



UNIVERSIDAD AUTÓNOMA
DEL ESTADO DE HIDALGO



INSTITUTO DE CIENCIAS BÁSICAS E INGENIERÍA

AREA ACADÉMICA DE COMPUTACIÓN Y ELECTRÓNICA

MAESTRÍA EN AUTOMATIZACIÓN Y CONTROL

CONTROL DE UN SISTEMA SUBACTUADO EMPLEANDO REDES
NEURONALES DE PULSO

T E S I S

QUE PARA OBTENER EL GRADO DE MAESTRO EN
CIENCIAS EN AUTOMATIZACIÓN Y CONTROL

PRESENTA:

ING. OMAR SAMPERIO VÁZQUEZ

ASESORES:

DR. LUIS ENRIQUE RAMOS VELASCO

DR. JOEL SUÁREZ CANSINO

PACHUCA HGO. MÉXICO, MARZO 2016



Ing. Omar Samperio Vázquez

**Control de un sistema
subactuado empleando
redes neuronales de pulso**



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE HIDALGO

Instituto de Ciencias Básicas e Ingeniería

Institute of Basic Sciences and Engineering

Área Académica de Computación y Electrónica




Computer Science and Electronics Department

Mineral de la Reforma, Hgo., a 14 de marzo de 2016

Oficio No. MCAC20_2016

Ing. Omar Samperio Vázquez
PRESENTE

Por medio de la presente y en mi calidad de coordinador de la Maestría en Ciencias en Automatización y Control, del Área Académica de Computación y Electrónica (AACyE) de la Universidad Autónoma del Estado de Hidalgo (UAEH), me es grato informarle que el Jurado asignado para la revisión de su trabajo de tesis titulado: "Control de un sistema subactuado empleando redes neuronales de pulso", dirigido por el tesis el Dr. Luis Enrique Ramos Velasco y Dr. Joel Suarez Cansino, que para obtener el grado de Maestro en Ciencias en Automatización y Control fue presentado por usted, ha tenido a bien en reunión de sinodales, autorizarlo para impresión. A continuación se integran las firmas de conformidad de los integrantes del Jurado:

Dr. Omar Jacobo Santos Sánchez	(Presidente)	AACyE-UAEH	
M. en C. Marco Márquez Vera	(Secretario)	UP-pachuca	
Dr. Luis Enrique Ramos Velasco	(Vocal)	AACyE-UAEH	
Dr. Joel Suarez Cansino	(Suplente)	AACyE-UAEH	_____

UNIVERSIDAD AUTÓNOMA DEL ESTADO DE HIDALGO

Atentamente
"Amor, Orden y Progreso"



Dr. Jesús Patricio Ordaz Oliver
Coordinador de la Maestría en Ciencias en Automatización y Control
Universidad Autónoma del Estado de Hidalgo



Instituto de Ciencias Básicas e Ingeniería

Área Académica de Computación y Electrónica

c.c.p. Dr. Orlando Ávila Pozos, Director del Instituto de Ciencias Básicas e Ingeniería
c.c.p. Dr. Hugo Romero Trejo, Jefe del Área Académica de Computación y Electrónica
c.c.p. Expediente/ apl



Ciudad del Conocimiento
Carretera Pachuca - Tulancingo km. 4.5
Colonia Carboneras
Mineral de la Reforma, Hidalgo, México, C.P. 42184
Tel. +52 771 7172000 exts. 2250 y 2251
jesus_ordaz@uaeh.edu.mx

www.uaeh.edu.mx

*A mis padres, mis hermanos y mis hijos
quienes me han dado todo su apoyo,
su comprensión, sus consejos y su paciencia
para culminar mis proyectos de manera exitosa.*

Gracias a

CONACYT por la beca de Maestría
otorgada durante el periodo
Agosto 2013 - Junio 2015, con
número de registro 551623.

