+1}}$
- Obtener $z_{n+1} = \Phi(x_{n+1})$

Con esto se obtiene z_n para $n \in \{0, 1, \dots, N_t\}$, de esta manera se puede despejar numéricamente g a partir de $z_{n+1} = G_\mu(z_n) := z_n + \Delta t g(y_n; \mu)$. Por lo cual, nos basta con calcular $G_\mu(\cdot)$. Para mayor claridad, el esquema (3.11) resume todos los cálculos realizados para cada paso.

$$\begin{array}{ccccc}
y_0 & \xrightarrow{n} & x_0 & \xrightarrow{\Phi} & z_0 \\
F_\mu \downarrow & & \downarrow & & \downarrow G_\mu \\
y_1 & \xrightarrow{n} & x_1 & \xrightarrow{\Phi} & z_1 \\
F_\mu \downarrow & & \downarrow & & \downarrow G_\mu \\
\vdots & & \vdots & & \vdots \\
y_n & \xrightarrow{n} & x_n & \xrightarrow{\Phi} & z_n
\end{array} \tag{3.11}$$

Gracias al esquema (3.11) se concluye:

$$G_\mu \circ \Phi \circ n = \Phi \circ n \circ F_\mu \tag{3.12}$$

Esto implica que:

$$G_\mu = \Phi \circ n \circ F_\mu \circ n^{-1} \circ \Phi^{-1} \tag{3.13}$$

Pero al desconocer Φ^{-1} , se cambia por una aproximación, el cual resulta ser Ψ . Obteniendo de esta forma:

$$G_\mu \approx \Phi \circ n \circ F_\mu \circ n^{-1} \circ \Psi \tag{3.14}$$

Ambas formas de abordar el problema son equivalentes y los resultados no cambian al momento de resolver el método continuo debido a que debe pasar por un esquema que lo discretice previamente.

Obtención de \bar{g} vista como solución de un problema inverso

Suponiendo que el nuevo sistema es autónomo y que \bar{g} sea continua, en otras palabras $\bar{g} \in \mathcal{C}([-1, 1]^d)$ donde d es la dimensión de z y $z \in \mathcal{C}([0, T])$. Entonces, como se espera que $\bar{g}(z) = \dot{z}$, se puede considerar un operador integral $K : (\mathcal{C}([-1, 1]^d), \|\cdot\|_\infty) \mapsto (\mathcal{C}([0, T]), \|\cdot\|_\infty)$, que cumpla (3.15).

$$K(\bar{g}) = z \tag{3.15}$$

De esta manera, si \bar{g} cumple (3.15), entonces \bar{g} es el lado derecho del sistema reducido. De esta forma, la solución al problema inverso, que es encontrar un K^{-1} , el cual es un operador diferencial, permite obtener los valores de \bar{g} , estos se obtienen a partir de (3.9), sin embargo, se desconoce su valor debido a la presencia de Φ^{-1} . Para obtener un valor numérico, se recurre a Ψ , término que se puede catalogar como una aproximación de Φ^{-1} , siendo una función que genera valores similares, pero perturbados por un término de error llamado ε , el cual cumple (3.16).

$$\Psi(z) = \Phi^{-1}(z) - \varepsilon(z) \tag{3.16}$$

Sin embargo, si el problema inverso está mal puesto, pequeñas perturbaciones como ε podría provocar resultados no deseados. Esto motiva a estudiar el error asociado a la recuperación del sistema reducido. Un caso de un problema similar que está mal puesto, se encuentra en el *Ejemplo 1.15* de *Kirsch* [6], pues, al operar un error de baja amplitud con una derivada, este podría verse aumentado debido a su naturaleza de alta frecuencia.

Error de la recuperación del sistema reducido

El nuevo error se obtiene al aproximar el nuevo sistema en función del error ε . Esto es, un error \mathcal{E} que cumpla la igualdad (3.17).

$$D\Phi(\Phi^{-1}(z)) [f(n^{-1}(\Phi^{-1}(z)))/(u_b - l_b)] = D\Phi(\Psi(z))f(n^{-1}(\Psi(z)))/(u_b - l_b) + \mathcal{E}(\varepsilon, z) \quad (3.17)$$

Para verificar si la reconstrucción del sistema reducido proviene de un problema que está *mal puesto* o no, se acota el término \mathcal{E} , observando el peor error que se puede alcanzar. Un valor alto podría significar que la aproximación vista en (3.10), produzca un error mayor al esperado y las soluciones sean diferentes a las esperadas.

Efecto de la Arquitectura del Decoder sobre el error de aproximación

El error \mathcal{E} , se puede definir como (3.19).

$$\begin{aligned} \mathcal{E}(\varepsilon, z) &= D\Phi(\Phi^{-1}(z)) [f(n^{-1}(\Phi^{-1}(z)))/(u_b - l_b)] - [D\Phi(\Psi(z))f(n^{-1}(\Psi(z)))/(u_b - l_b)] \\ &= D\Phi(\Phi^{-1}(z)) [f(n^{-1}(\Phi^{-1}(z)))/(u_b - l_b)] - D\Phi(\Phi^{-1}(z) - \varepsilon) [f(n^{-1}(\Phi^{-1}(z) - \varepsilon))/(u_b - l_b)] \end{aligned} \quad (3.18)$$

Lo cual, implica que la dependencia de la arquitectura del decoder está implícita en los valores que toma ε .

Efecto de la Arquitectura del Encoder sobre el error de aproximación

Caso Lineal Afín: Cuando es de sólo una capa, esto es, cuando el Encoder cumple $\Phi(z) = A_\Phi \cdot x + b_\Phi$, la nueva igualdad es (3.21).

$$\begin{aligned} D\Phi(\Phi^{-1}(z)) [f(n^{-1}(\Phi^{-1}(z)))/(u_b - l_b)] &= A_\Phi f(n^{-1}(\Psi(z)))/(u_b - l_b) + A_\Phi E_f(\varepsilon, z)/(u_b - l_b) \\ &= A_\Phi [f(n^{-1}(\Psi(z))) + E_f(\varepsilon, z)]/(u_b - l_b) \end{aligned} \quad (3.21)$$

Donde $\mathcal{E} = A_\Phi E_f(\varepsilon, z)/(u_b - l_b)$ es el nuevo error. El término E_f es, en este caso, el error relativo al considerar la aproximación de $f(n^{-1}(\Phi^{-1}(z)))$ a través de $f(n^{-1}(\Psi(z)))$.

Al agregar más capas al Encoder, los términos de E_f no cambian, sin embargo, cambia la matriz de peso por la cual se multiplica. Sea Φ_i , la i -ésima capa de las n_e capas que componen un Encoder y A_{Φ_i} su matriz de peso, entonces, el nuevo error es (3.22).

$$\mathcal{E} = A_{\Phi_{n_e}} A_{\Phi_{n_e-1}} \dots A_{\Phi_2} A_{\Phi_1} E_f/(u_b - l_b) \quad (3.22)$$

Redefiniendo $A_\Phi = A_{\Phi_{n_e}} A_{\Phi_{n_e-1}} \dots A_{\Phi_2} A_{\Phi_1}$, nuevamente, se obtiene una expresión similar a (3.21).

Como A_Φ también multiplica a $f(n^{-1}(\Psi(z)))$, el acotamiento interesante es el del término $E_f/(u_b - l_b)$.

Caso con función de paso \tanh : Al usar \tanh como función de paso, el Encoder (en una capa) es de la forma $\Phi(x) = \tanh(A_\Phi x + b_\Phi)$, entonces, su Jacobiano está dado por (3.23).

$$D\Phi(\Psi(z) + \varepsilon(z))[x] = \text{sech}^2(A_\Phi(\Psi(z) + \varepsilon(z)) + b_\Phi) \circ A_\Phi[x] \quad (3.23)$$

Donde sech^2 se aplica a cada posición y \circ es un producto donde el i -ésimo elemento del vector $\text{sech}^2(A_\Phi(\Psi(z) + \varepsilon(z)) + b_\Phi)$ multiplica la i -ésima fila de la matriz A_Φ . Debido a que no se puede separar de forma directa a $\Psi(z)$ y ε , se resta y suma un término a conveniencia, tal como se muestra en (3.24).

$$D\Phi(\Psi(z) + \varepsilon(z))[x] = D\Phi(\Psi(z))[x] + D\Phi(\Psi(z) + \varepsilon(z))[x] - D\Phi(\Psi(z))[x] \quad (3.24)$$

Obteniendo (3.25)

$$\begin{aligned} \mathcal{E} &= (\text{sech}^2(A_\Phi(\Psi(z) + \varepsilon(z)) + b_\Phi) - \text{sech}^2(A_\Phi\Psi(z) + b_\Phi)) \circ A_\Phi f(n^{-1}(\Psi(z) + \varepsilon)) / (u_b - l_b) \\ &\quad + \text{sech}^2(A_\Phi\Psi(z) + b) \circ A_\Phi E_f / (u_b - l_b) \\ &= (\text{sech}^2(A_\Phi(\Psi(z) + \varepsilon(z)) + b_\Phi) - \text{sech}^2(A_\Phi\Psi(z) + b_\Phi)) \circ A_\Phi f(n^{-1}(\Psi(z))) / (u_b - l_b) \\ &\quad + \text{sech}^2(A_\Phi(\Psi(z) + \varepsilon) + b) \circ A_\Phi E_f / (u_b - l_b) \end{aligned} \quad (3.25)$$

Donde E_f es el mismo error relativo visto en el caso lineal afín. Como $0 \leq \text{sech}^2(x) < 1$, se cumple:

$$\|\text{sech}^2(A_\Phi(\Psi(z) + \varepsilon) + b_\Phi)\|_\infty < 1$$

$$\|\text{sech}^2(A_\Phi(\Psi(z) + \varepsilon) + b_\Phi) - \text{sech}^2(A_\Phi\Psi(z) + b_\Phi)\|_\infty < 1$$

Al añadir más capas, similar al caso lineal afín, se multiplican las derivadas de las funciones anidadas, formando productos de las matrices de peso de cada capa A_Φ , cada una multiplicada por filas por sech^2 evaluados en las capas anteriores. Llamando a la i -ésima capa evaluada en las anteriores como Φ_i^C , esto es

$$\Phi_i^C(x) = \Phi_i(\Phi_{i-1}(\dots(\Phi_1(x))))$$

Se obtiene:

$$\partial_t \Phi_i^C = \text{sech}^2(A_{\Phi_i} \Phi_{i-1}^C + b_{\Phi_i}) \circ A_{\Phi_i} \partial_t \Phi_{i-1}^C \quad (3.26)$$

Usando desigualdad triangular y desigualdad de Hölder reiteradas veces, se obtiene (3.27).

$$\begin{aligned}
\|\mathcal{E}\|_2 &\leq \|(\operatorname{sech}^2(A_{\Phi_{n_e}} \Phi_{n_e-1}^C(\Psi(z) + \varepsilon) + b_{\Phi_{n_e}}) \circ A_{n_e} \dots \operatorname{sech}^2(A_{\Phi_1}(\Psi(z) + \varepsilon) + b_{\Phi_1}) \circ A_1 \\
&\quad - \operatorname{sech}^2(A_{\Phi_{n_e}} \Phi_{n_e-1}^C(\Psi(z)) + b_{\Phi_{n_e}}) \circ A_{n_e} \dots \operatorname{sech}^2(A_{\Phi_1}(\Psi(z)) + b_{\Phi_1})) \circ A_1\|_2 \\
&\quad \|f(n^{-1}(\Psi(z) + \varepsilon))/(u_b - l_b)\|_2 \\
&\quad + \|\operatorname{sech}^2(A_{\Phi_{n_e}} \Phi_{n_e-1}^C(\Psi(z) + \varepsilon) + b_{\Phi_{n_e}}) \circ A_{n_e} \dots \operatorname{sech}^2(A_{\Phi_1}(\Psi(z) + \varepsilon) + b_{\Phi_1}) \circ A_1 [E_f/(u_b - l_b)]\|_2 \\
&\leq (\|\operatorname{sech}^2(A_{\Phi_{n_e}} \Phi_{n_e-1}^C(\Psi(z) + \varepsilon) + b_{\Phi_{n_e}}) \circ A_{n_e} \dots \operatorname{sech}^2(A_{\Phi_1}(\Psi(z) + \varepsilon) + b_{\Phi_1}) \circ A_1\|_2 \\
&\quad + \|\operatorname{sech}^2(A_{\Phi_{n_e}} \Phi_{n_e-1}^C(\Psi(z)) + b_{\Phi_{n_e}}) \circ A_{n_e} \dots \operatorname{sech}^2(A_{\Phi_1}(\Psi(z)) + b_{\Phi_1}) \circ A_1\|_2) \\
&\quad \|[f(n^{-1}(\Psi(z) + \varepsilon))/(u_b - l_b)]\|_\infty \\
&\quad + \|\operatorname{sech}^2(A_{\Phi_{n_e}} \Phi_{n_e-1}^C(\Psi(z) + \varepsilon) + b_{\Phi_{n_e}}) \circ A_{n_e} \dots \operatorname{sech}^2(A_{\Phi_1}(\Psi(z) + \varepsilon) + b_{\Phi_1}) \circ A_1\|_2 \| [E_f/(u_b - l_b)]\|_\infty \\
&\leq (\|\operatorname{sech}^2(A_{\Phi_{n_e}} \Phi_{n_e-1}^C(\Psi(z) + \varepsilon) + b_{\Phi_{n_e}})\|_\infty \|A_{n_e} \dots \operatorname{sech}^2(A_{\Phi_1}(\Psi(z) + \varepsilon) + b_{\Phi_1}) \circ A_1\|_2 \\
&\quad + \|\operatorname{sech}^2(A_{\Phi_{n_e}} \Phi_{n_e-1}^C(\Psi(z)) + b_{\Phi_{n_e}})\|_\infty \|A_{n_e} \dots \operatorname{sech}^2(A_{\Phi_1}(\Psi(z)) + b_{\Phi_1}) \circ A_1\|_2) \\
&\quad \|[f(n^{-1}(\Psi(z) + \varepsilon))/(u_b - l_b)]\|_\infty \\
&\quad + \|\operatorname{sech}^2(A_{\Phi_{n_e}} \Phi_{n_e-1}^C(\Psi(z) + \varepsilon) + b_{\Phi_{n_e}})\|_\infty \|A_{n_e} \dots \operatorname{sech}^2(A_{\Phi_1}(\Psi(z) + \varepsilon) + b_{\Phi_1}) \circ A_1\|_2 \| [E_f/(u_b - l_b)]\|_\infty \\
&\leq (\|A_{n_e} \dots \operatorname{sech}^2(A_{\Phi_1}(\Psi(z) + \varepsilon) + b_{\Phi_1}) \circ A_1\|_2 + \|A_{n_e} \dots \operatorname{sech}^2(A_{\Phi_1}(\Psi(z)) + b_{\Phi_1}) \circ A_1\|_2) \\
&\quad \|[f(n^{-1}(\Psi(z) + \varepsilon))/(u_b - l_b)]\|_\infty \\
&\quad + \|A_{n_e} \dots \operatorname{sech}^2(A_{\Phi_1}(\Psi(z) + \varepsilon) + b_{\Phi_1}) \circ A_1\|_2 \| [E_f/(u_b - l_b)]\|_\infty \\
&\leq \dots \\
&\leq \|A_{n_e}\|_\infty \|A_{n_e-1}\|_\infty \dots \|A_2\|_\infty \|A_1\|_2 (2\|[f(n^{-1}(\Psi(z) + \varepsilon))/(u_b - l_b)]\|_\infty + \|[E_f/(u_b - l_b)]\|_\infty) \quad (3.27)
\end{aligned}$$

Donde $\|\cdot\|_2$ y $\|\cdot\|_\infty$ aplicadas en matrices son normas inducidas.

El término $\|A_{n_e}\|_\infty \|A_{n_e-1}\|_\infty \dots \|A_2\|_\infty \|A_1\|_2$ es independiente de ε y aparece en todos los términos de acotamiento por igual, esto incluye $f(n^{-1}(\Psi(z)))/(u_b - l_b)$, por lo cual, el acotamiento del nuevo error relativo está acotado, superiormente, por $2\|[f(n^{-1}(\Psi(z) + \varepsilon))/(u_b - l_b)]\|_\infty$ y $\|[E_f/(u_b - l_b)]\|_\infty$, el error relativo en el caso Lineal afín. Esto motiva a estudiar este último término, ya que, si sus valores son altos, también puede amplificar los valores que puede tomar \mathcal{E} , independiente del caso (ya sea lineal afín o usando \tanh).

Acotamiento de E_f

Sea f es una función cuyas componentes son continuas y suficientemente regulares para poseer derivadas continuas. En el caso de *HH* y *ten Tusscher-Panfilov*, ambos sistemas de ecuaciones poseen términos que se indefinen en un punto, sin embargo, son reparables a partir del uso del límite de las funciones, lo mismo ocurre con sus derivadas, logrando que sea una función tipo \mathcal{C}^∞ .

Como la variable V y las variables de compuerta, están normalizadas, entonces los valores de ε también están acotados, por lo cual, el conjunto donde reside la suma de $\Psi(z) + \varepsilon$ es acotado. De esta manera, sobre la cerradura de este conjunto, la función f y las derivadas de sus componentes alcanzan un máximo y mínimo, resultando ser Lipschitz en esa localidad. Se denomina L_i a la constante de Lipschitz de la i -ésima componente.

Para acotar $|E_f|$, se realiza componente a componente, llamando E_{f_i} a la i -ésima componente de E_f y se busca relacionar el valor máximo de estas componentes con los coeficientes de Lipschitz de la función f .

Como $f(n^{-1}(\cdot))$ es, por lo menos, \mathcal{C}^∞ , su polinomio de Taylor de orden 1 en torno a $\Psi(z)$ es (3.28).

$$f(n^{-1}(\Psi(z) + \varepsilon)) = f(n^{-1}(\Psi(z))) + Jf(n^{-1}(\Psi(z)))\varepsilon \circ (u_b - l_b) + O(\|\varepsilon\|_2^2) \quad (3.28)$$

Luego, gracias a lo visto en el caso Lineal Afín, se concluye (3.29).

$$E_f = Jf(n^{-1}(\Psi(z)))\varepsilon \circ (u_b - l_b) + O(\|\varepsilon\|_2^2) \quad (3.29)$$

Obteniendo (3.30) para cada componente de E_f .

$$|E_{f_i}(\varepsilon, z)| \leq L_i |y_i^{max} - y_i^{min}| \|\varepsilon\|_2 + O(\|\varepsilon\|_2^2) \quad (3.30)$$

De esta manera, para un dominio como $E^0 = \{\varepsilon \in \mathbb{R}^4 : \varepsilon \ll 1\}$, la cota de cada componente de E_f viene dada por la constante de Lipschitz de la misma componente de f , multiplicada por el término de normalización de esa componente, concluyendo que $E(\cdot, z)$ es Lipschitz sobre ese dominio con respecto a la variable ε y su constante de Lipschitz es $L\|u_b - l_b\|_\infty$, con L la constante de Lipschitz de f .

Finalmente, a partir del valor de esta constante se puede concluir el comportamiento de E_f con respecto a ε , por ejemplo, si es menor a 1, la función es contractante con respecto a las variables ε , lo que implica que los errores se reducen tras calcular E_f . Por otro lado, si es mayor a 1, no se podría concluir lo mismo, incluso, podría suponer una amplificación del error. Esto último, ocurriría debido a que el (3.15), podría estar *mal puesto*.

3.3. Resultados del método de reducción de variables usando Autoencoders

Hodgkin-Huxley

Como este modelo cuenta con 4 variables, el "input" es un arreglo de 4 canales. La reducción del modelo consiste en generar un "code" de 2 canales, reduciendo en dos la cantidad de estos. El conjunto de entrenamiento cuenta con 15365 elementos y el de test cuenta con 635. Se utilizan todos los parámetros que componen el sistema para la creación de estos datos, el objetivo es tener datos variados y evitar *overfitting*. Además, una vez revisados todos los datos, se realiza una nueva época de entrenamiento, esta vez, añadiendo un ruido Gaussiano. La primera aproximación consiste en reducir directamente los 4 canales (incluye V) a dos canales. La Figura (3.4) es una representación esquemática de las capas.

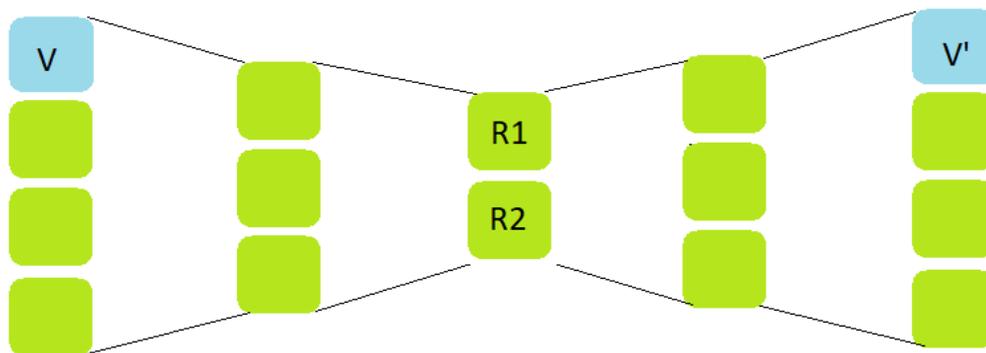


Figura 3.4: Primer esquema de reducción usando Autoencoders sobre modelo HH.

La Figura (3.5) muestra el resultado de usar la distancia entre entrada y salida en norma L^2 como función de pérdida, como también la distancia en norma L^∞ .

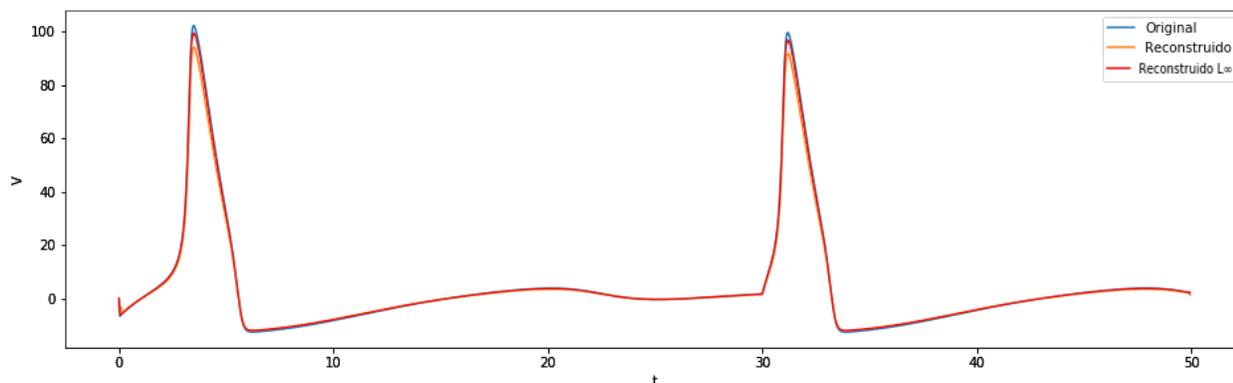


Figura 3.5: Resultado de reconstrucción usando dos diferentes normas en un elemento del conjunto de test.

En la Tabla (3.1), se observan los valores obtenidos al trabajar con distintos tipos de medición de error sobre el conjunto de test.

Métrica	Expresión	Valor aproximado
MAE (Error absoluto medio)	$\varepsilon_{\text{MAE}} = \frac{1}{ \mathcal{W} } \sum_{(i,j,k) \in \mathcal{W}} x_{i,j,k} - \hat{x}_{i,j,k} $	0.0165
RMSE (Raíz del error cuadrático medio)	$\varepsilon_{\text{RMSE}} = \sqrt{\frac{1}{ \mathcal{W} } \sum_{(i,j,k) \in \mathcal{W}} x_{i,j,k} - \hat{x}_{i,j,k} ^2}$	0.0206
Distancia en norma L^∞	$\varepsilon_{L^\infty} = \sup_{(i,j,k) \in \mathcal{W}} x_{i,j,k} - \hat{x}_{i,j,k} $	0.1032

Tabla 3.1: Valores obtenidos para cada métrica de la distancia entre el valor de salida y de entrada.

Donde $\mathcal{W} = \{(i, j, k); i \in \{1, \dots, n_{\text{test}}\}, j \in \{1, \dots, n_t\}, k \in \{1, \dots, n_f\}\}$, con n_{test} la cantidad de test.