Agradecimientos

En esta sección es mi deseo agradecer de forma breve a todos mis familiares, catedráticos y amigos que de alguna manera contribuyeron de manera personal y/o profesional en el desarrollo de esta tesis.

Agradezco a la Universidad Autónoma del Estado de Hidalgo, institución que me ha proporcionado los medios y conocimientos necesarios para culminar de manera exitosa la *Maestría en Automatización y Control*.

Especialmente, doy gracias a dios por darme la oportunidad de contar con mis padres Genaro Samperio Pacheco y Margarita Vázquez Ortíz así como a mis hermanos Brenda, Cesar y Eloy, y al resto de mi familia, quienes han sido grandes pilares en mi vida al darme siempre su amor y apoyo incondicional, por siempre alegrarse de mis triunfos y escuchar atentamente mis fracasos para poder aconsejarme y alentarme a culminar satisfactoriamente todos mis proyectos. Por su gran ejemplo de temple, honestidad y perseverancia que me han dado a lo largo de mi vida. Mis Hijos Anuhar Omar y Edgar Said, que son el motor de mi vida, y que algún día deseo que rebasen mis logros y metas. Los amo viejos. :)

Igualmente, agradezco a el Dr. Luis Enrique Ramos Velasco y al Dr. Joel Suárez Cansino, directores, de este trabajo de investigación por su tiempo y conocimientos brindados, los cuales; me permitieron realizar satisfactoriamente este proyecto. Así mismo les agradezco por su amistad la cual me ayudo a crecer personal y profesionalmente.

A los sinodales Dr. Omar Jacobo Santos Sánchez y M. en C. Marco Antonio Marques Vera, que dedicaron tiempo para este trabajo de tesis sea de mejor calidad. Gracias...

Resumen

Control de un sistema subactuado empleando redes neuronales de pulso

En este trabajo de tesis se propone una nueva arquitectura de control basada en la inteligencia artificial, haciendo uso de las redes neuronales de pulso, también llamadas redes neuronales de tercera generación, ésta arquitectura se propone para el control de un sistema no lineal del tipo subactuado como lo es el carro péndulo invertido.

Abstract

Control of underactuated system with a Spike Neural Networks

In this work a spiking neural networks controllers are designed, which is basically a controller whose gains are adapted online using a spiking neural network; , to obtain several signals on a different scale-frequency, which are scaled and summed to generate a control signal which compensates for uncertainties system. It is noteworthy that this controller is an original contribution as a result of the research.

The validation of the two control algorithms proposed, are doing on numerical simulation, is performed on a system underactuated inverted-pendulum constructed by Quanser. The results obtained are also part of the contribution of the research since according to our literature search did not find these drivers applications to underactuated systems.

Índice general

Agradecimientos	VII
Resumen	IX
Abstract	XI
Índice general	XIII
Índice de figuras	XVII
Índice de tablas	XIX
1. Introducción	1
1.1. Descripción de la problemática	2
1.2. Planteamiento del problema	3
1.3. Hipótesis	3
1.4. Justificación	3
1.5. Resultados esperados	4
1.6. Organización de la tesis	5
2. Estudio del estado del arte	7
2.1. Introducción	7
2.2. Trabajos relevantes	7
2.3. Marco teórico	8
2.4. Comentarios	9
3. Redes neuronales de tercera generación	11
3.1. Introducción	11
3.1.1. La neurona real	13
3.1.2. Potencial de la membrana	13
3.1.3. La acción potencial	15
3.1.4. La sinapsis	15
3.1.5. Los pesos sinápticos	17
3.2. Modelos matemáticos de las neuronas de tercera generación	18
3.2.1. Modelos basados en conducta	19
3.2.1.1. El modelo Izhikevich	19

3.2.2.	Modelos umbral y disparo	20
3.2.2.1.	Integración y disparo perfecto	22
3.2.2.2.	Integración y disparo ideal	24
3.3.	Código neuronal	25
3.3.1.	Codificación por tasa	25
3.3.2.	Codificación temporal	27
3.3.3.	Codificación por población	27
3.4.	Métodos de aprendizaje	28
3.4.1.	Aprendizaje supervisado	28
3.4.1.1.	Método SpikeProp	29
3.4.1.2.	Método asociado a la entrada-salida	31
4.	Aplicación de las SNN al procesamiento digital de señales	33
4.1.	Introducción	33
4.2.	Señales determinísticas	35
4.3.	Señales heurísticas	37
4.4.	Seguimiento de señales médicas mediante SNN	39
4.4.1.	Aproximación de la señal EEG con una SNN con 5 SN en la capa oculta	40
4.4.2.	Aproximación de la señal EEG con una SNN con 10 SN en la capa oculta	41
4.4.3.	Aproximación de la señal ECG con una SNN con 5 SN en la capa oculta	43
4.4.4.	Aproximación de la señal ECG con una SNN con 10 SN en la capa oculta	44
5.	Aplicación de las SNN al control automático	47
5.1.	La SNN como sistema de control	48
5.2.	Control del carro péndulo invertido con una arquitecturas de una SN y diferentes configuraciones en la entrada	48
5.2.1.	La SNN con una arquitectura de una SN y una entrada	48
5.2.2.	La SNN con una arquitectura de una SN y tres entradas	52
5.2.3.	La SNN con una arquitectura de una SN y cinco entradas	55
5.3.	Control de carro péndulo invertido con una arquitectura de tres SN	60
5.4.	Control de carro péndulo invertido con una SNN	65
5.5.	Conclusiones	77
5.6.	Trabajos futuros	77
	Bibliografía	79
	A. Artículos sometidos	87

B. Algoritmos programados en MATLAB	103
B.1. MATLAB	103
B.1.1. Programa principal	103
B.1.2. Etapa de entrenamiento	112
B.1.3. Sumatorias	113
B.1.4. Modelo Izhikevich	115
B.1.5. Modelo de integración	115
B.1.6. Modelo Disparo Perfecto	116
C. Modelo carro-péndulo invertido	117
C.1. Sistemas no lineales subactuados	117
C.1.1. Sistema no lineal subactuado tipo carro péndulo invertido . .	118
C.1.1.1. Carro-péndulo invertido	119
C.1.1.2. Ecuaciones dinámicas del sistema carro-péndulo inver-	
tido	119
C.1.1.3. Representación en variables de estado	124
C.2. Comentarios	125

Índice de figuras

3.1. Sinapsis Neuronal	14
3.2. Señal Sináptica	16
3.3. IPSP,EPSP	17
3.4. Los Pesos sinápticos	18
3.5. Parámetros del modelo de Ihzikevich	20
3.6. Modelo Ihzikevich	21
3.7. Respuesta Modelo LIF	22
3.8. Modelo PIF	23
3.9. Modelo LIF	24
3.10. Codificación por Tasa	26
3.11. Codificación por Población	27
4.1. Esquema propuesto para la aproximación de señales.	34
4.2. Arquitectura propuesta de la SNN para la aproximación de señales.	35
4.3. Aproximación de la señal determinística, mediante la SNN con una arquitectura de 3 SN en la capa oculta.	36
4.4. Error de aproximación de la señal determinística, mediante la SNN con una arquitectura de 3 SN en la capa oculta.	37
4.5. Aproximación de la señal heurística, mediante la SNN con una arquitectura de 3 SN en la capa oculta.	38
4.6. Error de aproximación de la señal heurística, mediante la SNN con una arquitectura de 3 SN en la capa oculta.	39
4.7. Aproximación de la señal Electroencefalograma, mediante la red neuronal de pulso con una arquitectura de 5 SN en la capa oculta.	40
4.8. Señal de error existente entre la aproximación y la señal EEG.	41
4.9. Aproximación de la señal Electroencefalograma, mediante la red neuronal de pulso con una arquitectura de 10 SN en la capa oculta.	42
4.10. Señal de error existente entre la aproximación y la señal EEG.. . . .	42
4.11. Aproximación de la señal ECG, conSNN con una arquitectura de 5 SN en la capa oculta.	43