Al usar el criterio propuesto en este trabajo sobre el conjunto determinado por los elementos test, la Figura (3.6) muestra como varía el porcentaje de aciertos dependiendo del nivel optado para ambas normas.

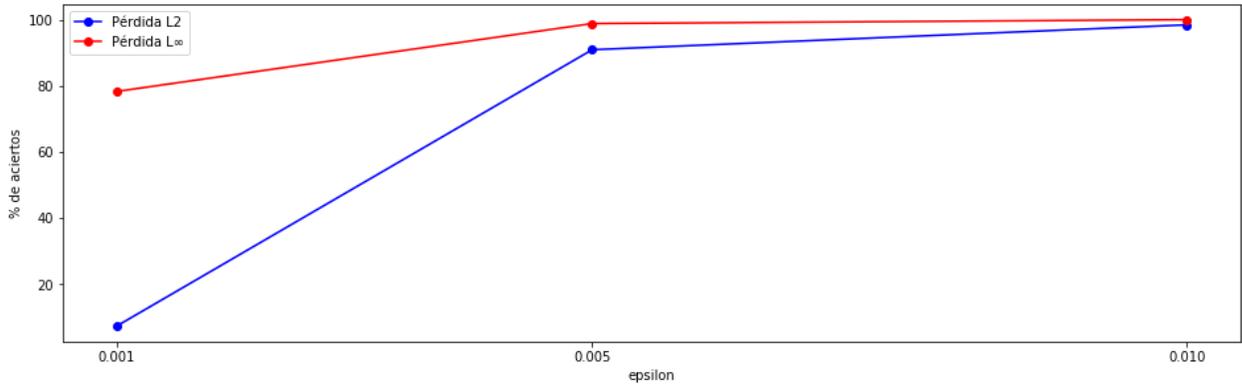


Figura 3.6: Porcentaje de test que cumplen el criterio (3.6).

Con respecto al representante latente o *code*, se obtiene un arreglo de dos canales en vez de 4, la Figura (3.7) muestra el primer canal.

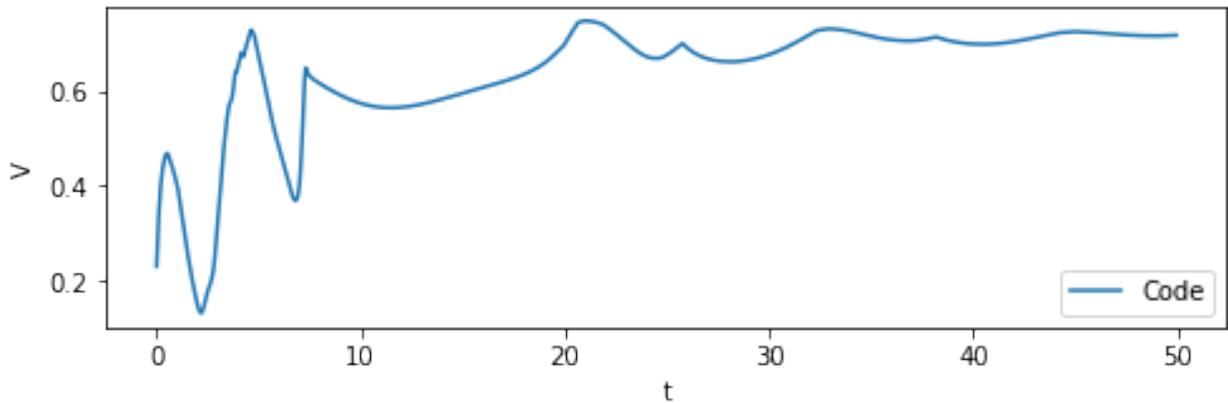


Figura 3.7: Primera variable del representante latente.

A partir de la Figura (3.7), se plantea el reto de mantener el voltaje, incluso en el representante latente. Para esto, se cambia a un esquema que adjunta el canal V en cada una de las capas. En este caso, el autoencoder trabaja para reducir a un nuevo canal, resultando en un "code" de dos canales, donde uno de estos es el voltaje. La Figura (3.8) es una representación simplificada del nuevo esquema.

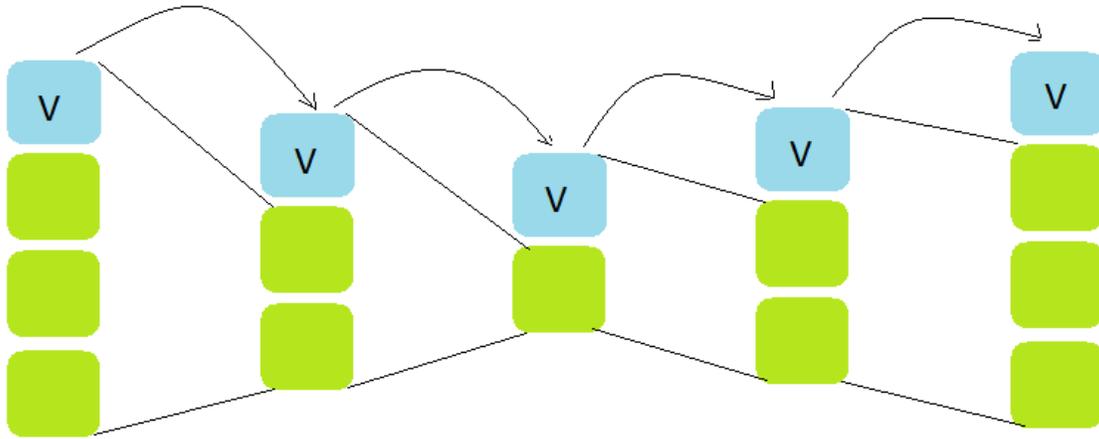


Figura 3.8: Nuevo esquema de reducción usando Autoencoders sobre modelo HH.

Considerando $\varepsilon = 0,05$, el porcentaje de aciertos usando norma L^∞ es aproximadamente 94,5% cuando el Encoder tiene 2 capas y el Decoder posee 8. La Figura (3.9) muestra los dos canales formados en *code*, siendo V uno de ellos, mientras las Figuras (3.10), (3.11), (3.12) y (3.13) muestran la reconstrucción de los 4 canales en uno de los casos test cuyo error relativo es menor a ε (recordemos que esta norma se calcula a partir de $t = 50$). Por otro lado, las Figuras (3.14), (3.15), (3.16) y (3.17) muestran un caso en el cual no se logra. Sobre el número de capas, la Figura (3.18) muestra el valor de la función de pérdida luego de la última iteración con respecto a la variación de capas, mientras que la Figura (3.19), muestra el tiempo promedio de cada época (número de iteraciones que demora el entrenamiento en estudiar todos los datos).

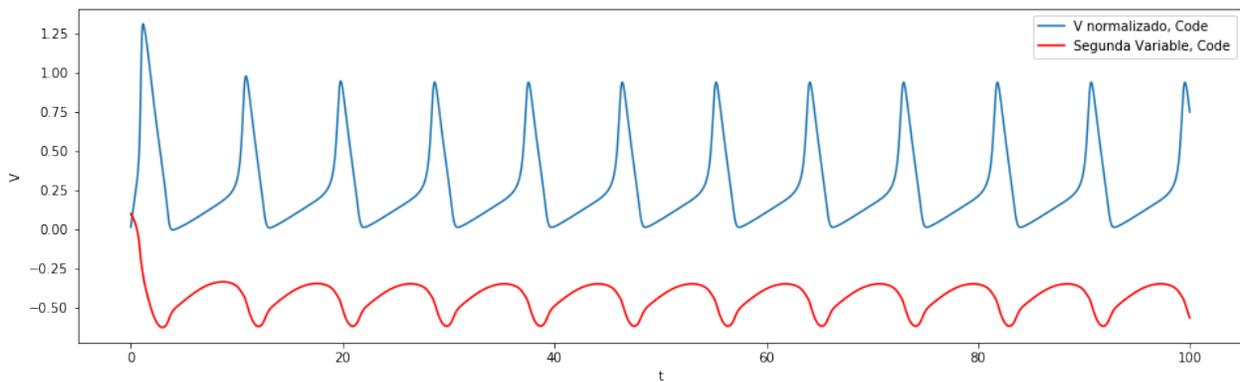


Figura 3.9: Code obtenido a partir de input mostrado en las Figuras (3.10), (3.11), (3.12) y (3.13).

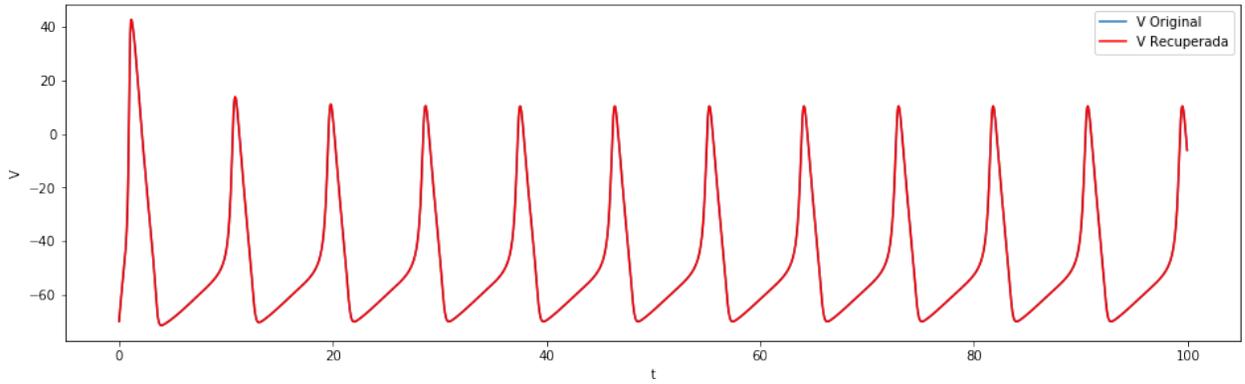


Figura 3.10: Voltaje [mV] con respecto al tiempo en [ms]. Caso test cuyo error relativo es menor al valor indicado. 2 capas de Encoder y 8 de Decoder

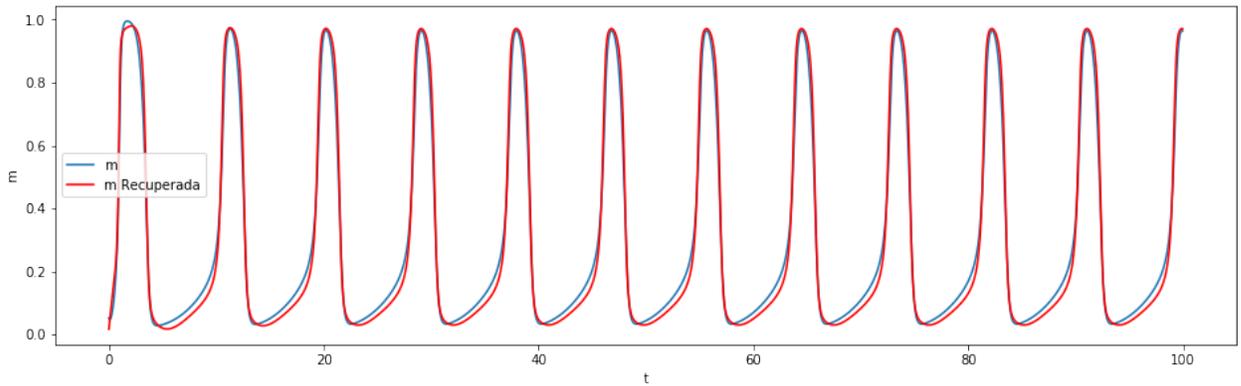


Figura 3.11: m con respecto al tiempo en [ms]. Caso test cuyo error relativo es menor al valor indicado. 2 capas de Encoder y 8 de Decoder

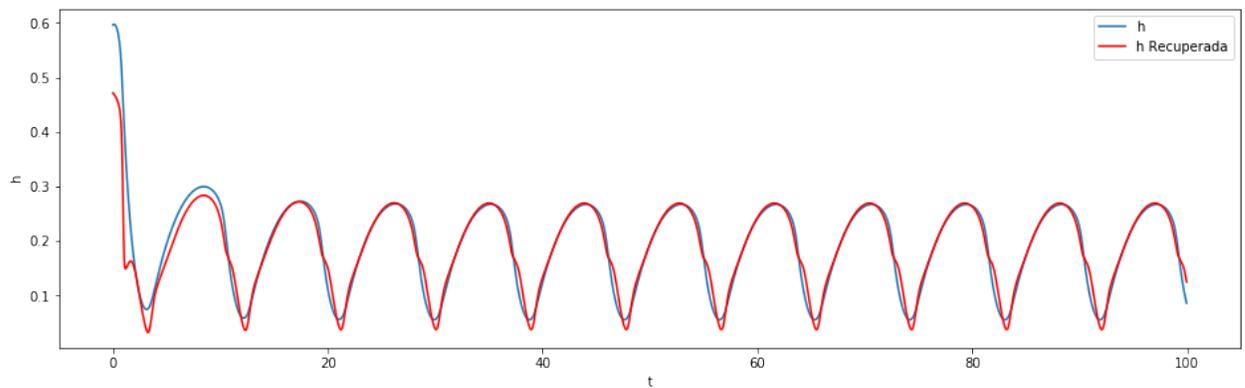


Figura 3.12: h con respecto al tiempo en [ms]. Caso test cuyo error relativo es menor al valor indicado. 2 capas de Encoder y 8 de Decoder

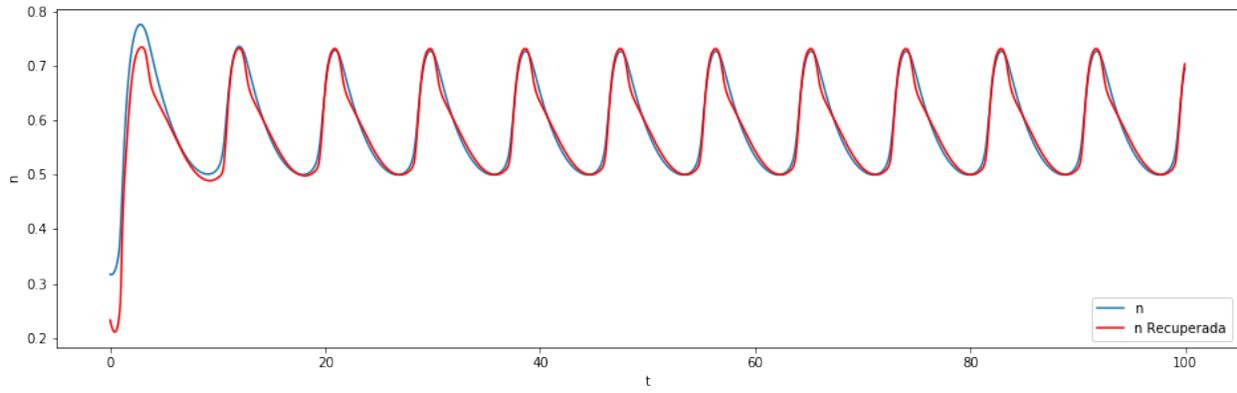


Figura 3.13: n con respecto al tiempo en $[ms]$. Caso test cuyo error relativo es menor al valor indicado. 2 capas de Encoder y 8 de Decoder

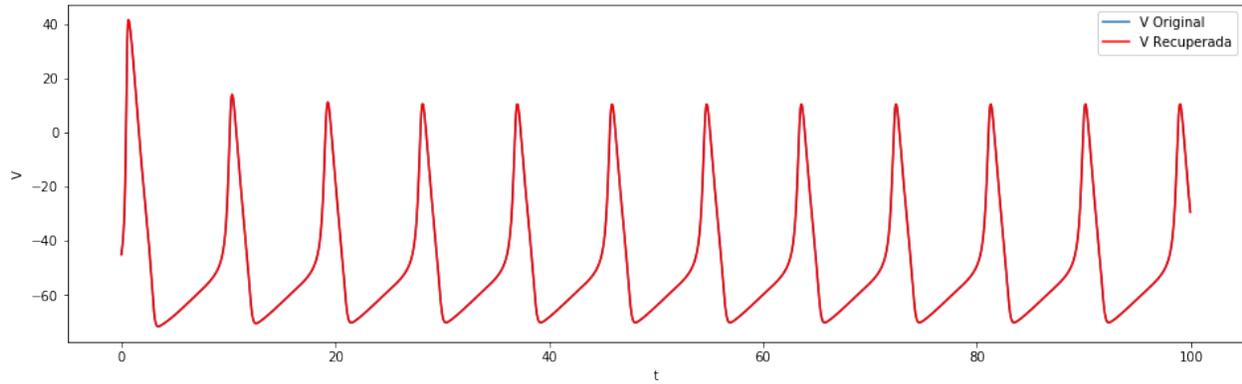


Figura 3.14: Voltaje $[mV]$ con respecto al tiempo en $[ms]$. Caso test cuyo error relativo es mayor al valor indicado. 2 capas de Encoder y 8 de Decoder

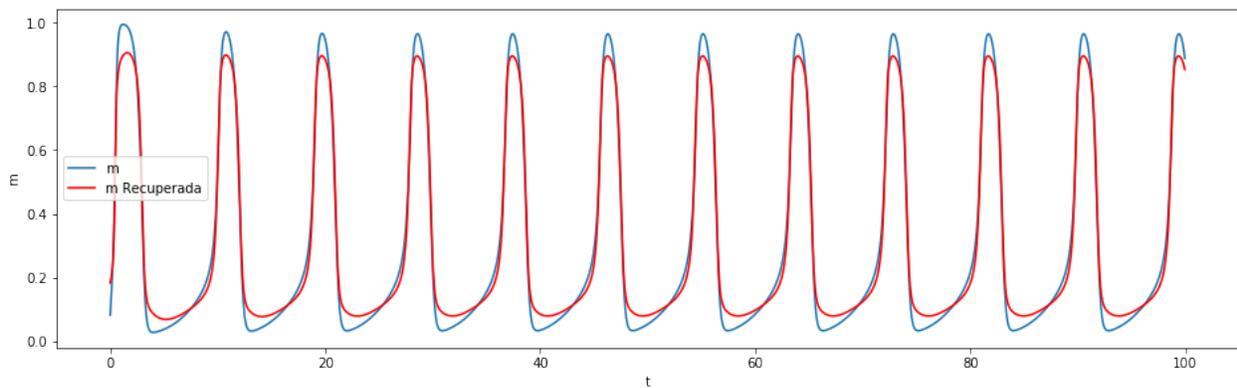


Figura 3.15: m con respecto al tiempo en $[ms]$. Caso test cuyo error relativo es mayor al valor indicado. 2 capas de Encoder y 8 de Decoder

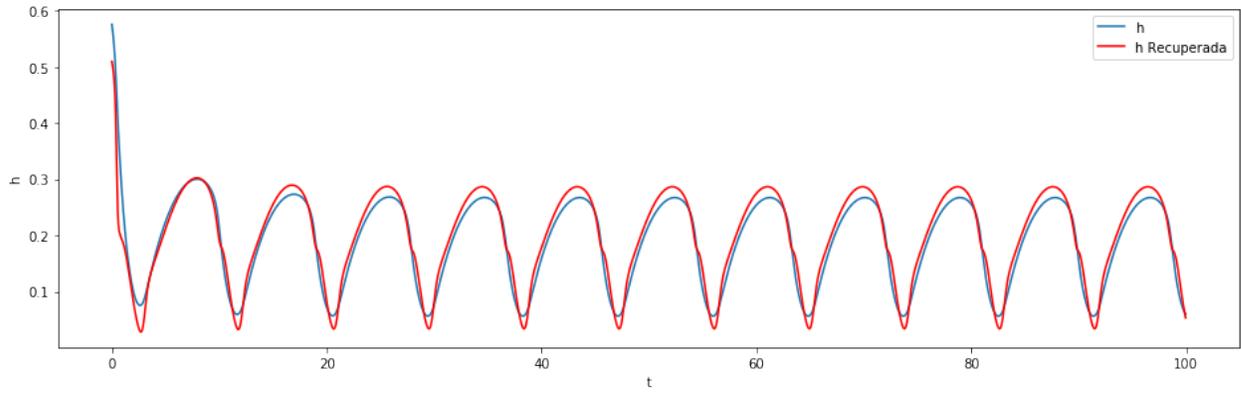


Figura 3.16: h con respecto al tiempo en $[ms]$. Caso test cuyo error relativo es mayor al valor indicado. 2 capas de Encoder y 8 de Decoder

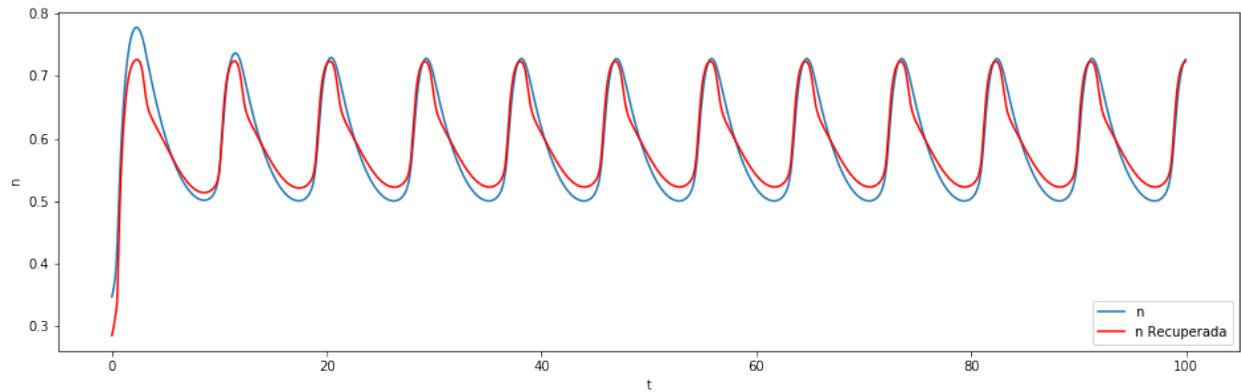


Figura 3.17: n con respecto al tiempo en $[ms]$. Caso test cuyo error relativo es mayor al valor indicado. 2 capas de Encoder y 8 de Decoder

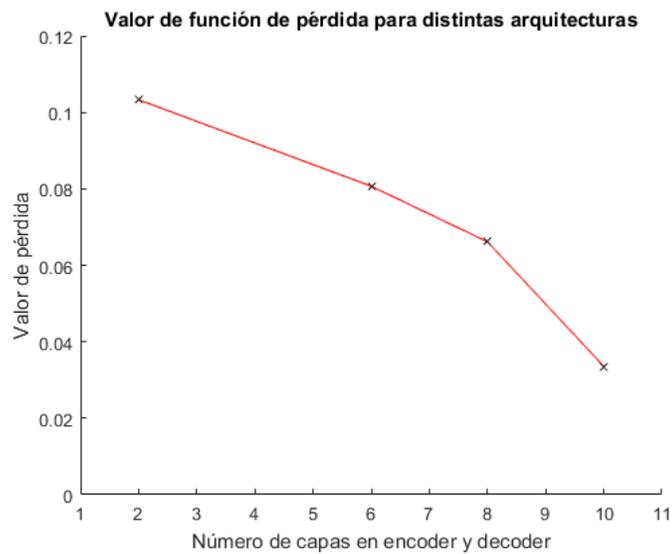


Figura 3.18: Valor de función de pérdida para distintas configuraciones de capas.

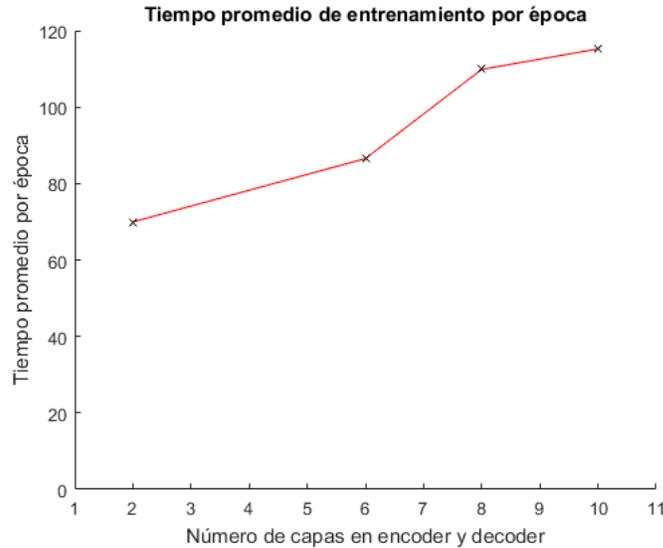


Figura 3.19: Tiempo de entrenamiento por época en promedio para distintas configuraciones de capas.

ten Tusscher-Panfilov

El modelo de ten Tusscher-Panfilov cuenta con 19 variables, resultando en 19 canales. La primera aproximación consiste en mantener la variable V , pero reducir el resto de las variables. El conjunto de entrenamiento cuenta con 7850 elementos, mientras que el test cuenta con 150. Para la creación de los datos, los parámetros que varían son algunas condiciones iniciales y la corriente aplicada.