4.12. Señal de error existente entre la aproximación y la señal ECG.	44
4.13. Aproximación de la señal Electrocardiograma, mediante la red neuronal de pulso con una arquitectura de 10 SN en la capa oculta.	45
4.14. Señal de error existente entre la aproximación y la señal ECG.	45
5.1. Diagrama con una arquitectura de una SN y el error como entrada.	49
5.2. Respuesta del sistema con condiciones iniciales $x = 0, \dot{x} = 0, \theta = .1rad, \dot{\theta} = 0$, para una arquitectura de una SN y una entrada	51
5.3. Diagrama con una arquitectura de una SN, el error, derivada del error e integral del error como entradas.	52
5.4. Respuesta del sistema con con una arquitectura de una SN con tres entradas, con condiciones iniciales $p = 0, \dot{p} = 0, \dot{\theta} = 0$, y en $\theta = .1rad$	54
5.5. Diagrama con una arquitectura de una SN, y cinco entradas.	56
5.6. Respuesta del sistema con una aquitectura de una SN con cinco entradas y con condiciones iniciales $p = 0, \dot{p} = 0, \dot{\theta} = 0$, y $\theta = .1rad$	58
5.7. Diagrama de una SNN con una arquitectura con dos SN en la capa de entrada, y una SN en la capa de salida.	60
5.8. Respuesta del sistema con condiciones iniciales $p = 0$ y $\theta = .1rad$ con una arquitectura de tres neuronas y entrenamiento separado.	64
5.9. Diagrama general de control con una SNN.	65
5.10. Representación de la combinación de las variables x y θ	66
5.11. Respuesta del sistema con condiciones iniciales diferentes de cero con una arquitectura de tres capas, una de entrada, una capa de salida y una capa oculta con i numero de neuronas.	69
5.12. Respuesta del sistema con condiciones iniciales [$\dot{x} = 0, \dot{\theta} = 0, x = 0$ mts, $\theta = 0.2$ rad] con una arquitectura de tres capas, una de entrada, una capa de salida y una capa oculta con i numero de neuronas.	72
5.13. Respuesta del sistema con condiciones iniciales [$\dot{x} = 0, \dot{\theta} = 0, x = 0.1$ mts, $\theta = 0.1$ rad] con una arquitectura de tres capas, una de entrada, una capa de salida y una capa oculta con i numero de neuronas.	74
5.14. Respuesta del sistema con condiciones iniciales [$\dot{x} = 0, \dot{\theta} = 0, x = 0.05$ mts, $\theta = 0.2$ rad], con una arquitectura de tres capas, una de entrada, una capa de salida y una capa oculta con i numero de neuronas.	76
C.1. Sistema mecánico subactuado tipo carro péndulo.	119
C.2. Diagrama de cuerpo libre correspondiente al carro.	120
C.3. Diagrama de cuerpo libre correspondiente al péndulo.	121

Índice de tablas

3.1. Algoritmo SpikeProp	30
5.2. Descripción de las entradas de la SN.	59
5.1. Arquitectura propuestas	78
C.1. Variables y parámetros del sistema carro - péndulo invertido.	127

Capítulo 1

Introducción

El estudio y aplicación de las Redes Neuronales Artificiales (RNA) en las áreas de cómputo inteligente para el reconocimiento de patrones, en el procesamiento e identificación de imágenes y en las áreas de control, han tomado un auge importante debido principalmente a su capacidad de aprendizaje y su gran versatilidad en la identificación de sistemas, sin embargo, actualmente existen redes neuronales artificiales de tercera generación, también llamadas de pulso, que forman parte del desarrollo de las RNA y son una alternativa para control no lineal, [39].

Los sistemas de control usados en la industria generalmente basados en el controlador PID clásico, tienen una gran diversidad de aplicaciones donde se incluyen el control de máquinas eléctricas con diferentes características mecánicas y configuraciones eléctricas que hacen que la operación y el funcionamiento sea adecuado, sin embargo con el paso del tiempo los sistemas sufren cambios debido a desgastes mecánicos y modificaciones en las partes eléctricas [66]. El advenimiento de nuevas tecnologías ha permitido que sea posible aplicar técnicas de control más complejas, esto con la finalidad de mejorar el desempeño de los sistemas de control en lazo cerrado. Muchas de estas técnicas requieren el conocimiento pleno del modelo matemático que en la mayoría de los casos no se tiene o se tienen dificultades para obtenerlo, pasando un gran número de horas hombre en la obtención de dicho modelo [40]. Aunado a esto los sistemas dinámicos en la industria sufren cambios en sus componentes, lo cual hace necesario sintonizar continuamente las leyes de control, surgiendo el control adaptable [36]. Muchos de estos controles adaptables hacen uso del modelo dinámico del sistema a controlar siendo ésta una problemática a ser considerada.

Para ayudar a solventar esta problemática, actualmente existen técnicas de identificación de sistemas que hacen uso de las RNA,[40].

La continuidad en los estudios, las aplicaciones y el desarrollo de los algoritmos

hacen que este trabajo sea la continuidad de una investigación en el ámbito del control utilizando redes neuronales de pulso, solo que en sistemas lineales, observando un comportamiento deseado para estos sistemas[39].

Razones por las cuales surge la idea de proponer un algoritmo de control auto ajustable, el cual no sólo eliminará la tediosa tarea de obtener el modelo matemático de la planta, sino que no importa cuánto cambie la dinámica de la planta o la señal de referencia, la señal de control siempre se irá adaptando de acuerdo a las condiciones exigidas. Para comprobar la eficacia del algoritmo propuesto con redes neuronales de pulso en sistemas reales, se realiza las pertinentes pruebas con sistemas subactuados, en este caso el péndulo invertido.

1.1. Descripción de la problemática

Existen diferentes alternativas para controlar un sistema. Estas alternativas incluyen el modelado del sistema a través de ecuaciones diferenciales, teoría de la probabilidad o métodos estocásticos, o bien el análisis, diseño e instrumentación de modelos basados en redes neuronales artificiales, modelos difusos, o modelos neuro-difusos, entre otros.

Cuando se opta por las alternativas que ofrecen las redes neuronales artificiales y la lógica difusa, entre otras, se requiere de una fase de entrenamiento que puede ser supervisado o no supervisado. En el aprendizaje supervisado, se requiere, además de los datos de entrada, la existencia de un supervisor que de alguna manera indica si las salidas del sistema son las correctas o no son las correctas. Generalmente, la definición del supervisor se obtiene, en parte, especificando la salida deseada para cada una de las entradas al sistema, las cuales son parte del conjunto de datos de entrenamiento. De esta forma, si $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ define el conjunto de datos de entrada para el entrenamiento, entonces se debe tener el correspondiente conjunto de datos de salida $\{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n\}$ para el entrenamiento. La dimensión de los datos de salida no necesariamente es igual a la dimensión de los datos de entrada. La definición del supervisor continúa ahora a través de la construcción de una función de error cuadrado total, $E^2(x, y, w)$. Donde w son los pesos a modificar con el entrenamiento. La determinación de los parámetros w se logra a través del entrenamiento supervisado. En el caso de las redes neuronales artificiales de segunda generación, esto se logra encontrando el gradiente de la función $E^2(x, y, w)$. Sin embargo, una aplicación directa de esto conduce a problemas de convergencia durante el proceso de entrenamiento, motivo por el cual se han sugerido variantes al esquema original: momentum, tasa variable de aprendizaje, gradiente conjugado, algoritmo de Levenberg-Marquardt, entre otros

[26].