Este método reconstruye las variables de compuerta, sin embargo, con las variables de corriente no ocurre lo mismo. La Figura (3.20) muestra la reconstrucción de una de las variables, en este caso, la de Sodio.

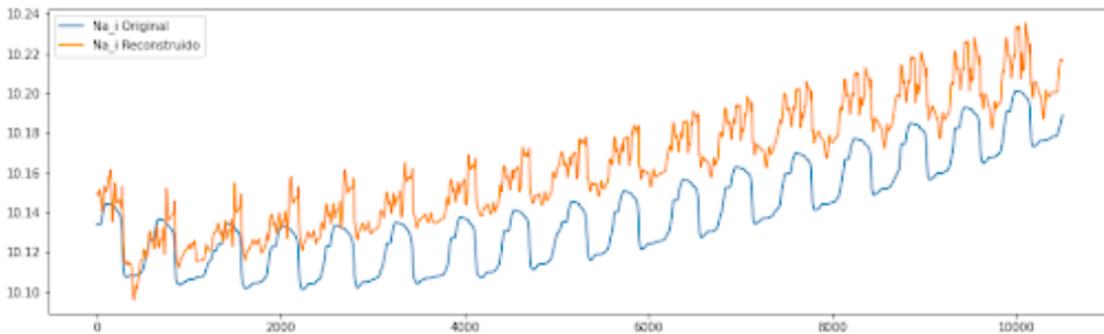


Figura 3.20: Caso de reconstrucción deficiente.

Para esto, una alternativa es mantener los términos de corriente V , K_i y Na_i , y reducir el resto de las variables. Siguiendo este esquema, tras un entrenamiento de 31261790 iteraciones en 100 épocas, se obtiene a una eficacia porcentual (calculada de la misma forma que en HH) de un 89,3% con 6 capas para encoder y 18 para decoder, con un tiempo promedio de 581,7

(aproximadamente 6,99 veces más que en el caso HH con la misma cantidad de capas), y un valor de pérdida de $1,524e - 02$, reduciendo de 19 a 13 variables, donde 10 son nuevas variables y 3 son las corrientes que se mantuvieron. En el Anexo, se muestran las nuevas variables (Figuras (4.35), (4.36), (4.37), (4.38), (4.39), (4.40), (4.41), (4.42), (4.43) y (4.44)). La reconstrucción de las variables asociadas a corrientes se muestran en la Figuras (3.21), (3.22), (3.23) y (3.24). La reconstrucción de variables de compuerta se encuentran en el Anexo.

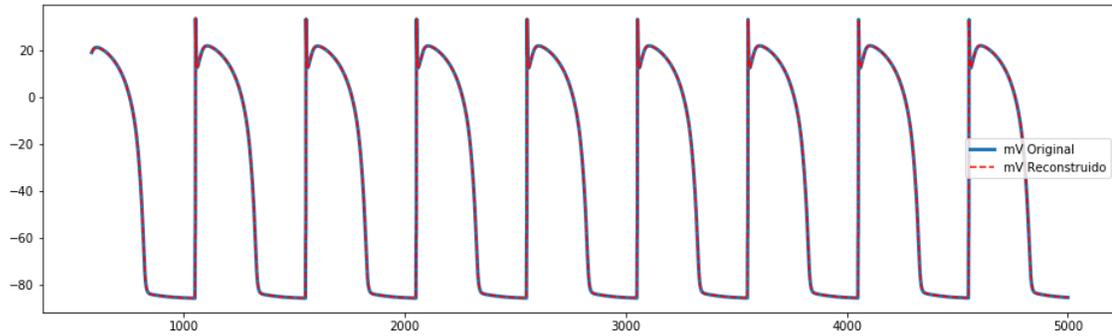


Figura 3.21: Reconstrucción de V .

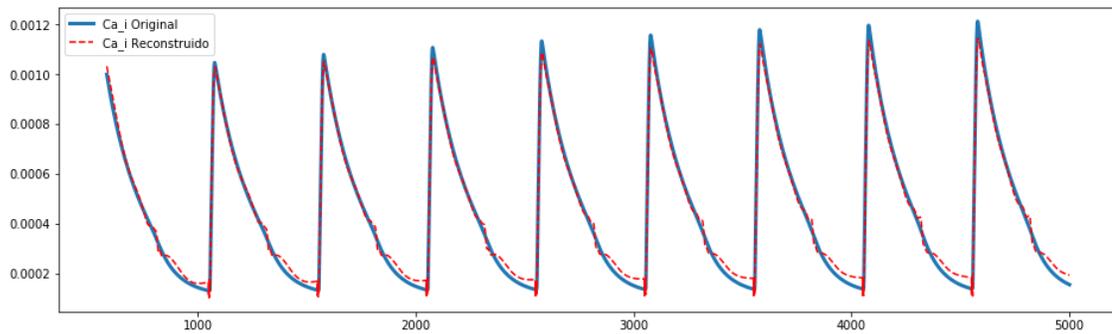


Figura 3.22: Reconstrucción de Ca_i .

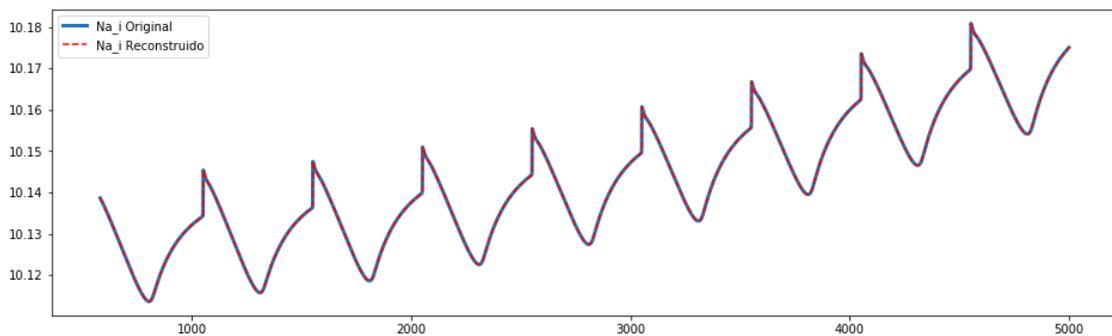


Figura 3.23: Reconstrucción de Na_i .

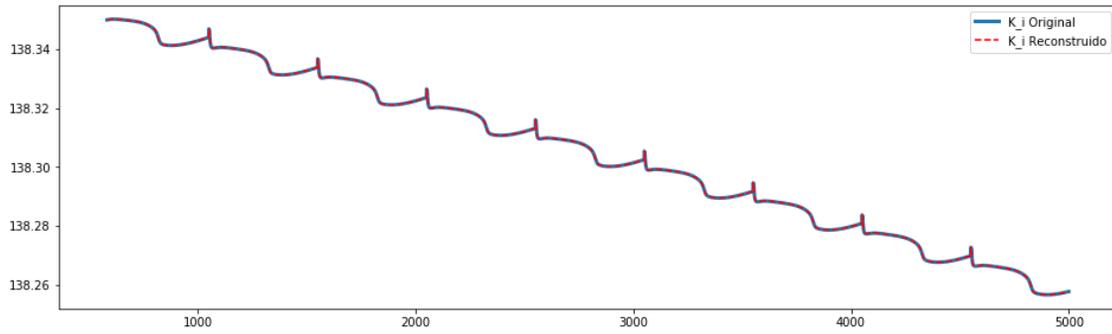


Figura 3.24: Reconstrucción de K_i .

En cuanto a la distancia en entrenamiento que hay entre la reconstrucción e *input*, medida a partir de la función de pérdida (en norma L^∞), en la Figura (3.25) se muestra cómo es esta para distintos tamaños de *code* para *POD* y *Autoencoders* usando el modelo de *ten Tusscher-Panfilov*. Por otro lado, la Figura (3.26), muestra el caso en el que se considera el término de error relativo máximo usando el conjunto test.

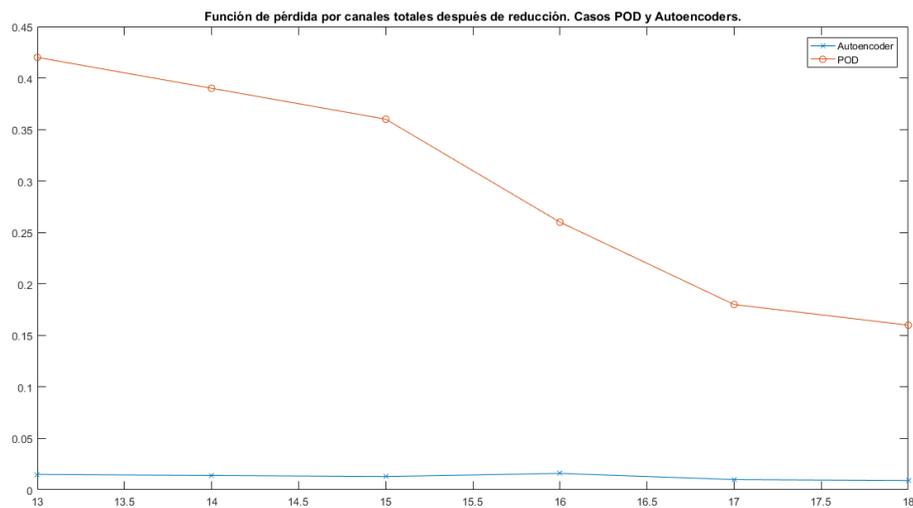


Figura 3.25: Función de pérdida para diferentes tamaño de reducción. POD y Autoencoders en *ten Tusscher-Panfilov*.

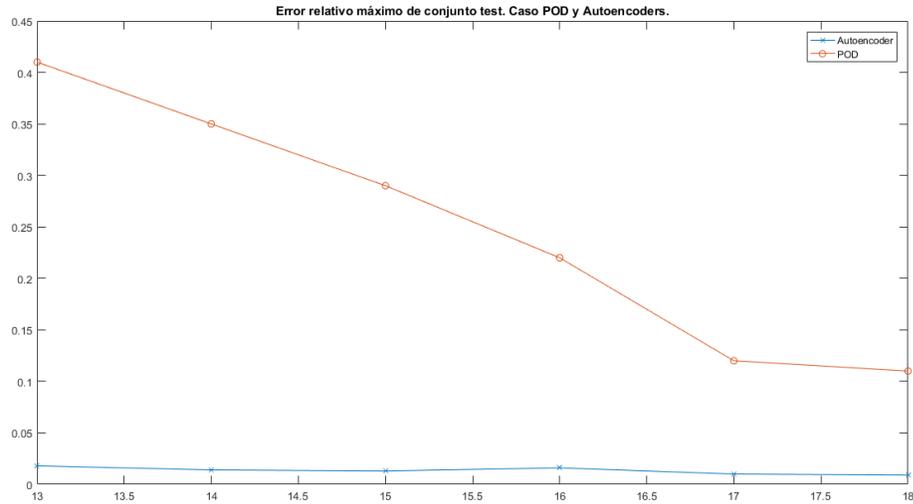


Figura 3.26: Error relativo máximo para diferentes tamaño de reducción usando el conjunto test. POD y Autoencoders en ten Tusscher-Panfilov.

Reconstrucción del sistema reducido

A priori, se espera que la reconstrucción provenga de un problema mal puesto, ya que, en casos simplificados, pequeñas perturbaciones de la identidad, provocan que la solución usando el nuevo lado derecho difiera a lo obtenido al reducir una solución original. Por ejemplo, suponiendo que tanto Encoder como Decoder son iguales a $2I$ y $0,5I$, respectivamente, la reconstrucción entrega el resultado esperado, esto se debe a que la composición de las funciones es la identidad. Sin embargo, si varía levemente uno de los valores, por ejemplo, cambiar el encoder por $1,9999I$, la composición de Encoder y Decoder sólo sería una aproximación. Al considerar 4 dígitos (o sea $1,9999I$) no se observan grandes diferencias (ver Figura (3.27)), sin embargo, al considerar 2, en otras palabras, aumentar un poco más el error, el resultado es diferente (ver Figura (3.28)).

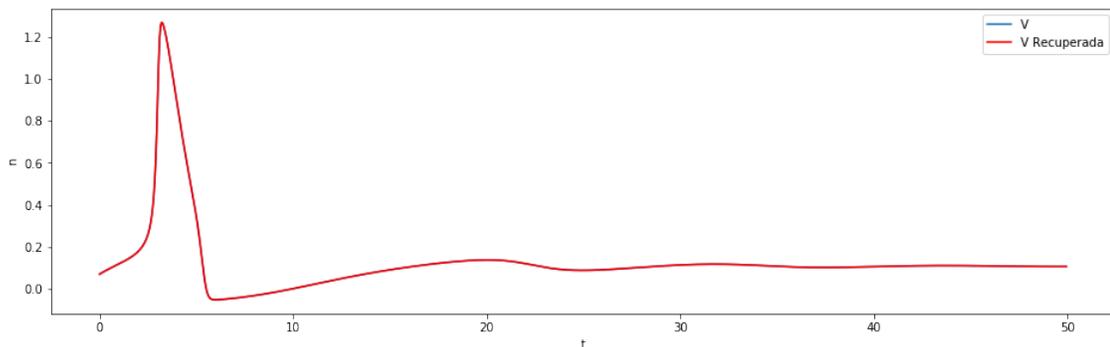


Figura 3.27: Voltaje original y reconstruido usando 4 dígitos.

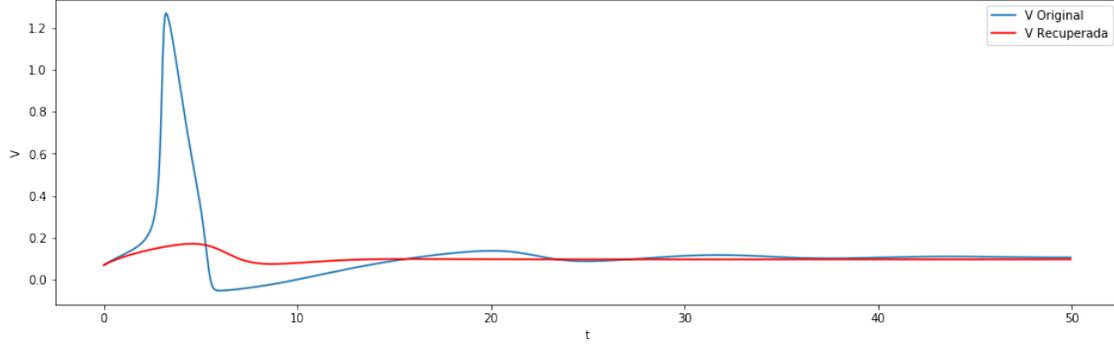


Figura 3.28: Voltaje original y reconstruido usando 2 dígitos.

Caso Hodgkin-Huxley

En el caso HH, se considera una red cuya arquitectura es de 1 capa por función y lineal afín en cada capa, de esta forma, $E_f/(u_b, l_b) = [E_{f1}, E_{f2}, E_{f3}, E_{f4}]/(u_b, l_b)$ cumple:

$$\begin{aligned}
E_{f1}/(V_{max} - V_{min}) &= -\varepsilon_1(G_K m^4 + G_{Na} n^3 h + G_l) \\
&\quad -\varepsilon_2 4G_k m^3 (m_{max} - m_{min})(V - V_K) \\
&\quad -\varepsilon_3 G_{Na} n^3 (h_{max} - h_{min})(V - V_{Na}) \\
&\quad -\varepsilon_4 3G_{Na} n^2 h (n_{max} - n_{min})(V - V_{Na}) \\
&\quad + O(\|\varepsilon\|^2) \\
&= -\varepsilon_1(36m^4 + 120n^3 h + 0,3) \\
&\quad -\varepsilon_2 135,5616m^3(V + 77) \\
&\quad -\varepsilon_3 58,428n^3(V - 50) \\
&\quad -\varepsilon_4 138,564n^2 h(V - 50) \\
&\quad + O(\|\varepsilon\|^2)
\end{aligned}$$

Se puede reescribir el término como:

$$E_{f1}/(V_{max} - V_{min}) = -\varepsilon \cdot \vec{v}_{f1} + O(\|\varepsilon\|^2)$$

Donde:

$$\vec{v}_{f1} = (36m^4 + 120n^3 h + 0,3; 135,5616m^3(V + 77); 58,428n^3(V - 50); 138,564n^2 h(V - 50))$$

Luego:

$$|E_{f1}/(V_{max} - V_{min})| \leq \|\varepsilon\|_2 \|\vec{v}_{f1}\|_2 + O(\|\varepsilon\|^2)$$

Las variables m , h y n , al ser variables de compuerta, son acotadas por 1, lo mismo ocurre con los términos de normalización multiplicados, mientras que la variable de la diferencia de voltaje V , se mantiene entre $-76mV$ y $30mV$, aproximadamente. De esta manera, se toman los valores que permitan maximizar la norma en módulo. Con esto, al reemplazar los términos anteriormente descritos se obtiene que:

$$\|\vec{v}_{f1}\|_2 \leq \|G_K + G_{Na} + G_l, 428G_K, -80G_{Na}, -240G_{Na}\|_2 \approx 34044,543$$

De esta forma, la primera componente de f es L_1 -Lipschitz, con $L_1 = 34044,543$.

Para ver como cambia el error en las otras componentes, el análisis inicial se realiza sobre la dinámica de m (para h y n es análogo). Las funciones α y β son continuas (salvo algunas excepciones en un punto, sin embargo, esa singularidad es reparable identificando el valor de la función como el límite) y diferenciables sobre el dominio considerado, previamente, para V .

En la segunda componente, se obtiene:

$$\begin{aligned} E_{f2}/(m_{max} - m_{min}) &= -\varepsilon_2(\alpha_m(V) + \beta_m(V)) - 86,9347\varepsilon_1 m(\alpha'_m(V) + \beta'_m(V)) + 86,9347\varepsilon_1 \alpha'_m(V) \\ &\quad + O(\|\varepsilon\|^2) \end{aligned}$$

De forma análoga, se determina que

$$\begin{aligned} E_{f3}/(h_{max} - h_{min}) &= -\varepsilon_3(\alpha_h(V) + \beta_h(V)) - 86,9347\varepsilon_1 h(\alpha'_h(V) + \beta'_h(V)) + 86,9347\varepsilon_1 \alpha'_h(V) \\ &\quad + O(\|\varepsilon\|^2) \end{aligned}$$

$$\begin{aligned} E_{f4}/(n_{max} - n_{min}) &= -\varepsilon_4(\alpha_n(V) + \beta_n(V)) - 86,9347\varepsilon_1 n(\alpha'_n(V) + \beta'_n(V)) + 86,9347\varepsilon_1 \alpha'_n(V) \\ &\quad + O(\|\varepsilon\|^2) \end{aligned}$$

Al igual que en el caso de E_{f1} , se pueden reescribir como:

$$E_{f2}/(m_{max} - m_{min}) = -\varepsilon \cdot \vec{v}_{f2} + O(\|\varepsilon\|^2)$$

$$E_{f3}/(h_{max} - h_{min}) = -\varepsilon \cdot \vec{v}_{f3} + O(\|\varepsilon\|^2)$$

$$E_{f4}/(n_{max} - n_{min}) = -\varepsilon \cdot \vec{v}_{f4} + O(\|\varepsilon\|^2)$$

Estos vectores se definen como:

$$\vec{v}_{f2} = (86,9347(m(\alpha'_m(V) + \beta'_m(V)) - \alpha'_m(V)), \alpha_m(V) + \beta_m(V), 0, 0)$$

$$\vec{v}_{f3} = (86,9347(h(\alpha'_h(V) + \beta'_h(V)) - \alpha'_h(V)), 0, \alpha_h(V) + \beta_h(V), 0)$$

$$\vec{v}_{f4} = (86,9347(n(\alpha'_n(V) + \beta'_n(V)) - \alpha'_n(V)), 0, 0, \alpha_n(V) + \beta_n(V))$$

Por otro lado, las derivadas se definen como:

- $\alpha'_m(V) = \frac{0,1(1-e^{-0,1(V+40)})-0,01(V+40)e^{-0,1*(V+40)}}{(1-e^{-0,1(V+40)})^2}$
- $\beta'_m(V) = -\frac{2}{9}e^{-(V+65)/18}$
- $\alpha'_h(V) = -\frac{0,07}{20}e^{-(V+65)/20}$
- $\beta'_h(V) = \frac{0,1e^{-0,1(V+35)}}{(1+e^{-0,1(V+35)})^2}$
- $\alpha'_n(V) = \frac{0,01(1-e^{-0,1(V+55)})-0,001(V+55)e^{-0,1(V+55)}}{(1-e^{-0,1(V+55)})^2}$
- $\beta'_n(V) = -\frac{0,125}{80}e^{-(V+65)/80}$

Como las funciones se definen en un compacto, α , β , α' y β' alcanzan máximos y mínimos en ese intervalo. De esta manera, a partir de los máximos y mínimos que alcanzan las funciones, se obtiene:

$$\|\vec{v}_{f2}\|_2 \leq \|(-33,0352, 7,1, 0, 0)\|_2 \approx 33,7896$$

$$\|\vec{v}_{f3}\|_2 \leq \|(2,1168, 0, 1, 0)\|_2 \approx 2,3411$$

$$\|\vec{v}_{f4}\|_2 \leq \|(0,1738, 0, 0, 0,8)\|_2 \approx 0,8187$$

De esta manera, se concluye que las siguientes 3 componentes de f , también son Lipschitz, donde sus constantes son $L_2 = 33,7896$, $L_3 = 2,3411$ y $L_4 = 0,8187$, respectivamente.

Se concluye que, en el caso HH, E_f también es Lipschitz sobre ese dominio E^0 , heredando las constantes de f . Por lo cual, sólo una de las componentes de E_f es contractiva cuando $\varepsilon \ll 1$, lo cual no garantiza que E_f contraiga los valores del error, incluso lo puede aumentar. Por ejemplo, considerando un error de $\varepsilon = (0,001, 0,001, 0,001, 0,001)$, para valores donde el voltaje es más alto (valores correspondientes al final de la fase de despolarización) la primera componente del error cumple que $|E_{f1}/(V_{max} - V_{min})| = 0,1980$. Hay otro máximo local que ocurre durante la hiperpolarización, en ese caso, considerando el mismo error, se obtiene $|E_{f1}/(V_{max} - V_{min})| = 0,1854$. Estos errores son grandes, ya que, la primera componente de $f(\cdot)/(u_b - l_b)$, varía entre $-0,8306$ y $3,3487$. Como los signos de cada término es distinto, se sufren variaciones cuando se considera la dirección en la cual se encuentra el error, sin embargo, cambiándola a una en la cual todos los términos comparten el signo, por ejemplo,

$$\varepsilon = (0,001, 0,001, -0,001, -0,001)$$

, no hay variación mayor en cuanto al valor máximo $|E_{f1}/(V_{max} - V_{min})| = 0,1797$ (también al final de la despolarización).

Para remediar esto y obtener errores cuyos valores sean del orden de 10^{-2} (o sea, que el porcentaje de error sea del orden de la unidad), necesitaríamos un ε cuya norma sea del orden de 10^{-4} , algo que a nivel numérico, no siempre se obtiene al usar autoencoders.

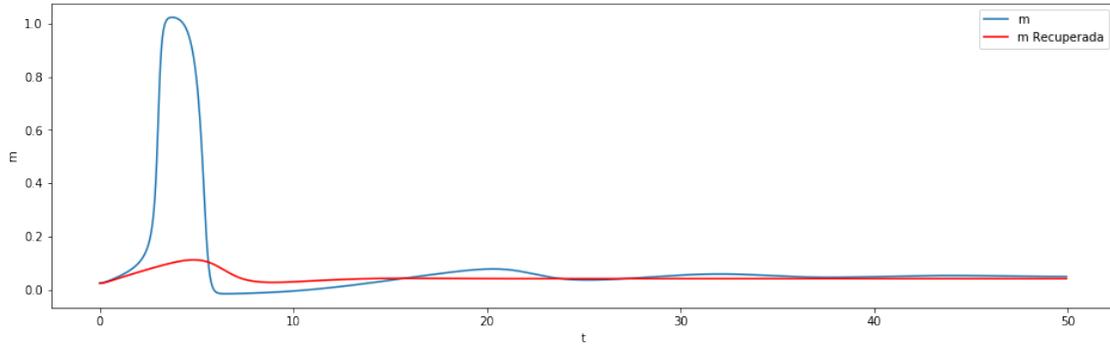


Figura 3.29: Ejemplo de m original y reconstruido.