1.2. Planteamiento del problema

Los sistemas no lineales en forma general tienen un comportamiento no homogéneo además de no estar sujetos al principio de superposición. Por lo tanto el diseño de un control para este tipo de sistemas no es trivial, más aún, cuando se requiere que el control adapte sus parámetros en línea. En los sistemas de control clásicos es común que no se logre esta situación, además de que cuando se aplican a los sistemas no lineales del tipo subactuado, los algoritmos convergen de forma lenta a la referencia deseada.

1.3. Hipótesis

Las características no lineales de una red neuronal de tercera generación (Redes Neuronales de Pulsos, Spiking Neural Network o SNN) y los resultados obtenidos con anterioridad en sistemas lineales, hacen que este modelo sea el candidato para analizar e instrumentar un control inteligente que reduzca el tiempo de asentamiento en sistemas no lineales subactuados. Desconociendo el número de neuronas implementadas para la resolución de un problema, el número de iteraciones requeridas para converger a los parámetros apropiados durante la fase de entrenamiento, la precisión con que se determinan estos parámetros y la capacidad de generalización y el tiempo de clasificación, para el caso de aprendizaje supervisado.

1.4. Justificación

La opción de controlar un sistema a través del modelado de su funcionamiento por medio de ecuaciones diferenciales, tiene el problema de que se debe proponer adecuadamente el sistema en ecuaciones diferenciales que lo describe correctamente. Este no es un problema trivial, ya que se requiere de un conocimiento profundo de las posibles fuerzas y variables dinámicas, así como de los valores de los diferentes parámetros y la interpretación física de éstos. Sin mencionar las condiciones iniciales y de frontera adecuadas, así como la posible dificultad en encontrar una solución analítica a las ecuaciones diferenciales, o la posible dificultad en instrumentar un método numérico apropiado.

En la actualidad los modelos que mejor representan al comportamiento de las redes neuronales biológicas están dados por las redes neuronales de pulsos, este tipo de

modelos tienen un sustento matemático basado en ecuaciones diferenciales y haciendo uso de sus capacidades de aprendizaje para la identificación de plantas dinámicas, se puede usar en forma efectiva en el control de éstos sistemas.

La aplicación de redes neuronales de tercera generación ha conducido a proponer modificaciones al algoritmo de Levenberg-Marquardt, las cuales ofrecen una mejora en la tasa de convergencia cuando se compara con el método Levenberg-Marquardt (LM) tradicional y tiene un menor costo computacional, además de que requiere una menor cantidad de memoria [74]. También se ha propuesto el algoritmo denominado SWAT (Synaptic Weight Association Training) que se construye a partir de la combinación del método BCM (Bienenstock-Cooper-Munro) y del método STDP (Spike Timing Dependent Plasticity), el cual garantiza una convergencia de hasta un 96.2%, que es mucho mayor en comparación con otros algoritmos de entrenamiento para redes neuronales de pulso [72].

Otro algoritmo de entrenamiento para redes neuronales de tercera generación es CS (Cuckoo Search Algorithm) el cual es un novedoso método meta-heurístico basado en un comportamiento anidado, este algoritmo ha sido aplicado para la solución de problemas de optimización y promete grandes resultados en el campo de las redes neuronales artificiales, especialmente durante el ajuste de los pesos sinápticos [70].

Además aprovechando el método de supervisión remota llamado ReSuMe (Remote Supervised Method), el cual bajo la combinación de tres neuronas (neurona de entrada, neurona de aprendizaje y la neurona de entrenamiento de pulsos deseados), es posible obtener un tren de pulsos deseados en las neuronas de aprendizaje con una precisión muy alta [56].

1.5. Resultados esperados

A lo largo de esta tesis se irán presentando los objetivos específicos que sirvieron como peldaños para ir avanzando y finalmente alcanzar el objetivo global, este proceso se realizó con el fin de obtener indicadores para saber que se avanzaba por buen camino.