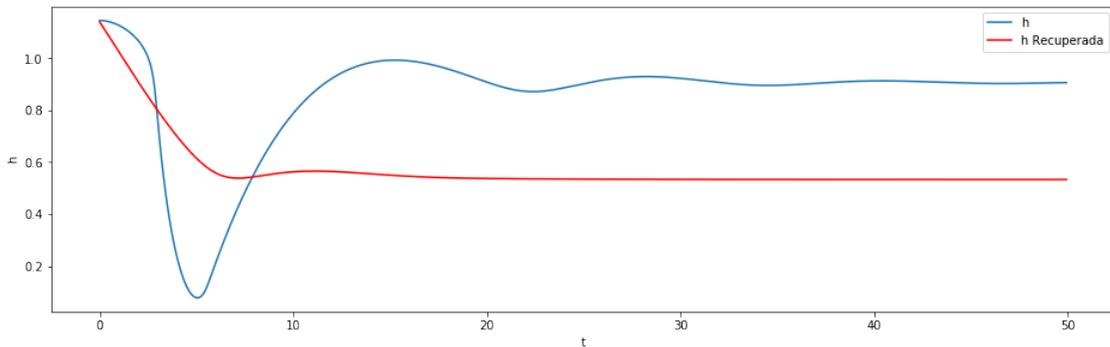


Figura 3.30: Ejemplo de h original y reconstruido.

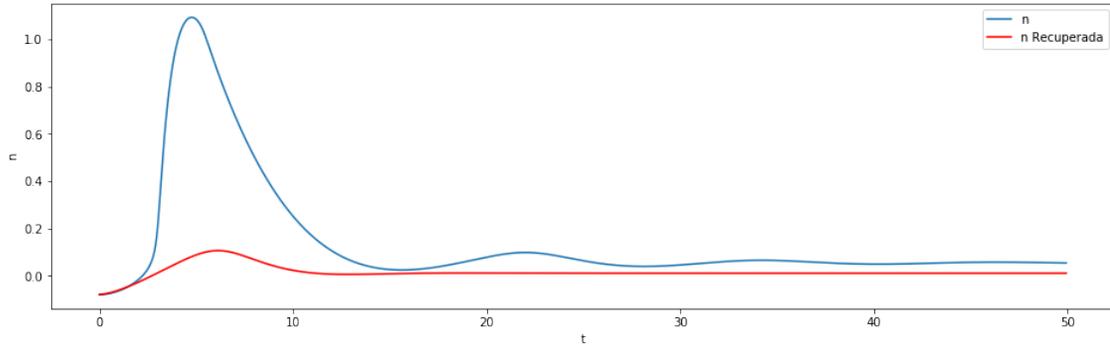


Figura 3.31: Ejemplo de n original y reconstruido.

Caso *ten Tusscher-Panfilov*

De igual manera que en HH, se puede calcular E_f para el caso de *ten Tusscher-Panfilov*, en este caso, dado a la gran cantidad de términos, algunos valores algebraicos se encuentran en el Anexo, por otro lado, para su primera posición, se obtiene que $L_1 = \|V_{v1}\|_2$, donde:

- $V_{v1}[1] = 48,7718$
- $V_{v1}[2] = 119,1847$
- $V_{v1}[3] = 0,1563$
- $V_{v1}[4] = 3,0557e + 05$
- $V_{v1}[5] = 3,8429$
- $V_{v1}[6] = 3,8749$
- $V_{v1}[7] = 3,8079$
- $V_{v1}[8] = 3,8059$
- $V_{v1}[9] = -2,1040e + 03$
- $V_{v1}[10] = -2,1048e + 03$
- $V_{v1}[11] = -4,7289e + 03$
- $V_{v1}[12] = 47,3635$
- $V_{v1}[13] = 99,7606$
- $V_{v1}[14] = 9,5520$
- $V_{v1}[15] = 47,7989$
- $V_{v1}[16] = 33,2443$
- $V_{v1}[17] = 19,5017$
- $V_{v1}[18] = 0$
- $V_{v1}[19] = 0$

Obteniendo $L_1 \approx 305625$, lo cual es mayor a 1, concluyendo que, al definir E_f en el caso de *ten Tusscher-Panfilov*, también se espera una amplificación de los valores del error.

Capítulo 4

Discusión

Tal como se muestra en el Capítulo Dos, el método de ajuste de nulclinas depende de dos pasos (ajuste polinomial y de velocidades), por lo cual, es difícil obtener una buena aproximación si una de estas falla. En el caso Polinomial, la no linealidad de las nulclinas originales, no permite obtener un buen ajuste con polinomios de grado bajo. La Figura (2.8), muestra como, por ejemplo, la nulclina asociada a V obtenida, presenta una curvatura menos pronunciada, provocando una forma de meseta en el potencial de acción que no posee la original, a esto se suma la lentitud que se advierte en el inicio de la despolarización que se equilibra con la alta velocidad en que ocurre la repolarización. Debido a esto, se procedió a realizar un ajuste polinomial por trozos. Un beneficio que presenta este método de polinomial por partes, tal como muestran las Figuras (2.13), (2.15), (2.17) y (2.19), es que permite obtener una curva parecida a la original, sin tener que requerir de términos de orden mayor, llegando a un modelo alternativo mejor ajustado con respecto al caso polinomial directo. Además, estas aproximaciones permiten variar las condiciones iniciales, sin tener que modificar tanto los valores en los polinomios como el de los ajustes de velocidad (ε). Sin embargo, esto depende del valor de las constantes que posee el modelo original. Por lo cual, si hay un cambio en las constantes, hay que ejecutar nuevamente el modelo original, realizar el ajuste polinomial y obtener los ajustes ε .

Queda propuesto trabajar una generalización del ajuste para distintos valores de constantes, una alternativa es analizar cómo ciertas perturbaciones en las constantes afectan los valores de ε en cada una de las regiones, como también el valor de los coeficientes polinomiales, quizás, de esta manera se pueda obtener alguna relación entre estos términos y las constantes del sistema original, evitando así, realizar un nuevo ajuste para cada cambio de constantes. Otra generalización interesante, es la posible presencia de más variables en el sistema original. Esto genera una complicación debido a que el espacio de fases es más complejo.

Un resultado obtenido en el caso de Morris-Lecar, es que el modelo equivalente no muestra una reducción en cuanto a tiempo de resolución (ver Tabla (2.3)), otro problema propuesto es lograr obtener un modelo equivalente que logre reducir el tiempo de cómputo del modelo original. Para el caso de Autoencoders, al analizar los errores medidos a través de MAE , $RMSE$ y la distancia L^∞ en el caso HH , se observa que el primero tiene valores más bajos, lo cual es esperable desde un punto de vista teórico. Sin embargo, como se busca reconstruir los valores de entrada, para evitar mayores errores se determinó usar la norma L^∞ como función

de pérdida para el resto del trabajo. De esta manera, al usar esta medida más robusta que las otras dos observadas, es mayor la penalización ante la aparición de puntos donde la curva original y de salida difieren en mayor medida. Un ejemplo es lo que se muestra en la Figura (3.5), la curva de salida tras el entrenamiento, usando como función de pérdida la norma L^∞ , tiende a ajustarse más en los máximos que la curva proveniente del caso L^2 .

Sobre la reconstrucción, en el caso HH se logra una reducción a través de un operador no lineal. Los tiempos de entrenamiento promedio por época, muestran un aumento a medida que se agregan capas, sin embargo, la función de pérdida, luego de la última iteración, baja con respecto a la cantidad de capas.

Para un modelo más complejo, como lo es el de *ten Tusscher-Panfilov*, el conjunto de test muestra una buena reconstrucción, sin embargo, es necesario tener una arquitectura más compleja, sumado a un régimen de entrenamiento mayor.

Un primer problema propuesto es mejorar la reconstrucción de los términos sin la necesidad de tener que añadir aún más capas. Una idea es seguir la idea del primer método y ajustar las curvas obtenidas tras pasar el input por la red, con el original. Para esto, se podría calcular las cotas superiores e inferiores individuales del sistema a reducir, como no se conocen a priori, ya que, una meta es lograr evitar resolver el sistema, se calculan sus cotas esperadas (u_{be} y l_{be}). Una posibilidad es que se puedan sólo conociendo su condición inicial y parámetros, sin embargo es un trabajo propuesto. En caso de lograrlo, se lograrían reconstrucciones aún más ajustadas si en vez de considerar

$$n^{-1}(\Psi(\Phi(n(y))))$$

Se considera:

$$(\Psi(\Phi(n(y))) - \Psi(\Phi(n(y)))_{min}) // (\Psi(\Phi(n(y)))_{max} - \Psi(\Phi(n(y)))_{min}) \circ (u_{be} - l_{be}) + l_{be}$$

Otro problema que suele presentarse es lograr un análisis de las variables de la reducción, esto se debe al uso de funciones de paso no lineales en cada una de las capas. Sin embargo, se pueden observar fuertes influencias de algunas variables originales en las nuevas. Como es el caso de la variable x_{r2} (Figura (4.1)) cuyo comportamiento es similar a ω_6 (Figura (4.2)). Otros casos similares son: Variable m (Figura (4.51)) con un comportamiento similar a ω_3 (Figura (4.37)), o j (Figura (4.53)) con ω_4 (Figura (4.38)). Los otros casos no poseen un término similar en el original, sin embargo, algunos poseen fuertes influencias, como es el caso de la variable ω_{10} (Figura (4.44)), que posee un comportamiento similar a x_{r2} , pero con una depresión en la meseta que va disminuyendo en cada ciclo.

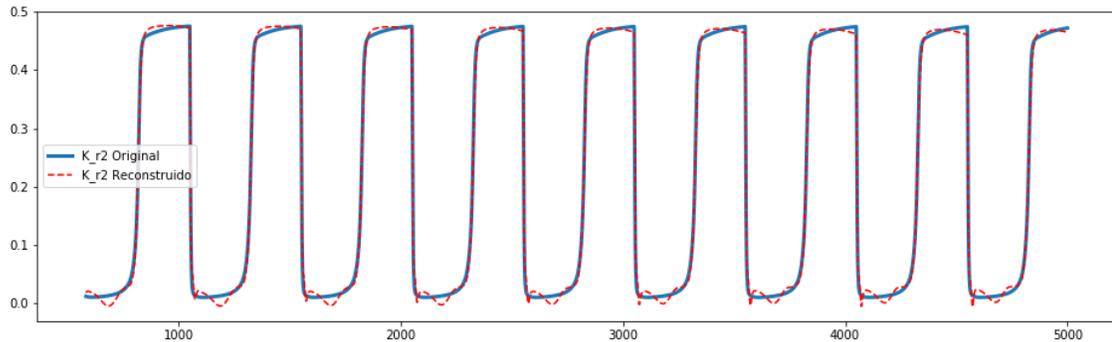


Figura 4.1: Variable x_{r2} .

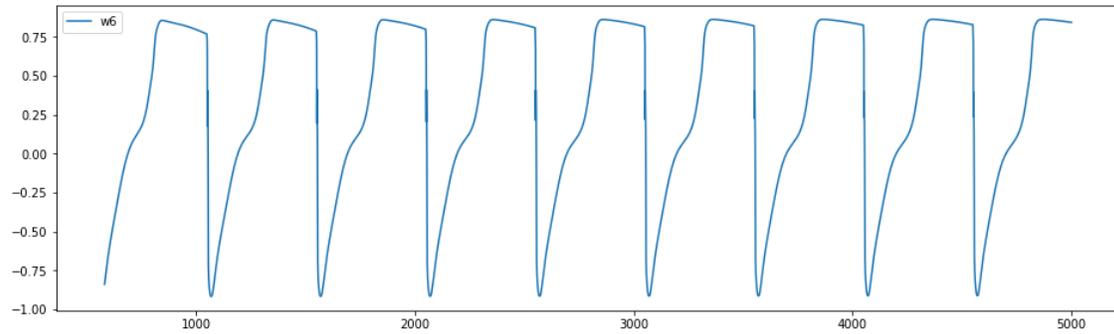


Figura 4.2: Variable ω_6 .

También queda abierto el poder disminuir el error de reconstrucción E_f , del nuevo sistema para z . Una alternativa es considerar un término de peso extra que dependa de la constante de Lipschitz L sobre la función de pérdida del esquema de entrenamiento, de esta forma, se busca que la recuperación de los datos, vía *Decoder*, considere el error de la reconstrucción del sistema reducido. Otra opción es considerar una norma equivalente a la norma \mathcal{C}^1 : $\|h\|_{\mathcal{C}^1} = \|h\|_{\infty} + \|h'\|_{\infty}$, esto podría mejorar la aproximación a nivel de la derivada, sin embargo, si la igualdad en derivada tiene un peso muy alto, podría verse afectada la reconstrucción de los datos originales. Finalmente, el uso de otras redes, de forma paralela, que aprendan los valores numéricos de la derivadas, también puede ser una opción viable.

Conclusión

El presente estudio mostró dos metodologías que trabajan con las soluciones provenientes de modelos introducidos en el primer capítulo.

El primer método logró encontrar un modelo equivalente mediante una representación polinomial del modelo original, mientras que el segundo muestra que es posible alcanzar, a través de una menor cantidad de variables, la información suficiente para lograr una reconstrucción satisfactoria de las variables originales, esto gracias al entrenamiento de una red neuronal, cuya arquitectura se especializa en el proceso de reconstrucción.

Esta investigación requirió el uso de diferentes tipos de software, lo cual implicó un desafío en el sentido de aprender a utilizarlos correctamente, una descripción detallada de estas herramientas computacionales se encuentra en el Anexo.

Finalmente, es importante agregar que aún se puede ahondar en cada uno de los temas, como por ejemplo, obtener un modelo equivalente más eficiente a través del ajuste de nulclinas o lograr determinar el sistema reducido, obtenido a partir de la proyección vía *Encoder*.

Bibliografía

- [1] Richard FitzHugh. Impulses and physiological states in theoretical models of nerve membrane. *Biophysical Journal*, 1(6):445–466, July 1961.
- [2] David Hartman and Lalit Mestha. A deep learning framework for model reduction of dynamical systems. pages 1917–1922, 08 2017.
- [3] A. L. Hodgkin and A. F. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology*, 117(4):500–544, August 1952.
- [4] Daniel E. Hurtado and Duvan Henao. Gradient flows and variational principles for cardiac electrophysiology: Toward efficient and robust numerical simulations of the electrical activity of the heart. *Computer Methods in Applied Mechanics and Engineering*, 273:238–254, May 2014.
- [5] Gaetan Kerschen, Jean Claude Golinval, Alexander F. Vakakis, and Lawrence A. Bergman. The method of proper orthogonal decomposition for dynamical characterization and order reduction of mechanical systems: An overview. *Nonlinear Dynamics*, 41(1-3):147–169, August 2005.
- [6] Andreas Kirsch. *An Introduction to the Mathematical Theory of Inverse Problems*. Springer New York, 2011.
- [7] Sanjay Lall, Jerrold E. Marsden, and Sonja Glavaški. *A subspace approach to balanced truncation for model reduction of nonlinear control systems*, volume 12. 2002.
- [8] Kookjin Lee and Kevin Carlberg. Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. 12 2018.
- [9] C. Morris and H. Lecar. Voltage oscillations in the barnacle giant muscle fiber. *Biophysical Journal*, 35(1):193–213, July 1981.
- [10] Luca F. Pavarino Piero Colli Franzone and Simone Scacchi. Mathematical cardiac electrophysiology. *Springer*, 13, 2014.
- [11] Eugene B. Postnikov and Olga V. Titkova. A correspondence between the models of hodgkin-huxley and FitzHugh-nagumo revisited. *The European Physical Journal Plus*, 131(11), November 2016.

- [12] Alfio Quarteroni and Gianluigi Rozza. *Reduced Order Methods for Modeling and Computational Reduction*. 01 2014.
- [13] C. W. Rowley. Model reduction for fluids, using balanced proper orthogonal decomposition. *International Journal of Bifurcation and Chaos*, 15(03):997–1013, 2005.
- [14] K. H. W. J. ten Tusscher and A. V. Panfilov. Alternans and spiral breakup in a human ventricular tissue model. *American Journal of Physiology-Heart and Circulatory Physiology*, 291(3):H1088–H1100, September 2006.

Anexo

Valores numéricos trabajados

Hodgkin-Huxley

Constante	Valor
V_{Na}	50
V_K	-77
V_l	-54,4
G_{Na}	120
G_k	36
\bar{g}_l	0,3
C_m	1

Tabla 4.1: Valores de las constantes usadas en el modelo de Hodgkin-Huxley.

Función	Expresión
$\alpha_m(V)$	$0,1 \frac{V+40}{1-e^{-(V+40)/10}}$
$\beta_m(V)$	$4e^{-(V+65)/18}$
$\alpha_h(V)$	$0,07e^{-(V+65)/20}$
$\beta_h(V)$	$\frac{1}{1+e^{-(V+35)/10}}$
$\alpha_n(V)$	$0,01 \frac{V+55}{1-e^{-(V+55)/10}}$
$\beta_n(V)$	$0,125e^{-(V+65)/80}$

Tabla 4.2: Funciones α y β , utilizadas en 1.15, 1.16 y 1.17.

Morris-Lecar

Constante	Valor
V_{Ca}	120
V_K	-84
V_l	-60
G_{Ca}	4,4
G_k	8
\bar{g}_l	2
C_m	1
ε	0,04

Tabla 4.3: Valores de las constantes usadas en el modelo de Morris-Lecar.

Función	Expresión
$m_\infty(V)$	$0,5(1 + \tanh((V + 1,2)/18))$
$\omega_\infty(V)$	$0,5(1 + \tanh((V - 2)/30))$
$\tau(V)$	$\frac{1}{\cosh((V-2)/(60))}$

Tabla 4.4: Funciones m_∞ , ω_∞ y τ , usadas en (1.18) y (1.19).

FitzHugh-Nagumo

Constante	Valor
ν	0,35
α	0,12
V_l	1
δ	0,06
a	0
b	0,55

Tabla 4.5: Valores de las constantes usadas en el modelo de FitzHugh-Nagumo.

ten Tusscher-Panfilov

Corrientes totales usadas en (1.27) están descritas en la Tabla (4.6). Por otro lado, el detalle de los términos de corriente usados, tanto en el sistema como en la Tabla (4.6), se encuentran en la Tabla (4.7).

Corriente total	Expresión
I_{tCa}	$I_{CaL} + I_{CaB} + I_{pCa} - 2I_{NaCa}$
I_{tNa}	$I_{Na} + I_{NaB} + 3I_{NaK} + 3I_{NaCa}$
I_{tK}	$I_{K1} + I_{To} + I_{Kr} + I_{Ks} + I_{pK} - 2I_{NaK}$

Tabla 4.6: Corrientes totales usadas en (1.27).

Función	Expresión
I_{CaL}	$g_{CaL} \cdot d \cdot f \cdot f_2 \cdot f_{Cass} \cdot 4(V - 15) \cdot FFTR \cdot \frac{0,25 \cdot Ca_{ss} e^{(2(V-15) \cdot FTR)} - Ca_o}{e^{(2(V-15) \cdot FTR)} - 1}$
I_{CaB}	$g_{bCa}(V - E_{Ca})$
I_{pCa}	$g_{pCa} \frac{Ca_i}{Ca_i + K_{pCa}}$
I_{Na}	$g_{Na} m^3 h_j (V - E_{Na})$
I_{K1}	$g_{K1} I_{K1\infty} \sqrt{K_o/5,4} (V - E_K)$
I_{pK}	$g_{pK} (V - E_K) / (1 + e^{(25-V)/5,98})$
I_{Kr}	$g_{Kr} \sqrt{K_o/5,4} x_{r1} x_{r2} (V - E_K)$
I_{Ks}	$g_{Ks} x_s^2 (V - E_{Ks})$
I_{NaB}	$g_{NaB} (V - E_{Na})$
I_{NaCa}	$K_{NaCa} \frac{e^{g V FTR} Na_i^3 Ca_o - e^{(g-1)V FTR} Na_o^3 Ca_i a}{(Km_{Na_i}^3 + Na_o^3)(Km_{Ca} + Ca_o)(1 + K_{sat} e^{(g-1)V FTR})}$
I_{NaK}	$P_{NaK} \frac{K_o}{(K_o + K_{mk})} \frac{Na_i}{(Na_i + K_{mNa})} \frac{1}{(1 + 0,1245 e^{-0,1V FTR} + 0,0353 e^{-V FTR})}$
I_{To}	$g_{To} r s (V - E_K)$
I_{leak}	$V_{leak} (Ca_{sr} - Ca_i)$
I_{rel}	$V_{rel} \Theta (Ca_{sr} - Ca_{ss})$
I_{up}	$\frac{V_{max}^{up}}{(1 + K_{up}^2 / Ca_i^2)}$
I_{xfer}	$V_{xfer} (Ca_{ss} - Ca_i)$

Tabla 4.7: Otras corrientes usadas en el sistema y en la Tabla (4.6).

Función	Expresión
α_d	$\frac{1,4}{1+e^{((-35-V)/13)}} + 0,25$
β_d	$\frac{1,4}{1+e^{((V+5)/5)}}$
γ_d	$\frac{1}{1+e^{((50-V)/20)}}$
τ_d	$\alpha_d\beta_d + \gamma_d$
\bar{d}_∞	$\frac{1}{1+e^{((-8-V)/7,5)}}$
τ_{f2}	$\frac{0,67}{1+e^{((V+35)/7)}} + 0,33$
$f_{2\infty}$	$562 e^{-(V+27)^2/240} + \frac{31}{1+e^{(25-V)/10}} + \frac{80}{1+e^{(V+30)/10}}$
τ_{fCass}	$\frac{0,6}{1+(C_{ass}/0,05)^2} + 0,4$
$fCass_\infty$	$\frac{80}{1+(C_{ass}/0,05)^2} + 2$
τ_f	$\frac{1}{1+e^{((V+20)/7)}}$
f_∞	$1102,5 e^{-(V+27)^2/225} + \frac{200}{1+e^{(13-V)/10}} + \frac{180}{1+e^{(V+30)/10}} + 20$
α_h	$0,057 e^{-((V+80)/6,8)} \mathbb{1}_{\{V < -40\}}$
β_h	$(2,7 e^{0,079V} + 310000 e^{0,3485V}) \mathbb{1}_{\{V < -40\}} + \frac{0,77}{0,13(1+e^{(V+10,66)/-11,1})} \mathbb{1}_{\{V \geq -40\}}$
τ_h	$\frac{1}{\alpha_h + \beta_h}$
h_∞	$\frac{1}{(1+e^{((V+71,55)/7,43)})^2}$
α_j	$(-25428 e^{0,2444V} - 6,948 \cdot 10^{-6} e^{-0,04391V} \frac{(V+37,78)}{1+e^{0,311(V+79,23)}}) \mathbb{1}_{\{V < -40\}}$
β_j	$0,02424 \frac{e^{-0,01052V}}{1+e^{-0,1378(V+40,14)}} \mathbb{1}_{\{V < -40\}} + 0,6 \frac{e^{0,057V}}{1+e^{-0,1(V+32)}} \mathbb{1}_{\{V \geq -40\}}$
τ_j	$\frac{1}{\alpha_j + \beta_j}$
j_∞	$\frac{1}{(1+e^{((V+71,55)/7,43)})^2}$
α_m	$\frac{1}{1+e^{(-60-V)/5}}$
β_m	$\frac{0,1}{1+e^{(V+35)/5}} + \frac{0,1}{1+e^{(V-50)/200}}$
τ_m	$\alpha_m\beta_m$
m_∞	$\frac{1}{(1+e^{((-V-56,86)/9,03)})^2}$
α_{IK1}	$\frac{0,1}{1+e^{0,06(V-E_K-200)}}$
β_{IK1}	$\frac{3,0e^{0,0002(V-E_K+100)} + e^{0,1(V-E_K-10)}}{1+e^{-0,5(V-E_K)}}$
$IK1_\infty$	$\frac{\alpha_{IK1}}{\alpha_{IK1} + \beta_{IK1}}$
$\alpha_{x,r1}$	$\frac{450}{1+e^{(-45-V)/10}}$

$\beta_{x_{r1}}$	$\frac{6}{1+e^{(V+30)/11,5}}$
$\tau_{x_{r1}}$	$\alpha_{x_{r1}}\beta_{x_{r1}}$
$x_{r1\infty}$	$\frac{1}{1+e^{(-26-V)/7}}$
$\alpha_{x_{r2}}$	$\frac{3}{1+e^{(-60-V)/20}}$
$\beta_{x_{r2}}$	$\frac{1,12}{1+e^{(V-60)/20}}$
$\tau_{x_{r2}}$	$\alpha_{x_{r2}}\beta_{x_{r2}}$
$x_{r2\infty}$	$\frac{1}{1+e^{(V+88)/24}}$
α_{x_s}	$\frac{1400}{\sqrt{1+e^{(5-V)/6}}}$
β_{x_s}	$\frac{1}{1+e^{(V-35)/15}}$
τ_{x_s}	$\alpha_{x_s}\beta_{x_s} + 80$
$x_{s\infty}$	$\frac{1}{1+e^{(-V-5)/14}}$
τ_r	$9,5 e^{-(V+40)^2/1800} + 0,8$
r_∞	$\frac{1}{1+e^{(20-V)/6}}$
τ_s	$1000 e^{-(V+67)^2/1000} + 8$ (Endocardio). $85 e^{-(V+45)^2/320} + \frac{5}{1+e^{(20-V)/5}} + 3$ (Epicardio y Mid-Miocardio).
s_∞	$\frac{1}{1+e^{(V+28)/5}}$ (Endocardio). $\frac{1}{1+e^{(V+20)/5}}$ (Epicardio y Mid-Miocardio).
Θ	$\frac{k_1 C a_{ss}^2 R_{prime}}{(k_3 + k_1 C a_{ss}^2)}$
RTF	$1000 \frac{RT}{F}$
FTR	$\frac{F}{1000RT}$
$FFTR$	$\frac{F^2}{(1000 \cdot RT)}$
$kcasr$	$max_{sr} - \frac{max_{sr} - min_{sr}}{1+(EC/Ca_{sr})^2}$
k_1	$\frac{k1_{prime}}{kcasr}$
k_2	$k2_{prime} kcasr$

Tabla 4.8: Funciones usadas en el sistema de ecuaciones de ten Tusscher-Panfilov.

Constante	Valor	Unidad de medida
F	96485,3415	$[C/m \text{ mol}]$
R	8,314472	$[J/mol/K]$
T	310	$[K]$
C_m	0,185	$[\mu F]$
V_c	0,016404	$[\mu m^3]$
$gCaL$	3,98e - 05	$[gm^5/s^5/A^2]$

g_{bCa}	0,000592	$[mS/\mu F]$
g_{pCa}	0,1238	$[A/F]$
g_{Na}	14,838	$[mS/\mu F]$
g_{K1}	5,405	$[mS/\mu F]$
g_{pK}	0,0146	$[mS/\mu F]$
g_{Kr}	0,153	$[mS/\mu F]$
g_{Ks}	0,392 (endocardio y epicardio). 0,098 (en mid-miocardio).	$[mS/\mu F]$
g_{NaB}	0,00029	$[mS/\mu F]$
g_{To}	0,073 (endocardio). 0,294 (en mid-miocardio y epicardio).	$[mS/\mu F]$
K_{pCa}	0,0005	$[mM]$
K_{NaCa}	1000	$[A/F]$
Km_{Ca}	1,38	$[mM]$
Km_{Nai}	87,5	$[mM]$
Km_{Na}	40	$[mM]$
Km_K	1	$[mM]$
P_{NaK}	2,724	$[A/F]$
V_{sr}	0,001094	$[\mu m^3]$
V_{ss}	$5,468e - 5$	$[\mu m^3]$
Bu_{fc}	0,2	$[nM]$
KBu_{fc}	0,001	$[nM]$
Bu_{fss}	0,4	$[nM]$
KBu_{fss}	0,00025	$[nM]$
$Bu_{f_{sr}}$	10	$[nM]$
$KBu_{f_{sr}}$	0,3	$[nM]$
K_{up}	0,00025	$[nM]$
EC	1,5	$[nM]$
k_3	0,06	$[mS/\mu F]$
k_4	0,06	$[mS/\mu F]$
V_{leak}	0,00036	$[mS/\mu F]$
V_{rel}	0,102	$[mS/\mu F]$
V_{xfer}	0,0038	$[mS/\mu F]$
$V_{max_{up}}$	0,006375	$[mol/m^3/ms]$

Tabla 4.9: Valores de las constantes físicas usadas en el modelo de ten Tusscher-Panfilov.

Potencial	Valor	Unidad de medida
E_{Ca}	$0,5 \cdot RTF \ln(Ca_o/Ca_i)$	$[mV]$
E_K	$RTF \ln(K_o/K_i)$	$[mV]$
E_{Ks}	$RTF \ln((K_o + P_{K-Na}Na_o)/(K_i + P_{K-Na}Na_i))$	$[mV]$
E_{Na}	$RTF \ln(Na_o/Na_i)$	$[mV]$

Tabla 4.10: Potenciales de membrana (o de *Nerst*) usadas en el modelo de ten Tusscher-Panfilov.

Potencial	Valor	Unidad de medida
K_o	5,4	$[mM]$
Na_o	140	$[mM]$
Ca_o	2	$[mM]$

Tabla 4.11: Concentraciones externas usadas en el modelo de ten Tusscher-Panfilov.

Potencial	Valor
K_{sat}	0,1
a	2,5
g	0,35
max_{sr}	2,5
min_{sr}	1
$k1_{prime}$	0,15
$k2_{prime}$	0,045
Ca_i^{bufc}	$1/(1 + Buf_c K Buf_c / (Ca_i + K Buf_c)^2)$
Ca_{ss}^{bufss}	$1/(1 + Buf_{ss} K Buf_{ss} / (Ca_{ss} + K Buf_{ss})^2)$
Ca_{sr}^{bufsr}	$1/(1 + Buf_{sr} K Buf_{sr} / (Ca_{sr} + K Buf_{sr})^2)$

Tabla 4.12: Otras constantes usadas en el modelo de ten Tusscher-Panfilov.

Resultados de reducción y reconstrucción sobre otras variables

Reducción vía POD

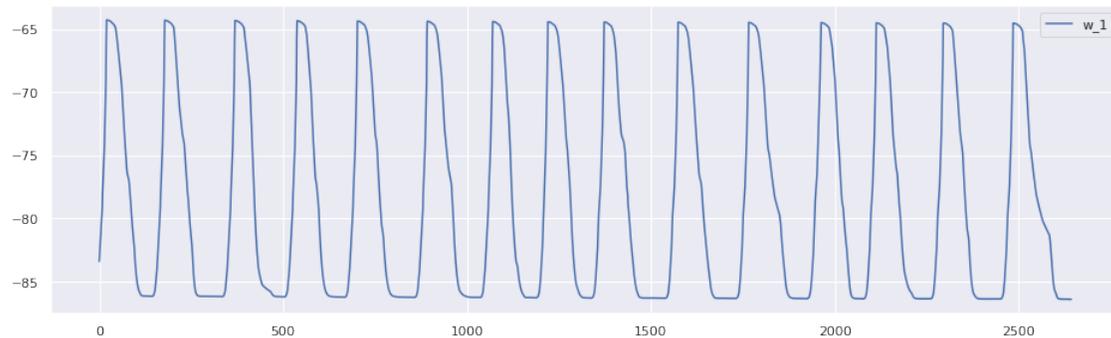


Figura 4.3: Variable ω_1 de reducción. Usando POD.

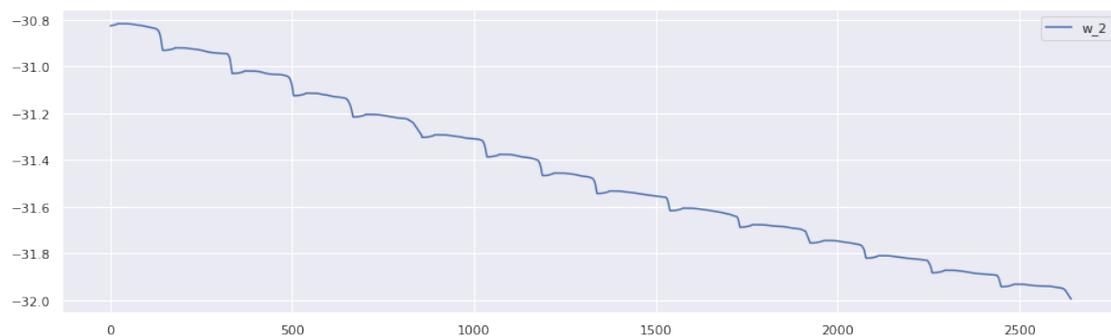


Figura 4.4: Variable ω_2 de reducción. Usando POD.

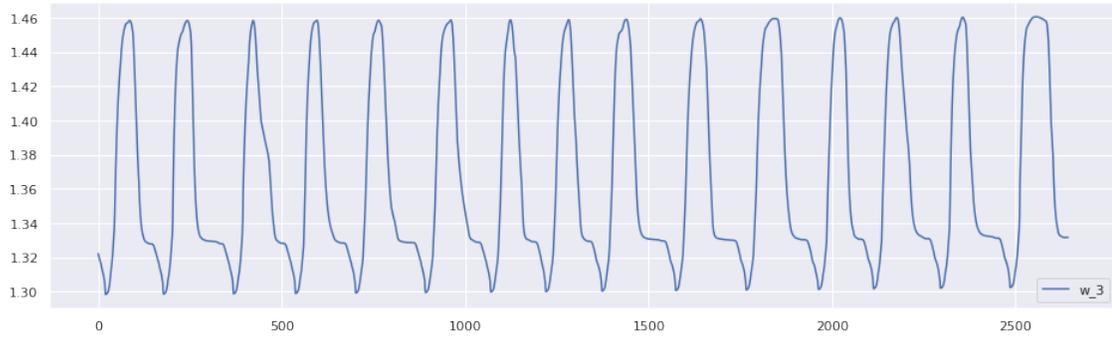


Figura 4.5: Variable ω_3 de reducción. Usando POD.

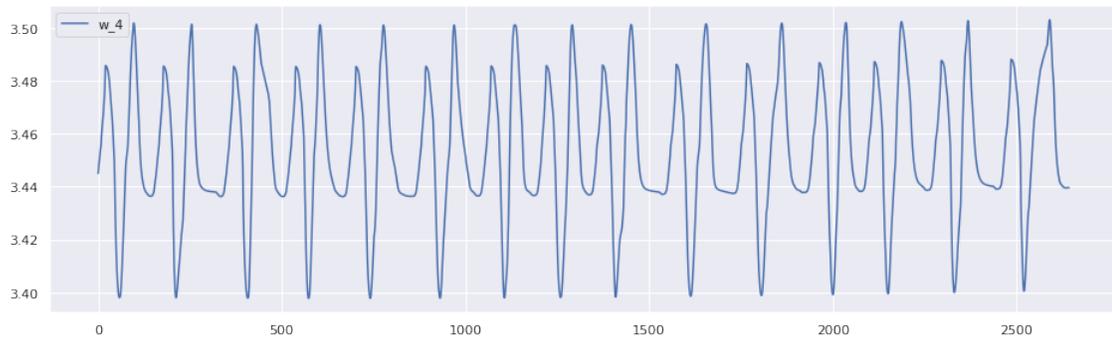


Figura 4.6: Variable ω_4 de reducción. Usando POD.

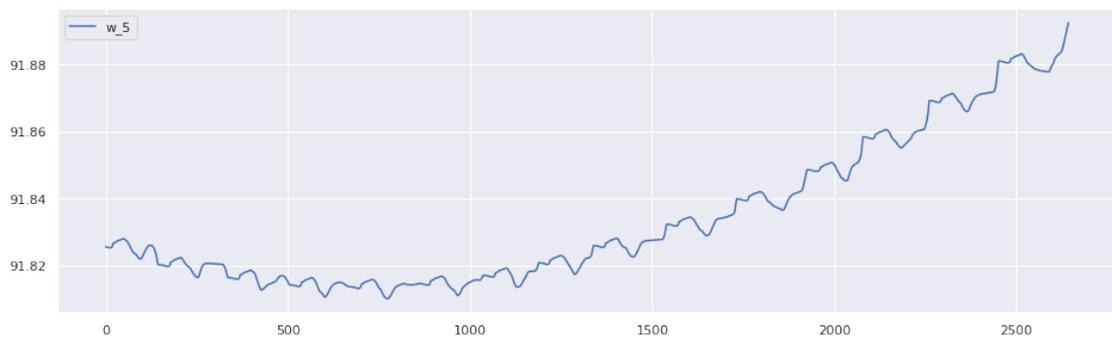


Figura 4.7: Variable ω_5 de reducción. Usando POD.

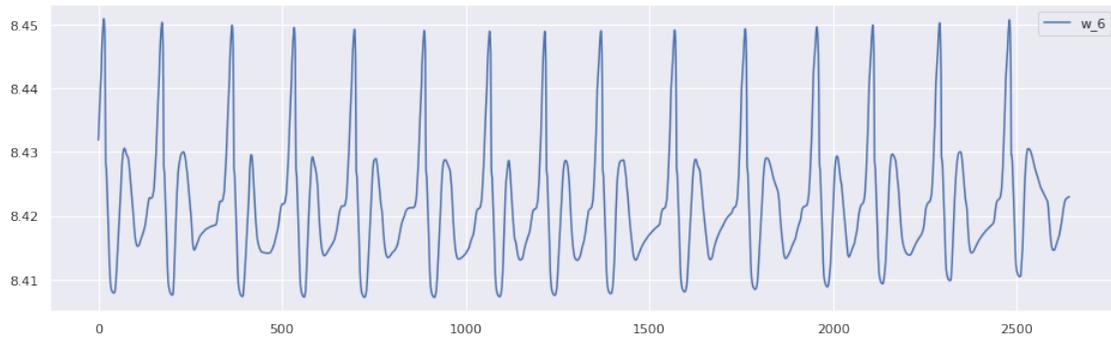


Figura 4.8: Variable ω_6 de reducción. Usando POD.

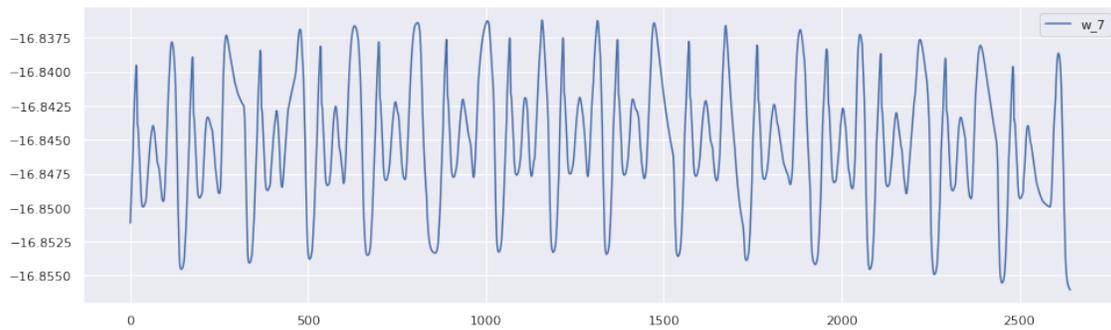


Figura 4.9: Variable ω_7 de reducción. Usando POD.

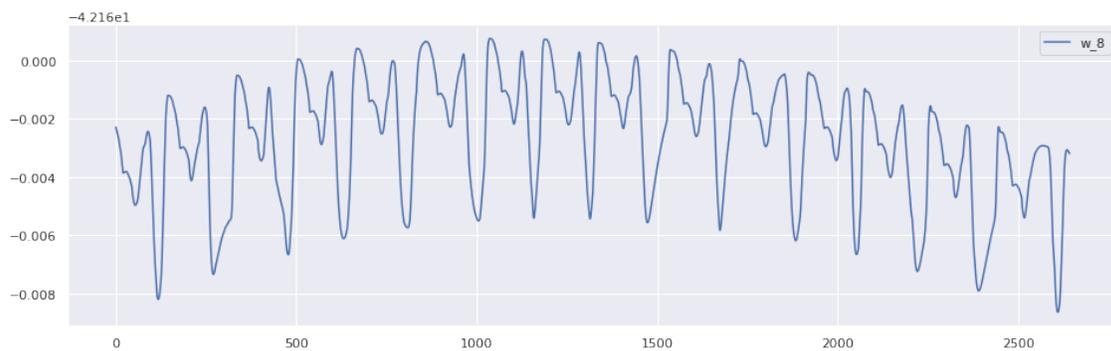


Figura 4.10: Variable ω_8 de reducción. Usando POD.

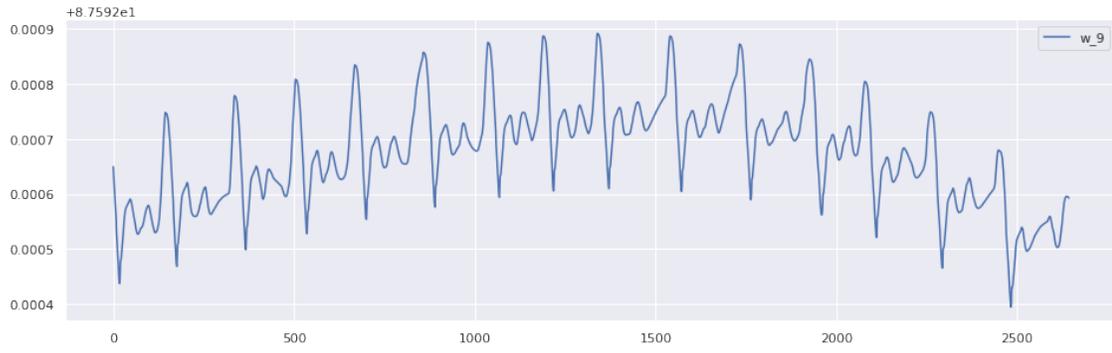


Figura 4.11: Variable ω_9 de reducción. Usando POD.

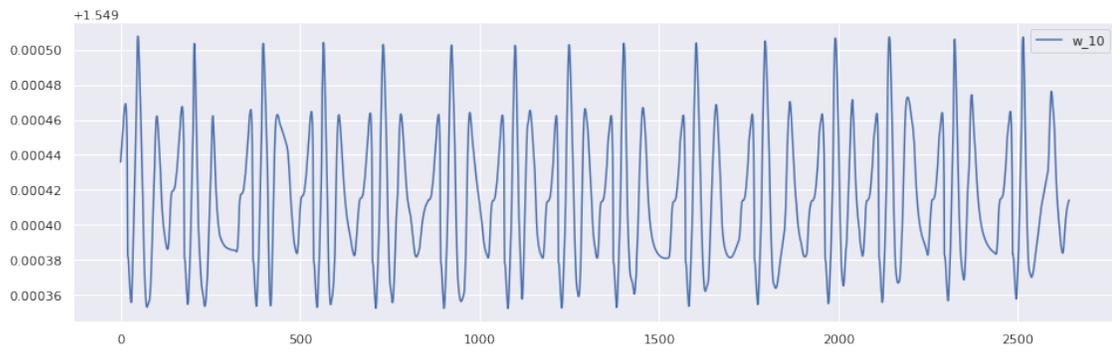


Figura 4.12: Variable ω_{10} de reducción. Usando POD.

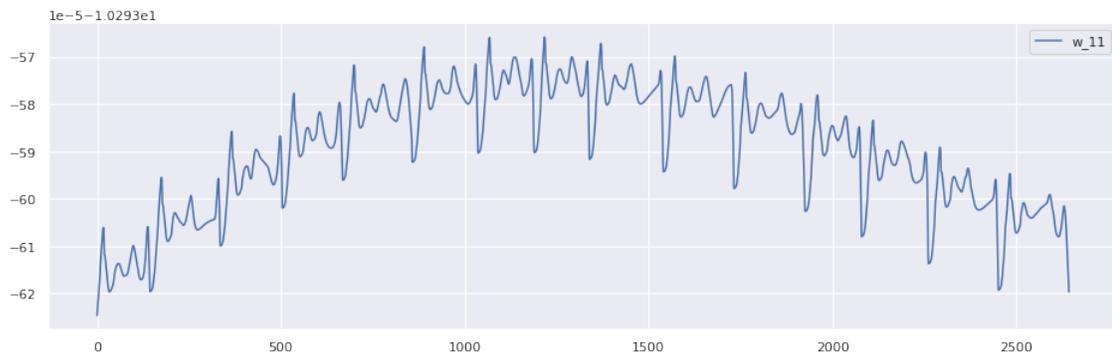


Figura 4.13: Variable ω_{11} de reducción. Usando POD.

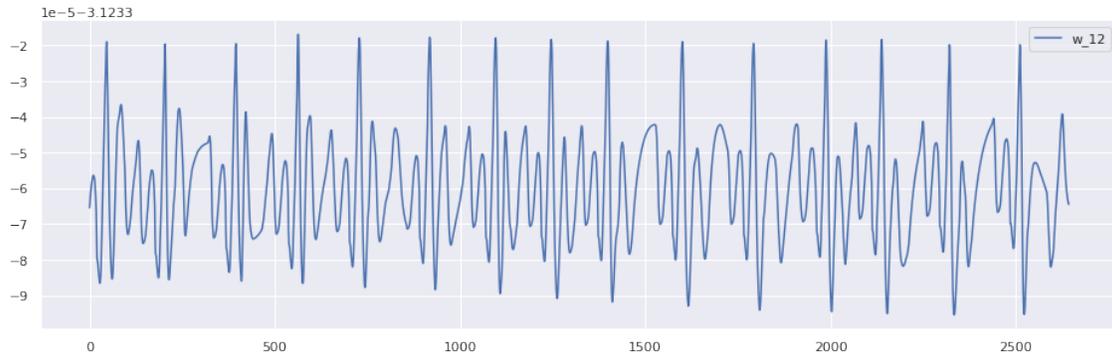


Figura 4.14: Variable ω_{12} de reducción. Usando POD.

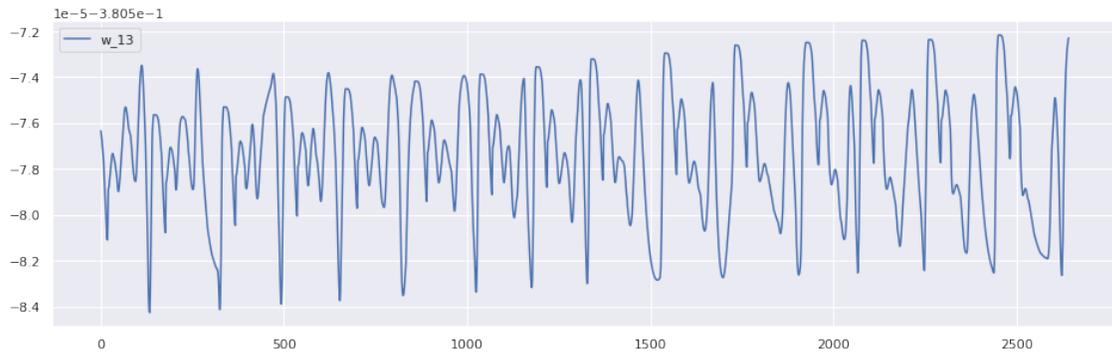


Figura 4.15: Variable ω_{13} de reducción. Usando POD.

Reconstrucción vía POD

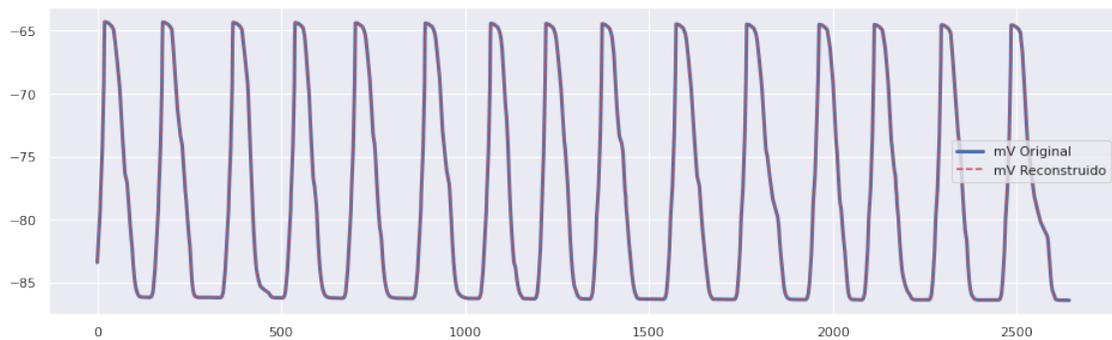


Figura 4.16: Reconstrucción de V . Usando POD.