Objetivo general

Diseñar e implementar un algoritmo de control inteligente, para controlar un sistema no lineal del tipo subactuado (carro péndulo invertido), mediante las redes

neuronales de pulso.

Objetivos específicos

Los objetivos específicos de la tesis son los siguientes:

- **Controlar un sistema dinámico del tipo subactuado en simulación**, uno de los objetivos específicos más relevantes es el de unir el conocimiento de las redes neuronales de pulso, con el área de la ingeniería de control, siendo esta, la clave para que la tesis rinda buenos resultados. Este tipo de sinergia se ha llevado a cabo utilizando redes neuronales de segunda generación en conjunto con un control PID [77][37] o utilizando el error generado por la diferencia entre la salida de la planta retroalimentada y la referencia, como el propio error de la red, este control será llevado a cabo con todos los algoritmos de entrenamiento previamente mencionados [25].
- **Comparar los tiempos y exactitud de convergencia** para el control de sistemas no lineales entre redes neuronales artificiales de tercera generación y algoritmos existentes. Este objetivo particular se realizará para poder corroborar que la implementación de redes neuronales de tercera generación ofrece resultados que superan a los algoritmos convencionales e incluso a las redes neuronales de segunda generación, para un mismo problema, enfocados al tiempo de convergencia a una referencia deseada.

1.6. Organización de la tesis

En el Capítulo 2, *Estudio del estado del arte*, se analizan los trabajos relacionados con el desarrollo de esta investigación. En el Capítulo 3, *Redes neuronales de tercera generación*, se profundiza en el tema de las redes neuronales de tercera generación, y se da una clara revisión a los trabajos que sobresalen para el diseño y desarrollo de arquitecturas de redes neuronales de tercera generación con sus variantes adaptables, enfocadas a la implementación en el control de sistemas dinámicos subactuados que se muestran en el Apéndice C, los resultados que se presentan en este trabajo han sido divididos en dos Capítulos, mostrando en el Capítulo 4 resultados aplicados a procesamiento digital de señales, y en el Capítulo 5 se muestran resultados aplicados al control de una planta no lineal.

Capítulo 2

Estudio del estado del arte

2.1. Introducción

Desde muchas décadas atrás hasta la actualidad el desarrollo de las técnicas de control para sistemas electromecánicos es uno de los campos de investigación más estudiados de la ingeniería, esto debido a la problemática existente en los procesos industriales, y a los nuevos desarrollos de sistemas, que requieren de nuevas técnicas de control para mejorar la eficiencia y comportamiento de los mismos. En esta tesis se aplica una propuesta de control para los sistemas no-lineales en específico los sistemas electromecánicos subactuados y en especial el carro péndulo invertido.

2.2. Trabajos relevantes

Las redes neuronales artificiales en el área de control, ha ido en auge en los últimos años, esto debido a su gran capacidad de adaptación y gran flexibilidad para la identificación y control de sistemas dinámicos, no importando su orden. Un arreglo de redes neuronales artificiales de segunda generación puede controlar una planta no lineal adaptándose a cambios suaves en el modelo [9], pero si la señal de referencia y/o el cambio del modelo de la dinámica del sistema, no son derivables temporalmente, el control ocasiona que la variable de proceso tenga un sobreimpulso el cual podría llevar al sistema a una zona inestable, para evitar esto, se ha complementado la teoría de redes neuronales con controles lineales. Debido a estas necesidades, las redes neuronales han sido enfocadas a la mejorara de las leyes de control de ganancias ajustables, las cuales ofrecen una mejor respuesta en comparación con teorías de auto ajustes clásicos como cancelación y reasignación de polos [63], además se ha demostrado que son superiores en lo que se refiere al tiempo y valor de convergencia contra técnicas más complejas como lo es la sintonización de ganancias por Ziegler-Nichols o LGR [10]. Incluso el control GPIDNN soporta de una mejor manera los retardos