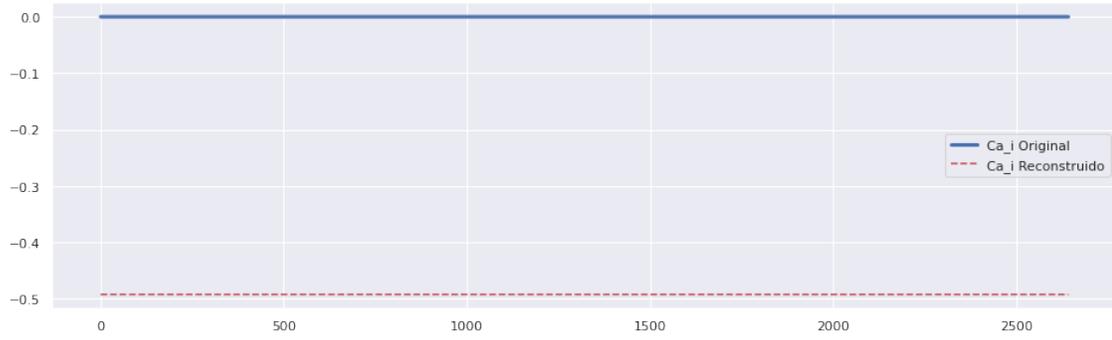


Figura 4.17: Reconstrucción de Ca_i . Usando POD.

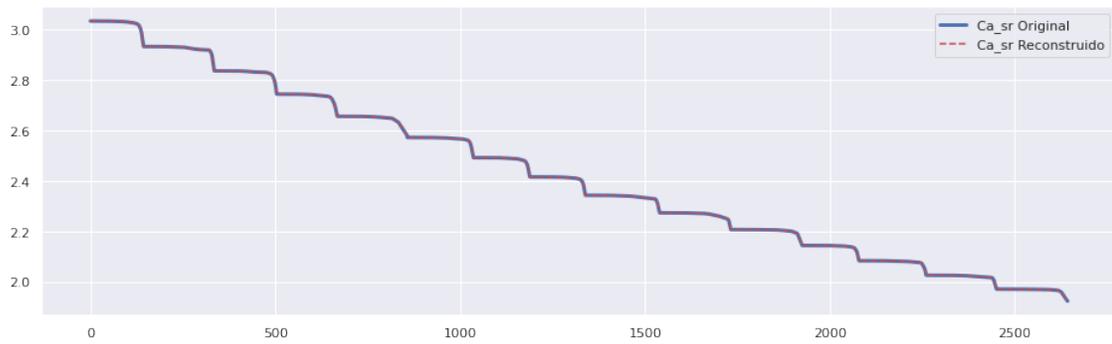


Figura 4.18: Reconstrucción de Ca_{sr} . Usando POD.

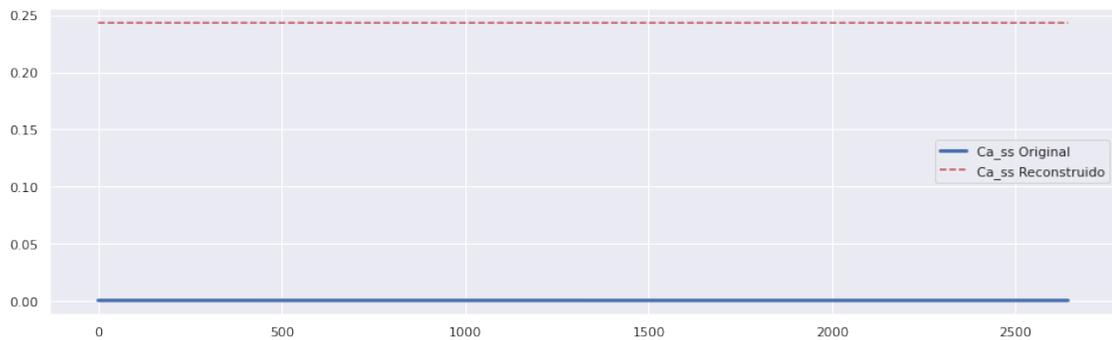


Figura 4.19: Reconstrucción de Ca_{ss} . Usando POD.

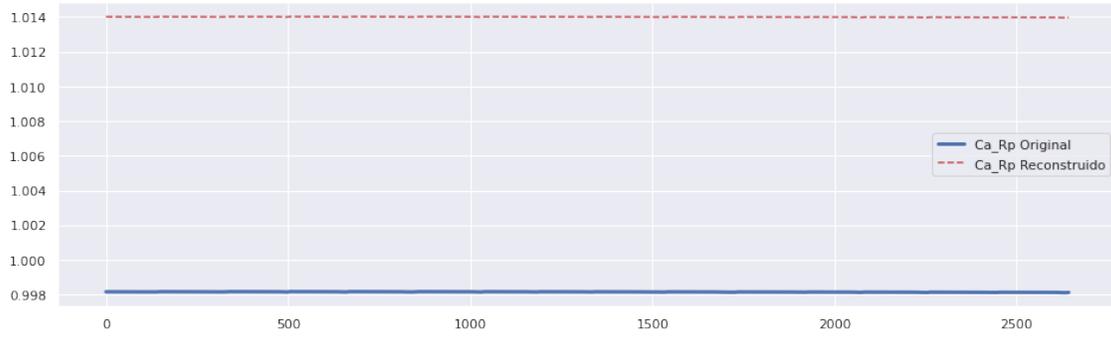


Figura 4.20: Reconstrucción de R_{prime} . Usando POD.

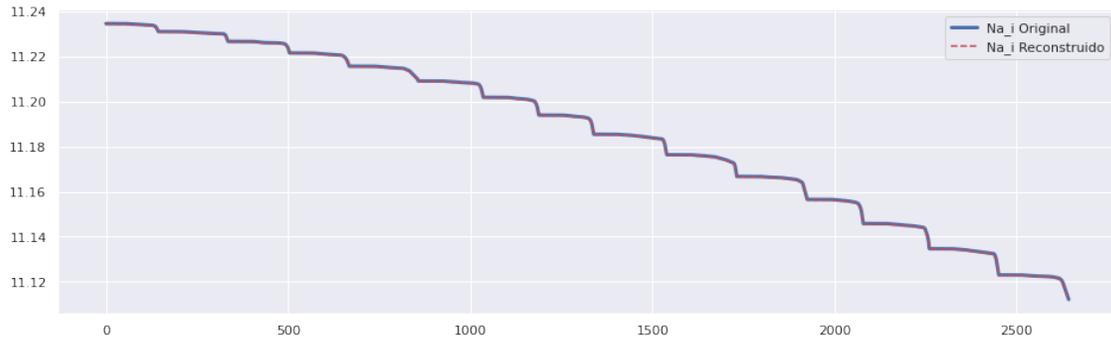


Figura 4.21: Reconstrucción de Na_i . Usando POD.

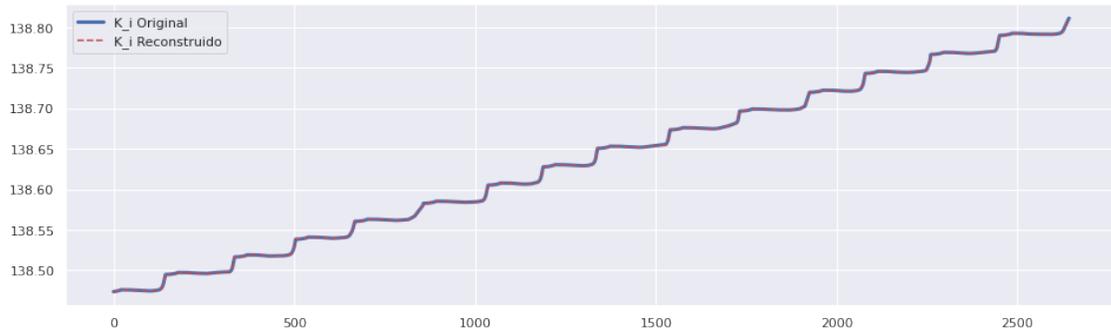


Figura 4.22: Reconstrucción de K_i . Usando POD.

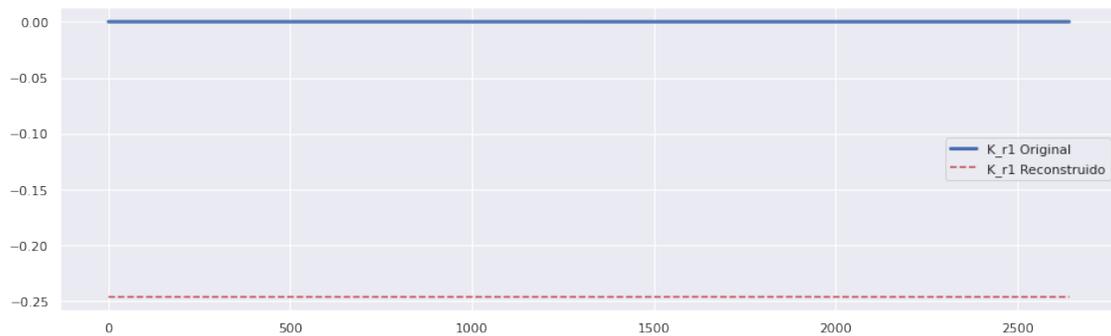


Figura 4.23: Reconstrucción de x_{r1} . Usando POD.

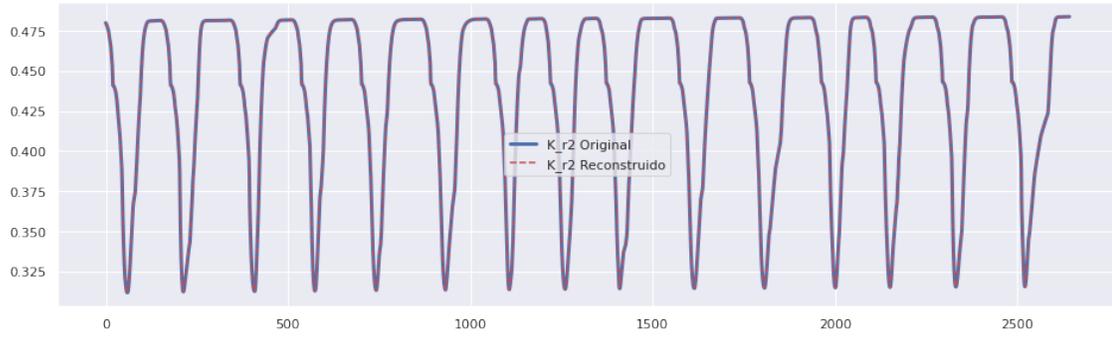


Figura 4.24: Reconstrucción de x_{r2} . Usando POD.

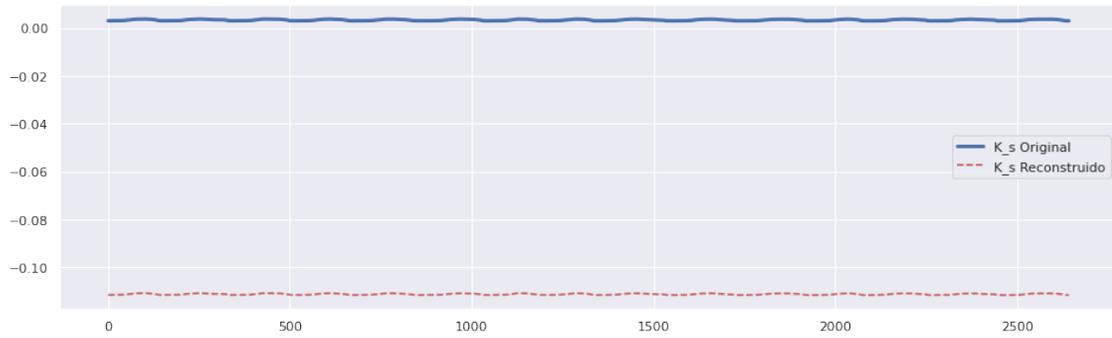


Figura 4.25: Reconstrucción de x_s . Usando POD.

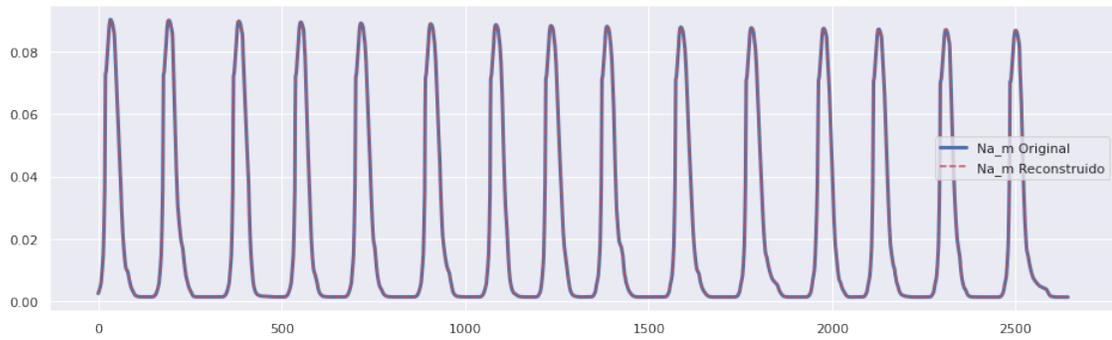


Figura 4.26: Reconstrucción de m . Usando POD.

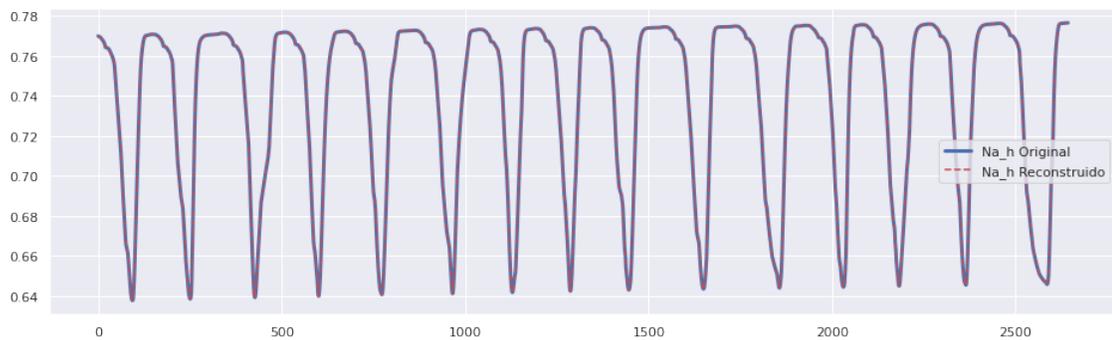


Figura 4.27: Reconstrucción de h . Usando POD.

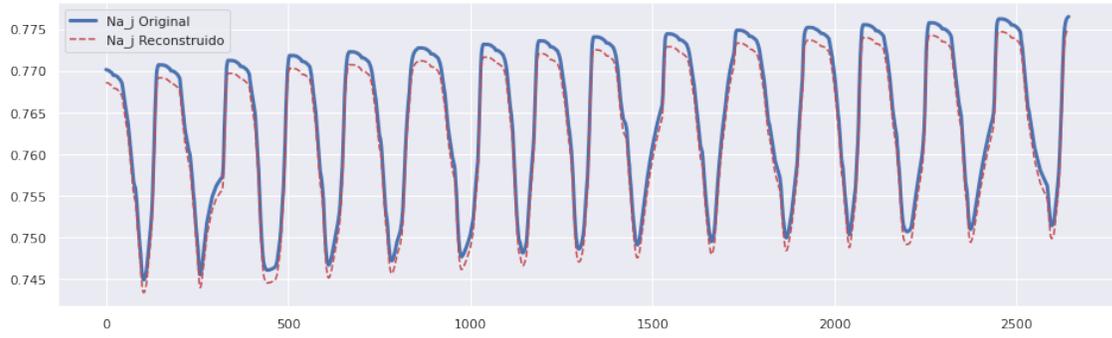


Figura 4.28: Reconstrucción de j . Usando POD.

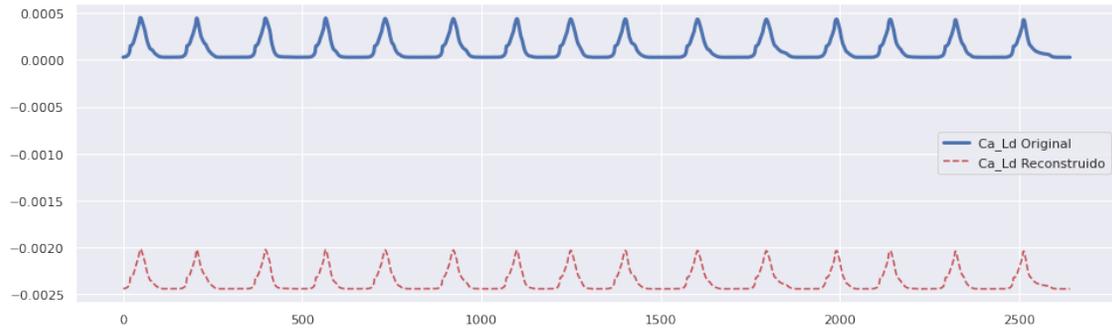


Figura 4.29: Reconstrucción de d . Usando POD.

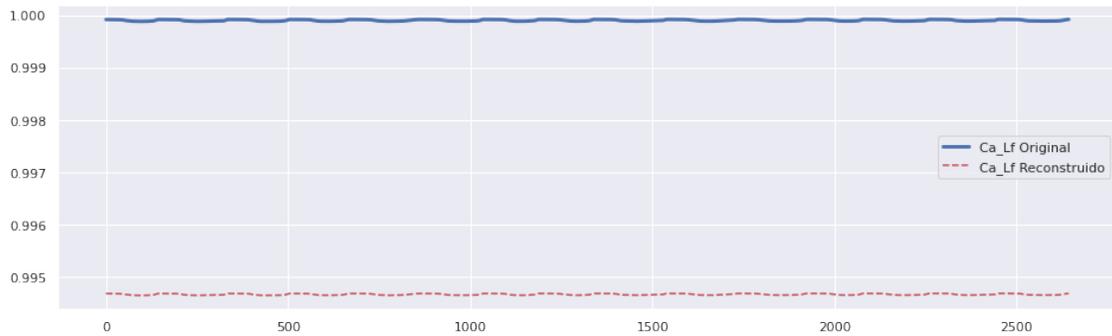


Figura 4.30: Reconstrucción de f . Usando POD.

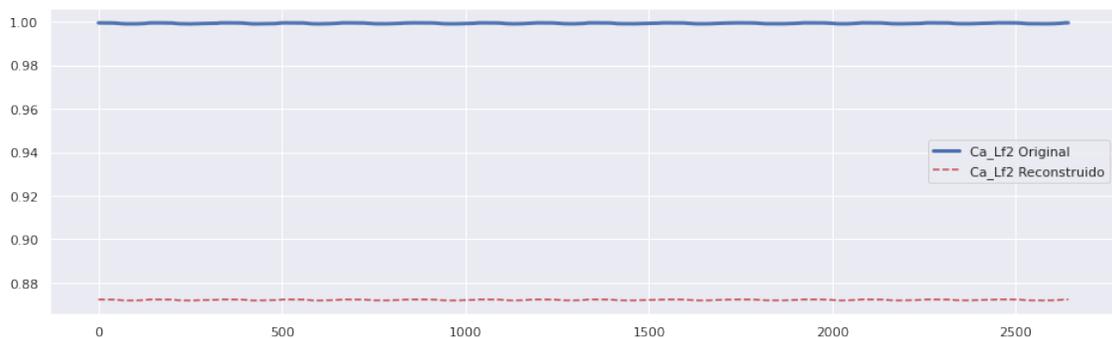


Figura 4.31: Reconstrucción de f_2 . Usando POD.

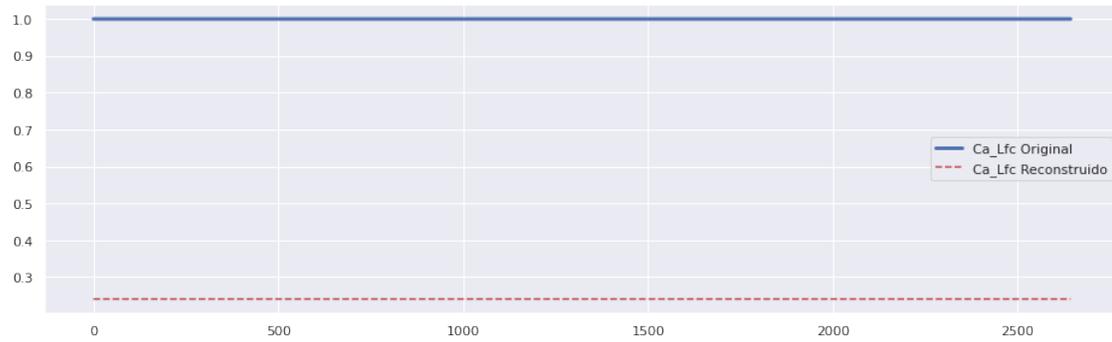


Figura 4.32: Reconstrucción de $fCass$. Usando POD.

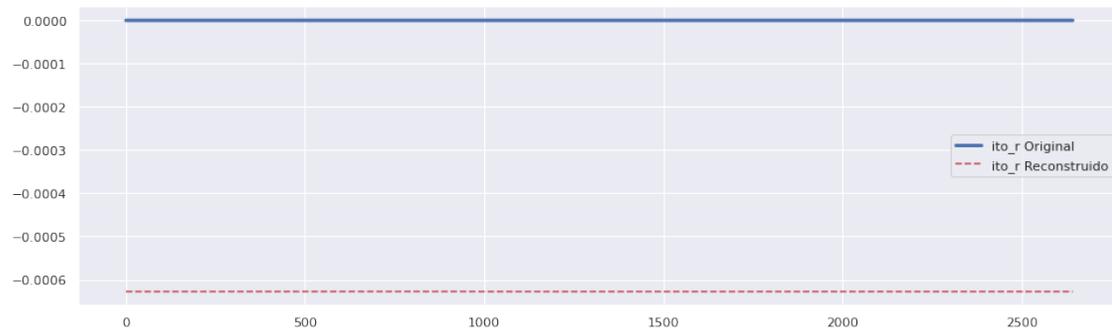


Figura 4.33: Reconstrucción de r . Usando POD.

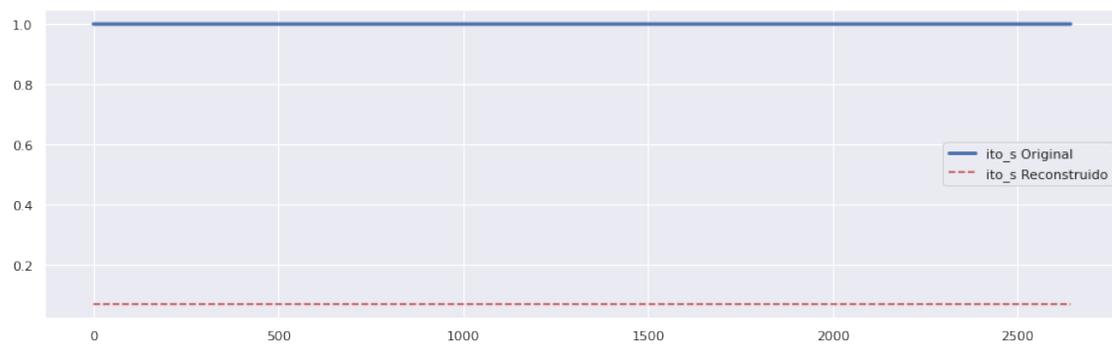


Figura 4.34: Reconstrucción de s . Usando POD.

Variables del sistema reducido, caso ten Tusscher-Panfilov.

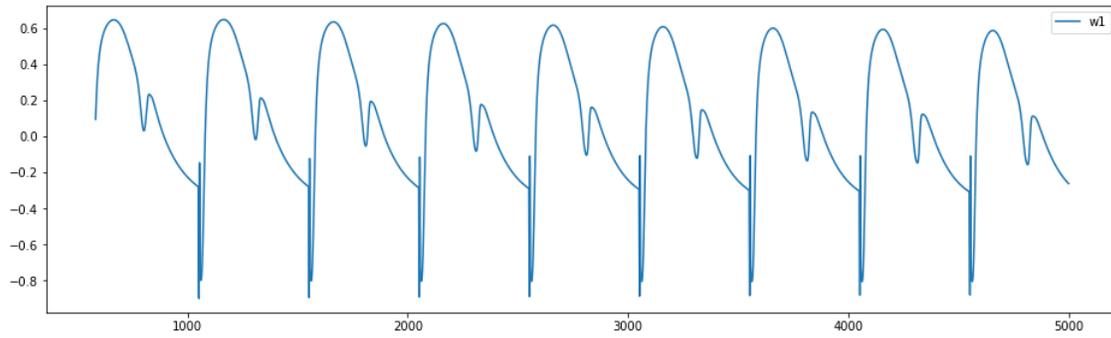


Figura 4.35: Variable ω_1 .

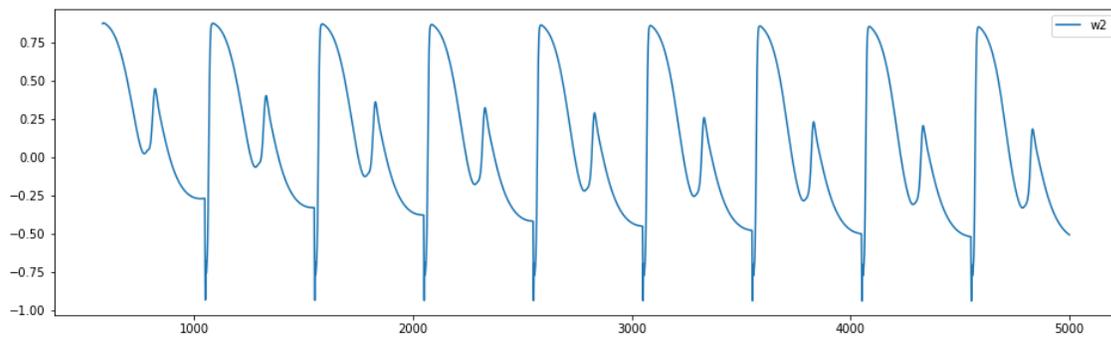


Figura 4.36: Variable ω_2 .

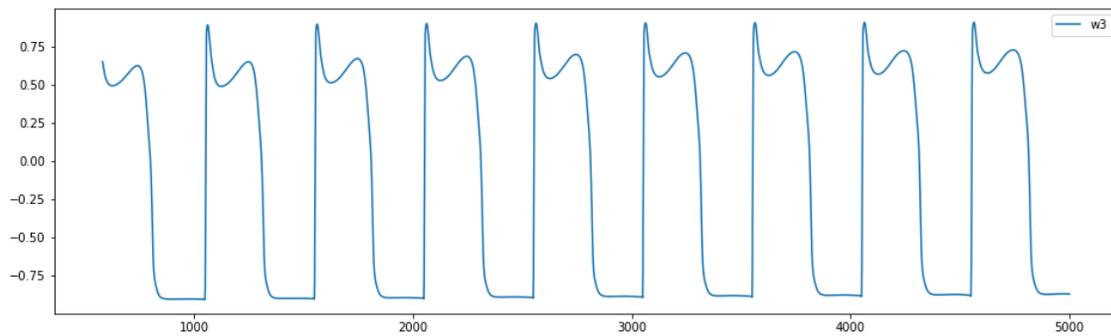


Figura 4.37: Variable ω_3 .

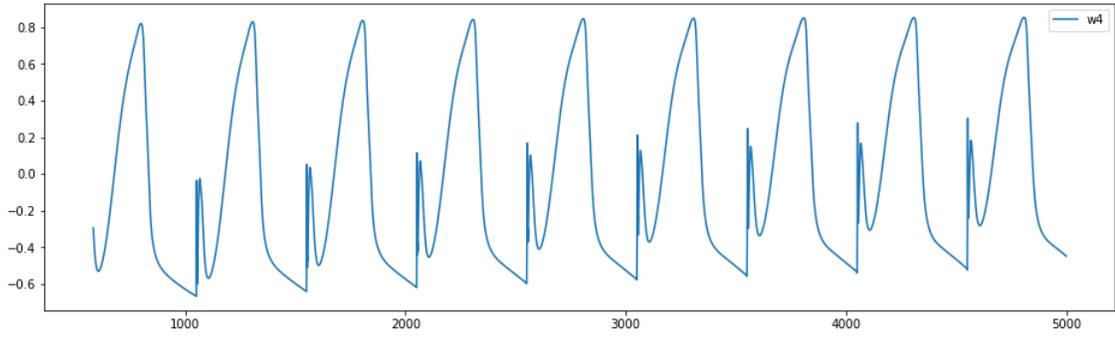


Figura 4.38: Variable ω_4 .

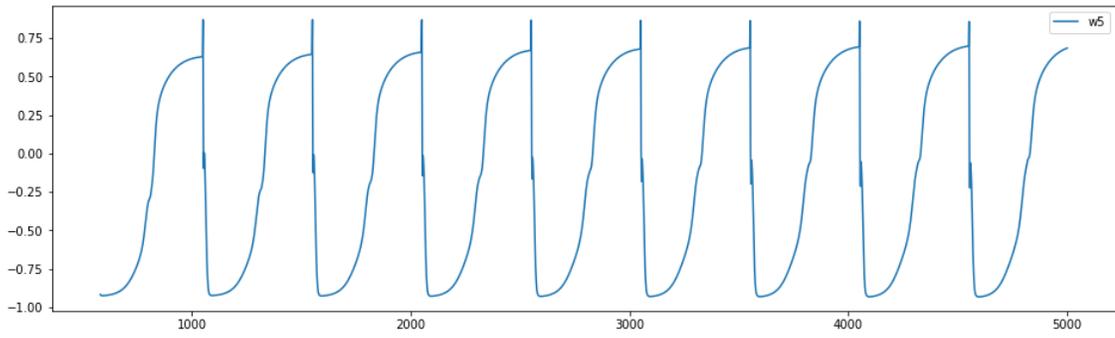


Figura 4.39: Variable ω_5 .

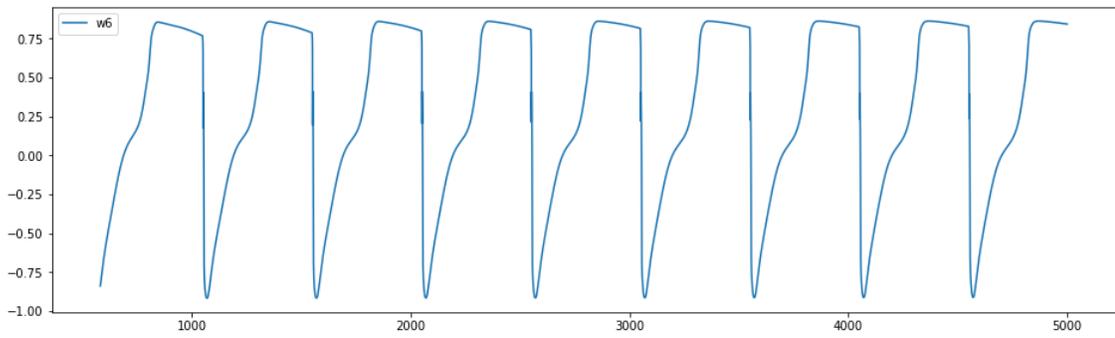


Figura 4.40: Variable ω_6 .

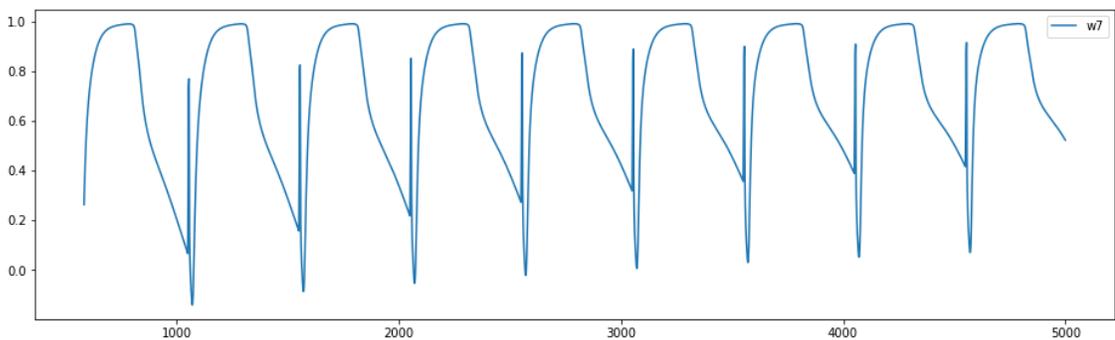


Figura 4.41: Variable ω_7 .

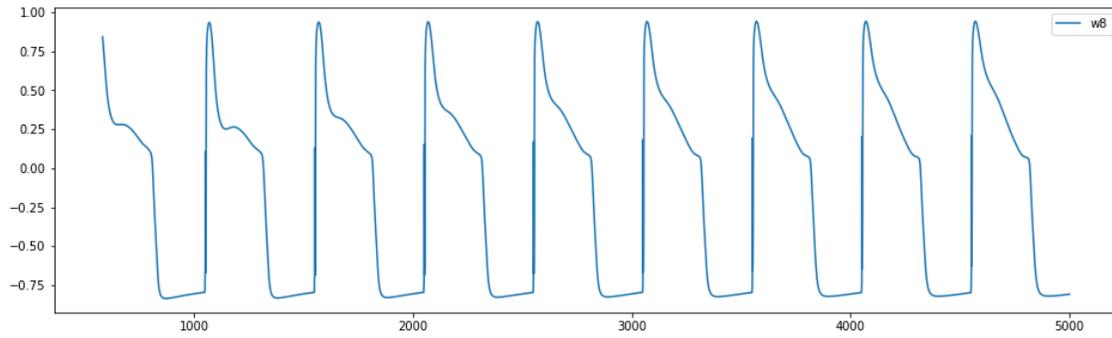


Figura 4.42: Variable ω_8 .

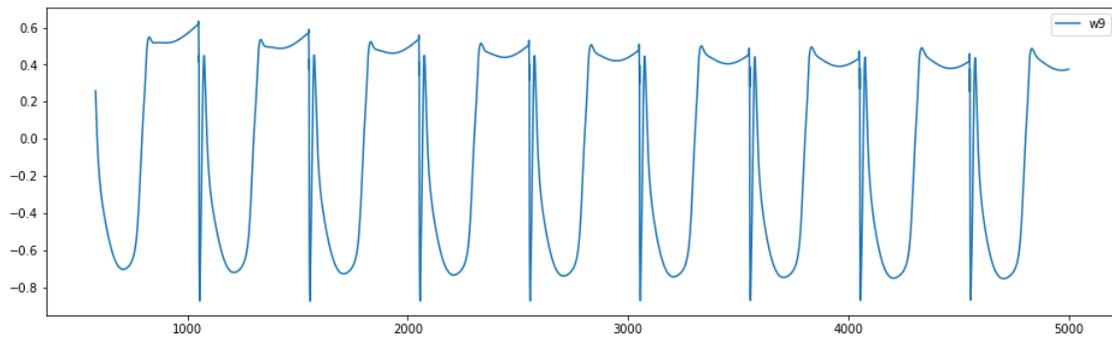


Figura 4.43: Variable ω_9 .

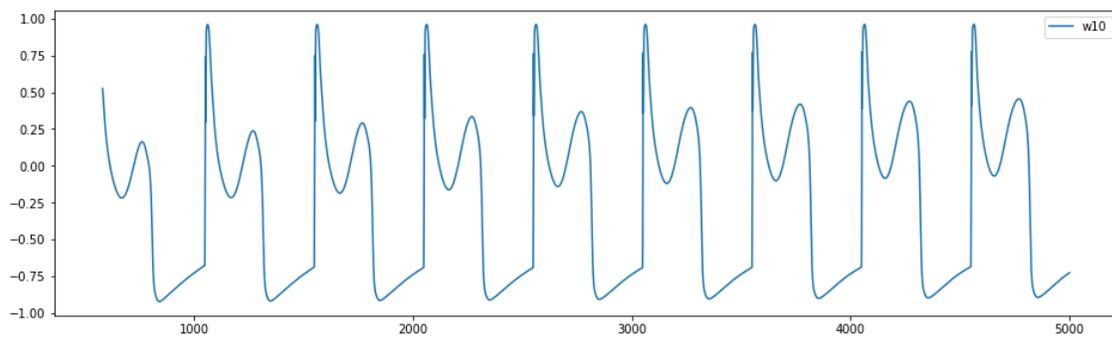


Figura 4.44: Variable ω_{10} .

Reconstrucción de variables de ten Tusscher-Panfilov usando Auto-encoders.

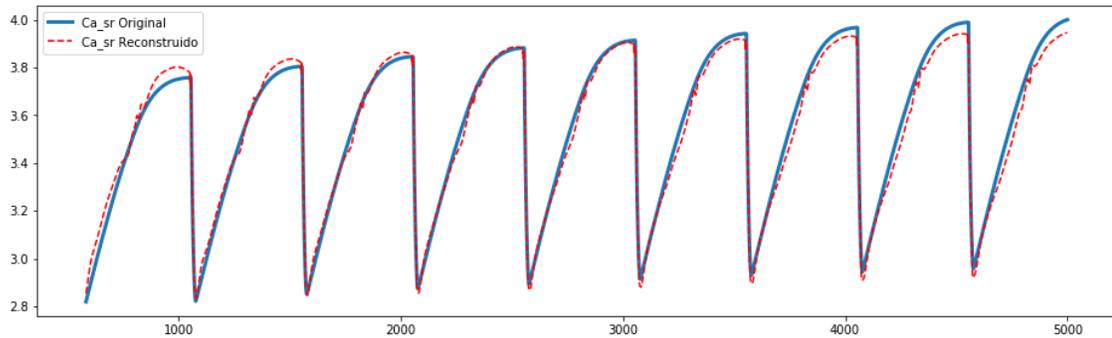


Figura 4.45: Reconstrucción de Ca_{sr} .

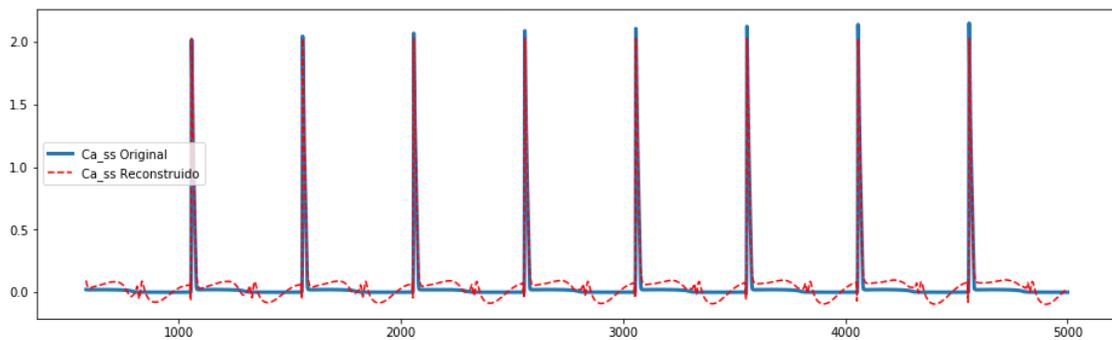


Figura 4.46: Reconstrucción de Ca_{ss} .

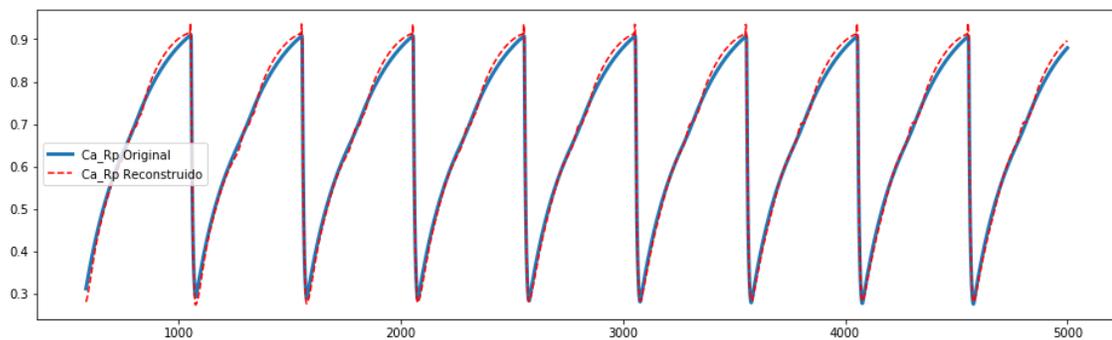


Figura 4.47: Reconstrucción de R_{prime} .

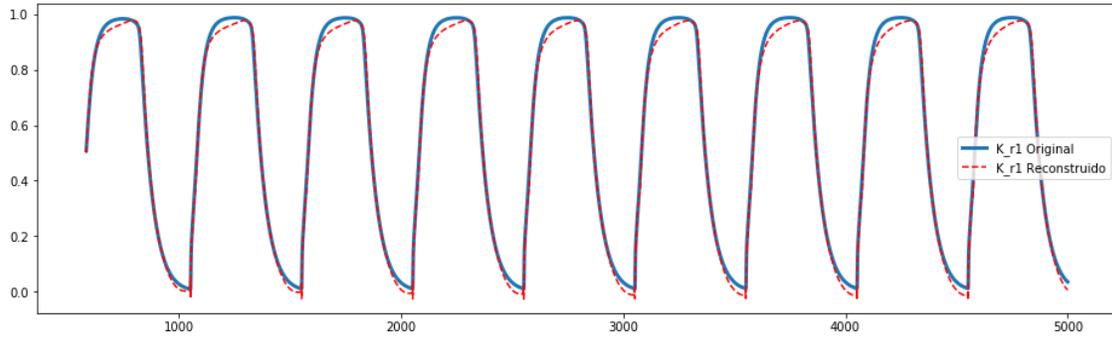


Figura 4.48: Reconstrucción de x_{r1} .

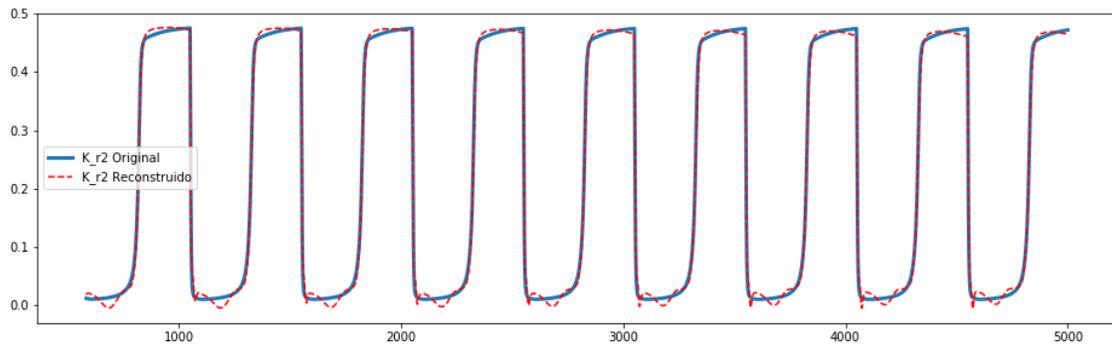


Figura 4.49: Reconstrucción de x_{r2} .

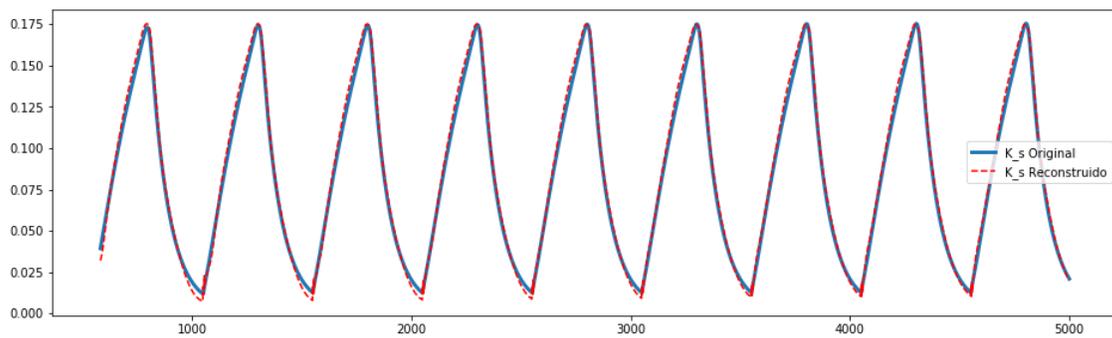


Figura 4.50: Reconstrucción de x_s .

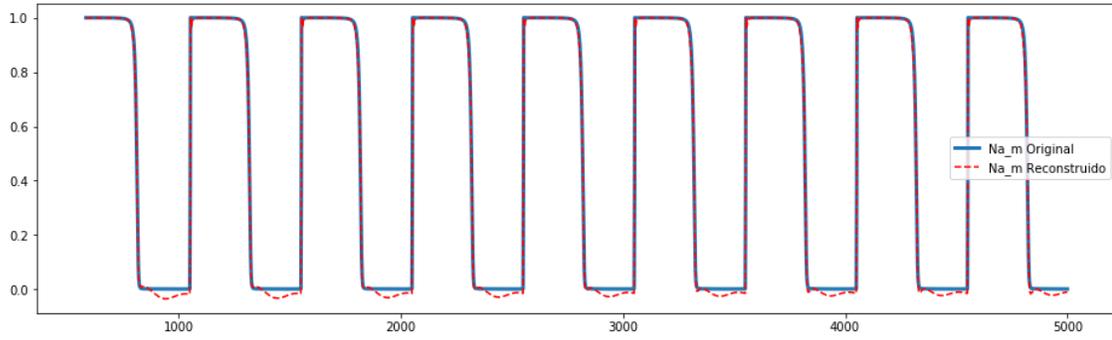


Figura 4.51: Reconstrucción de m .

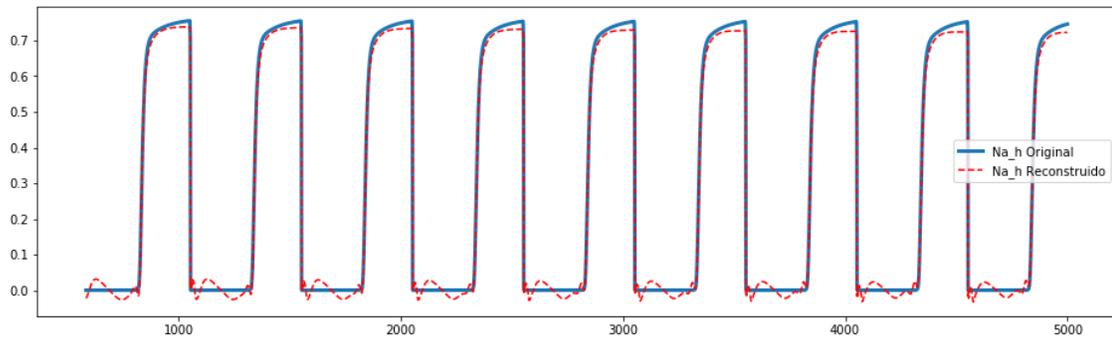


Figura 4.52: Reconstrucción de h .

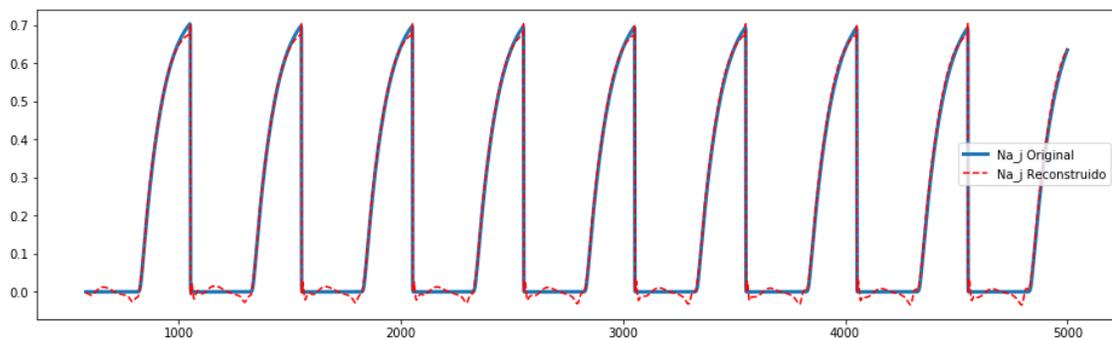


Figura 4.53: Reconstrucción de j .

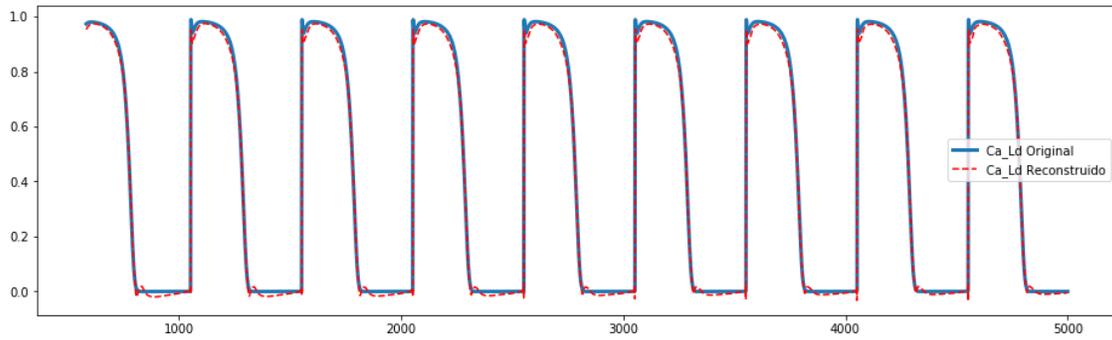


Figura 4.54: Reconstrucción de d .

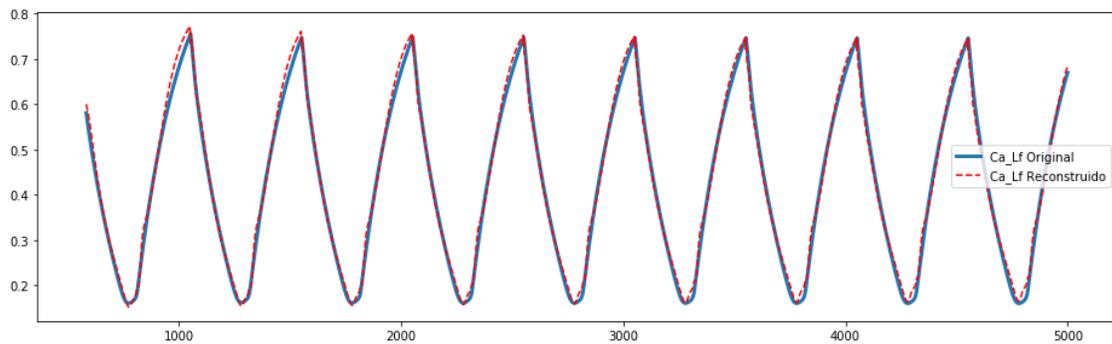


Figura 4.55: Reconstrucción de f .

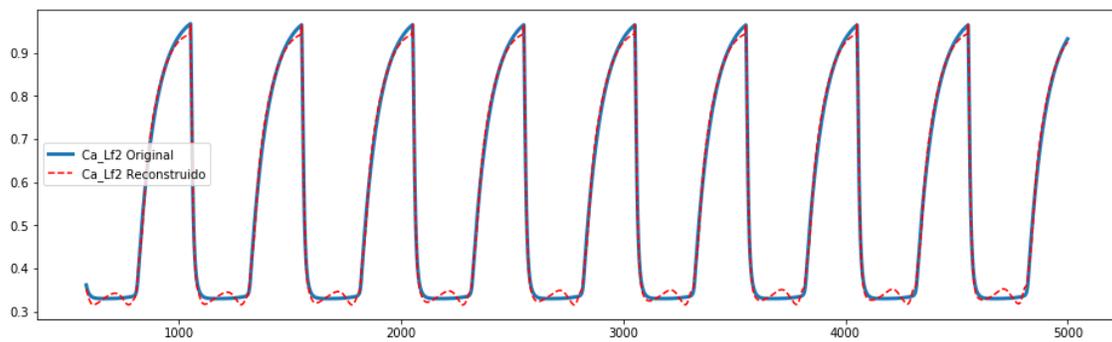


Figura 4.56: Reconstrucción de f_2 .

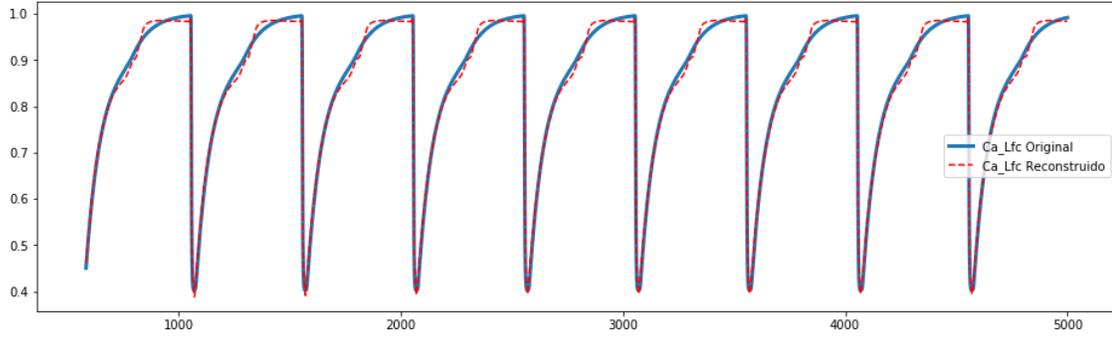


Figura 4.57: Reconstrucción de $fCass$.

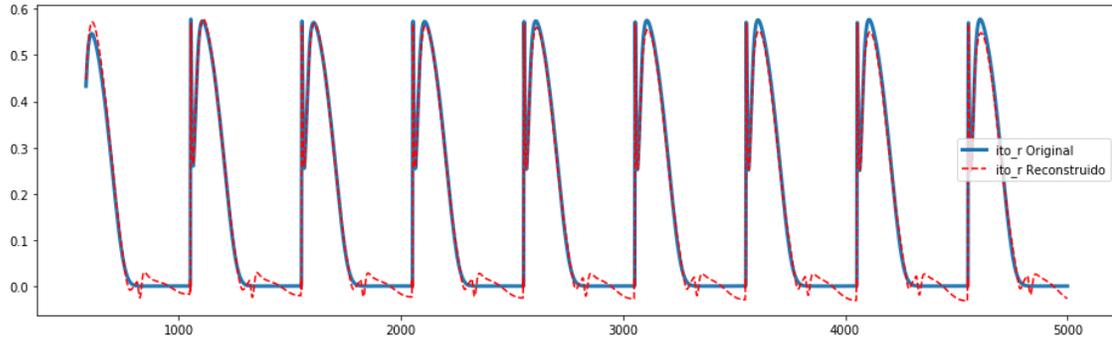


Figura 4.58: Reconstrucción de r .

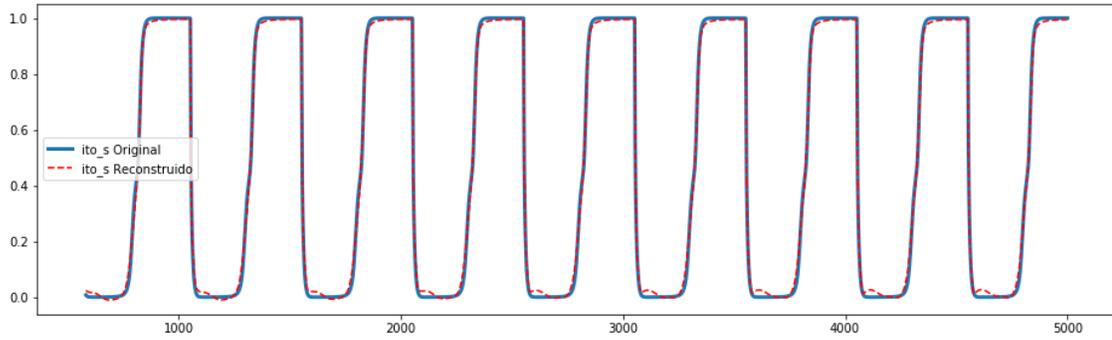


Figura 4.59: Reconstrucción de s .

Cálculo de E_V en el caso ten Tusscher-Panfilov

Dado:

$$\varepsilon = [\varepsilon_V, \varepsilon_{Nai}, \varepsilon_{Ki}, \varepsilon_{Cai}, \varepsilon_d, \varepsilon_f, \varepsilon_{f2}, \varepsilon_{fCass}, \varepsilon_h, \varepsilon_j, \varepsilon_m, \varepsilon_{xr1}, \varepsilon_{xr2}, \varepsilon_{xs}, \varepsilon_r, \varepsilon_s, \varepsilon_{Cass}, \varepsilon_{Casr}, \varepsilon_{Rprime}]$$

La primeras componente de E_f , definido como:

$$E_f = [E_V, E_{Nai}, E_{Ki}, E_{Cai}, E_d, E_f, E_{f2}, E_{fCass}, E_h, E_j, E_m, E_{xr1}, E_{xr2}, E_{xs}, E_r, E_s, E_{Cass}, E_{Casr}, E_{Rprime}]$$

Cumple:

$$E_V = -\varepsilon \cdot V1/C_m + O(\|\varepsilon\|_2^2)$$

Con

$$V1 = [V1_V, V1_{Nai}, V1_{Ki}, V1_{Cai}, V1_d, V1_f, V1_{f2}, V1_{fCass}, V1_h, V1_j, V1_m, V1_{xr1}, V1_{xr2}, V1_{xs}, V1_r, V1_s, V1_{Cass}, V1_{Casr}, V1_{Rprime}]$$

Donde

$$\begin{aligned} V1_V = & 4g_{CaL}FFTR \, d \, f \, f_2 \, fCass \left(\frac{0,25Ca_{ss}e^{2(V-15)FTR} - Ca_0}{e^{2(V-15)FTR} - 1} \right. \\ & \left. + (V-15)FTR e^{2(V-15)FTR} \frac{2Ca_0 - 0,5Ca_{ss}}{(e^{2(V-15)FTR} - 1)^2} \right) \\ & + \frac{K_{NaCa}e^{gVFTR}}{(Km_{Na}^3 + Na_0^3)(Km_{Ca} + Ca_0)(1 + K_{sat}e^{(g-1)VFTR})} aNa_0^3Ca_iFTR e^{-VFTR} \\ & + \frac{K_{NaCa}FTR e^{gVFTR}(g(1 + K_{sat}e^{(g-1)VFTR}) - K_{sat}(g-1)e^{(g-1)VFTR})}{(Km_{Na}^3 + Na_0^3)(Km_{Ca} + Ca_0)(1 + K_{sat}e^{(g-1)VFTR})^2} (Na_i^3Ca_0 - ae^{-VFTR}Na_0^3Ca_i) \\ & + P_{NaK}FTR \frac{K_0Na_i}{(K_0 + Km_K)(Na_i + Km_{Na})} \frac{0,01245e^{-0,1VFTR} + 0,0353e^{-VFTR}}{(1 + 0,1245e^{-0,1VFTR} + 0,0353e^{-VFTR})^2} \\ & + 0,1gK1 \sqrt{\frac{K_0}{5,4}} \left(\frac{1}{0,1 + (1 + e^{0,06(V+RTF \log \frac{K_i}{K_0} - 200)}) \left(\frac{3e^{0,0002(V+RTF \log \frac{K_i}{K_0} - 100)} + e^{0,1(V+RTF \log \frac{K_i}{K_0} - 10)} \right)}{1 + e^{-0,5(V+RTF \log \frac{K_i}{K_0})}} \right) \\ & \frac{(V + RTF \log \frac{K_i}{K_0})0,06e^{0,06(V+RTF \log \frac{K_i}{K_0} - 200)} 3e^{0,0002(V+RTF \log \frac{K_i}{K_0} - 100)} + e^{0,1(V+RTF \log \frac{K_i}{K_0} - 10)}}{1 + e^{-0,5(V+RTF \log \frac{K_i}{K_0})}} \\ & \frac{\left(0,1 + (1 + e^{0,06(V+RTF \log \frac{K_i}{K_0} - 200)}) \left(\frac{3e^{0,0002(V+RTF \log \frac{K_i}{K_0} - 100)} + e^{0,1(V+RTF \log \frac{K_i}{K_0} - 10)} \right)}{1 + e^{-0,5(V+RTF \log \frac{K_i}{K_0})}} \right)^2}{(V + RTF \log \frac{K_i}{K_0}) \frac{1 + e^{0,06(V+RTF \log \frac{K_i}{K_0} - 200)}}{1 + e^{-0,5(V+RTF \log \frac{K_i}{K_0})}} \left(0,0006e^{0,0002(V+RTF \log \frac{K_i}{K_0} - 100)} + 0,1e^{0,1(V+RTF \log \frac{K_i}{K_0} - 10)} \right)} \\ & \frac{\left(0,1 + (1 + e^{0,06(V+RTF \log \frac{K_i}{K_0} - 200)}) \left(\frac{3e^{0,0002(V+RTF \log \frac{K_i}{K_0} - 100)} + e^{0,1(V+RTF \log \frac{K_i}{K_0} - 10)} \right)}{1 + e^{-0,5(V+RTF \log \frac{K_i}{K_0})}} \right)^2}{(V + RTF \log \frac{K_i}{K_0}) \frac{1 + e^{0,06(V+RTF \log \frac{K_i}{K_0} - 200)}}{(1 + e^{-0,5(V+RTF \log \frac{K_i}{K_0})})^2} 1,5e^{0,0002(V+RTF \log \frac{K_i}{K_0} - 100)} - 0,5(V+RTF \log \frac{K_i}{K_0})} \\ & \frac{\left(0,1 + (1 + e^{0,06(V+RTF \log \frac{K_i}{K_0} - 200)}) \left(\frac{3e^{0,0002(V+RTF \log \frac{K_i}{K_0} - 100)} + e^{0,1(V+RTF \log \frac{K_i}{K_0} - 10)} \right)}{1 + e^{-0,5(V+RTF \log \frac{K_i}{K_0})}} \right)^2}{\left(0,1 + (1 + e^{0,06(V+RTF \log \frac{K_i}{K_0} - 200)}) \left(\frac{3e^{0,0002(V+RTF \log \frac{K_i}{K_0} - 100)} + e^{0,1(V+RTF \log \frac{K_i}{K_0} - 10)} \right)}{1 + e^{-0,5(V+RTF \log \frac{K_i}{K_0})}} \right)^2} \\ & \frac{(V + RTF \log \frac{K_i}{K_0}) \frac{1 + e^{0,06(V+RTF \log \frac{K_i}{K_0} - 200)}}{(1 + e^{-0,5(V+RTF \log \frac{K_i}{K_0})})^2} 0,5e^{0,1(V+RTF \log \frac{K_i}{K_0} - 10)} - 0,5(V+RTF \log \frac{K_i}{K_0})}{\left(0,1 + (1 + e^{0,06(V+RTF \log \frac{K_i}{K_0} - 200)}) \left(\frac{3e^{0,0002(V+RTF \log \frac{K_i}{K_0} - 100)} + e^{0,1(V+RTF \log \frac{K_i}{K_0} - 10)} \right)}{1 + e^{-0,5(V+RTF \log \frac{K_i}{K_0})}} \right)^2} \\ & + g_{Nam}^3 h_j + g_{NaB} + g_{bCa} + g_{To} r_s \\ & + g_{Kr} \sqrt{\frac{K_0}{5,4}} x_{r1} x_{r2} + g_{Ks} X_s^2 + g_{pK} \frac{1 + e^{(25-V)/5,98} + 1/5,98e^{(25-V)/5,98} (V + RTF \log \frac{K_i}{K_0})}{(1 + e^{(25-V)/5,98})^2} \end{aligned}$$

$$\begin{aligned}
V1_{Nai} &= \frac{K_{NaCa}e^{gVFTR}}{(Km_{Na}^3 + Na_0^3)(Km_{Ca} + Ca_0)(1 + K_{sat}e^{(g-1)VFTR})} 3Ca_0Na_i^2 \\
&+ P_{NaK} \frac{K_0Km_{Na}}{(K_0 + Km_K)(Na_i + Km_{Na})^2} \frac{1}{1 + 0,1245e^{-0,1VFTR} + 0,0353e^{-VFTR}} \\
&+ g_{Na}m^3hj \frac{RTF}{Na_i} + g_{Nab} \frac{RTF}{Na_i} + g_{K_s}x_s^2 \frac{P_{NaK}RTF}{K_i + P_{NaK}Na_i} \\
V1_{Ki} &= 0,1gK_1 \sqrt{\frac{K_0}{5,4}} \frac{RTF}{K_i} \left(\frac{1}{0,1 + (1 + e^{0,06(V+RTF \log \frac{K_i}{K_0} - 200)}) \left(\frac{3e^{0,0002(V+RTF \log \frac{K_i}{K_0} - 100)} + e^{0,1(V+RTF \log \frac{K_i}{K_0} - 10)}\right)}{1 + e^{-0,5(V+RTF \log \frac{K_i}{K_0})}} \right)} \\
&\frac{(V + RTF \log \frac{K_i}{K_0})0,06e^{0,06(V+RTF \log \frac{K_i}{K_0} - 200)} \frac{3e^{0,0002(V+RTF \log \frac{K_i}{K_0} - 100)} + e^{0,1(V+RTF \log \frac{K_i}{K_0} - 10)}}{1 + e^{-0,5(V+RTF \log \frac{K_i}{K_0})}}}{\left(0,1 + (1 + e^{0,06(V+RTF \log \frac{K_i}{K_0} - 200)}) \left(\frac{3e^{0,0002(V+RTF \log \frac{K_i}{K_0} - 100)} + e^{0,1(V+RTF \log \frac{K_i}{K_0} - 10)}\right)}{1 + e^{-0,5(V+RTF \log \frac{K_i}{K_0})}} \right)^2} \\
&\frac{(V + RTF \log \frac{K_i}{K_0}) \frac{1 + e^{0,06(V+RTF \log \frac{K_i}{K_0} - 200)}}{1 + e^{-0,5(V+RTF \log \frac{K_i}{K_0})}} \left(0,0006e^{0,0002(V+RTF \log \frac{K_i}{K_0} - 100)} + 0,1e^{0,1(V+RTF \log \frac{K_i}{K_0} - 10)} \right)}{\left(0,1 + (1 + e^{0,06(V+RTF \log \frac{K_i}{K_0} - 200)}) \left(\frac{3e^{0,0002(V+RTF \log \frac{K_i}{K_0} - 100)} + e^{0,1(V+RTF \log \frac{K_i}{K_0} - 10)}\right)}{1 + e^{-0,5(V+RTF \log \frac{K_i}{K_0})}} \right)^2} \\
&\frac{(V + RTF \log \frac{K_i}{K_0}) \frac{1 + e^{0,06(V+RTF \log \frac{K_i}{K_0} - 200)}}{(1 + e^{-0,5(V+RTF \log \frac{K_i}{K_0})})^2} 1,5e^{0,0002(V+RTF \log \frac{K_i}{K_0} - 100) - 0,5(V+RTF \log \frac{K_i}{K_0})}}{\left(0,1 + (1 + e^{0,06(V+RTF \log \frac{K_i}{K_0} - 200)}) \left(\frac{3e^{0,0002(V+RTF \log \frac{K_i}{K_0} - 100)} + e^{0,1(V+RTF \log \frac{K_i}{K_0} - 10)}\right)}{1 + e^{-0,5(V+RTF \log \frac{K_i}{K_0})}} \right)^2} \\
&\frac{(V + RTF \log \frac{K_i}{K_0}) \frac{1 + e^{0,06(V+RTF \log \frac{K_i}{K_0} - 200)}}{(1 + e^{-0,5(V+RTF \log \frac{K_i}{K_0})})^2} 0,5e^{0,1(V+RTF \log \frac{K_i}{K_0} - 10) - 0,5(V+RTF \log \frac{K_i}{K_0})}}{\left(0,1 + (1 + e^{0,06(V+RTF \log \frac{K_i}{K_0} - 200)}) \left(\frac{3e^{0,0002(V+RTF \log \frac{K_i}{K_0} - 100)} + e^{0,1(V+RTF \log \frac{K_i}{K_0} - 10)}\right)}{1 + e^{-0,5(V+RTF \log \frac{K_i}{K_0})}} \right)^2} \\
&+ g_{Tors} \frac{RTF}{K_i} + g_{Kr}x_{r1}x_{r2} \frac{RTF}{K_i} + g_{K_s}x_s^2 \frac{RTF}{K_i + P_{NaK}Na_i} + g_{pK} \frac{RTF}{K_i(1 + e^{(25-V)/5,98})} \\
V1_{Cai} &= \frac{0,5g_{bCa}RTF}{Ca_i} + \frac{g_{pCa}}{(Ca_i + K_{pCa})^2} \\
&- \frac{K_{NaCa}e^{gVFTR}}{(Km_{Na}^3 + Na_0^3)(Km_{Ca} + Ca_0)(1 + K_{sat}e^{(g-1)VFTR})} aNa_0^3e^{-VFTR} \\
V1_d &= 4g_{CaL}FFTR f f_2 fCass(V - 15) \left(\frac{0,25Ca_{ss}e^{2(V-15)FTR} - Ca_0}{e^{2(V-15)FTR} - 1} \right) \\
V1_f &= 4g_{CaL}FFTR d f_2 fCass(V - 15) \left(\frac{0,25Ca_{ss}e^{2(V-15)FTR} - Ca_0}{e^{2(V-15)FTR} - 1} \right) \\
V1_{f2} &= 4g_{CaL}FFTR d f fCass(V - 15) \left(\frac{0,25Ca_{ss}e^{2(V-15)FTR} - Ca_0}{e^{2(V-15)FTR} - 1} \right) \\
V1_{fCass} &= 4g_{CaL}FFTR d f f_2(V - 15) \left(\frac{0,25Ca_{ss}e^{2(V-15)FTR} - Ca_0}{e^{2(V-15)FTR} - 1} \right) \\
V1_h &= g_{Na}m^3j \left(V + RTF \log \frac{Na_i}{Na_0} \right) \\
V1_j &= g_{Na}m^3h \left(V + RTF \log \frac{Na_i}{Na_0} \right) \\
V1_m &= g_{Na}3m^2hj \left(V + RTF \log \frac{Na_i}{Na_0} \right)
\end{aligned}$$

$$\begin{aligned}
V1_{xr1} &= g_{Kr} \sqrt{\frac{K_0}{5,4}} x_{r2} \left(V + RTF \log \frac{K_i}{K_0} \right) \\
V1_{xr2} &= g_{Kr} \sqrt{\frac{K_0}{5,4}} x_{r1} \left(V + RTF \log \frac{K_i}{K_0} \right) \\
V1_{xs} &= 2g_{Ks} x_s \left(V + RTF \log \frac{K_i + P_{NaK} Na_i}{K_0 + P_{NaK} Na_0} \right) \\
V1_r &= g_{Tos} \left(V + RTF \log \frac{K_i}{K_0} \right) \\
V1_s &= g_{Tor} \left(V + RTF \log \frac{K_i}{K_0} \right) \\
V1_{Cass} &= g_{CaL} FFTR \text{ d f } f_2 \text{ fCass} (V - 15) \frac{e^{2(V-15)FTR}}{e^{2(V-15)FTR} - 1} \\
V1_{Casr} &= 0 \\
V1_{Rprime} &= 0
\end{aligned}$$

Software utilizado

Lista de software y librerías utilizados:

- MatLab R2016a: Usado para programar el método de ajuste de nulclinas. Comandos de *ode45* y *polyfit* fueron usados para resolver los sistemas y ajustar los polinomios a las nulclinas, respectivamente.
- Python 3: Generación del dataset para el entrenamiento y test de los autoencoders. Esto se realizó mediante la resolución de *HH* y *ten Tusscher-Panfilov* usando *Scipy*.
- MyoKit: Paquete de Python, disponible como software que facilita la obtención de soluciones numéricas para modelos asociados al análisis de miocitos. Más información en la página oficial www.myokit.org
- Tensorflow 1.15: Librería de código abierto especializada en Machine Learning. Programado en notebooks de Jupyter usando el lenguaje *Python*. Para el entrenamiento del caso *HH*, se requirió del uso de Jupyter. Mientras que el entrenamiento del caso de *ten Tusscher-Panfilov* fue realizado en *Google Colab*, como también fue aprovechado el cómputo que entrega el Laboratorio Nacional de Computación de Alto Rendimiento (*NLHPC*).