inherentes del sistema, antes de que este se haga inestable [76], esto por la mayor velocidad de procesamiento de las redes neuronales artificiales de segunda generación en comparación con las técnicas clásicas antes mencionadas. Pero ahora con esta nueva generación de redes neuronales se demuestra que no es necesario complementarlas con algún otro tipo de controlador como lo es el PID pues sólo con una neurona de tercera generación se puede controlar un sistema de primer orden [39]. La implementación de redes neuronales de tercera generación para el control de una planta no lineal se ha realizado en simulación. Por lo que esta tesis no sólo busca llevar a una implementación física de una planta, sino que se aventura a modificar la arquitectura convencional para optimizar el control.

Los métodos para entrenar las redes neuronales de tipo Spike, contiene ventajas y desventajas. La velocidad de entrenamiento y la exactitud de la señal de salida con respecto a la señal deseada son factores importantes en este trabajo. Por lo que se implementarán [56], [70], [51], [72], [74] para la identificación y control de un sistema físico en tiempo real, ambos métodos manejan de forma superior tanto el tiempo de convergencia, como la precisión del valor de salida con respecto a otros tipos de entrenamiento para clasificación y/o salidas condicionadas en neuronas de pulso, aunque en este trabajo estos métodos serán aplicados a la identificación de señales médicas, se tiene plena confianza que se conserven las características y cualidades de estos entrenamientos.

Uno de los trabajos relevantes en este tema son los desarrollados por A. Jiménez. et al. donde se controla un motor de corriente continua con un FPGA [51], y el desarrollado por el M. en C. L. Lechuga, que aborda también el tema de control de velocidad y posición de un motor de corriente continua [39], trabajo que sirve de antecedente a esta tesis, cambiando el sistema del motor de CD a un sistema subactuado.

2.3. Marco teórico

Las bases teóricas necesarias para resolver la problemática, apuntan al estudio a los sistemas dinámicos, tanto lineales [38] como no lineales [15], para comprender su comportamiento y las implicaciones de un control en sistemas dinámicos por redes neuronales de tercera generación, lo cual nos lleva al estudio sobre los controles clásicos involucrados [53], por otra parte, se estudiarán las dinámicas de las redes neuronales de segunda [13] y tercera generación [22], y más específicamente, con un enfoque a la identificación de sistemas en tiempo real [25], [1]. Una vez obtenido el algoritmo de control, se realizará la comparación en simulación con un sistema no lineal por lo que será necesario conocer acerca del modelado de sistemas dinámicos [41].

Para la etapa de implementación física del control se requieren conocimientos sobre dos áreas específicas, electrónica, de la cual surgirá la etapa de potencia.[6] y programación sobre la cual radica la elaboración de las leyes de control sobre redes neuronales de tercera generación [16].

2.4. Comentarios

En la actualidad, el ser humano ha invertido recursos materiales y humanos con la finalidad de comprender el funcionamiento de un cerebro de un ser vivo, el del ser humano, teniendo en su interior millones de neuronas conectadas entre sí. Uno de los resultados obtenidos, es el de emular el comportamiento de una neurona matemáticamente obteniendo características importantes como que aprenden, recuerdan y evolucionan con una velocidad y exactitud superior a sus antecesoras de segunda generación.

Por otro lado, las principales desventajas de las redes neuronales de tercera generación son:

- **La poca información sobre el tema.** Las redes neuronales de tercera generación, son un área nueva de investigación activa, surgieron alrededor del 2000, mejorando cada día su comprensión, pues aún se siguen diseñando algoritmos de entrenamiento y modelos neuronales, más eficientes en tiempo de convergencia y costo computacional, sin contar que las aplicaciones en la ingeniería de control son escasas.
- **La salida de las neuronas son acotadas respecto a sus entradas.** Aunque las redes neuronales de tercera generación tienen una gran flexibilidad a la adaptación, su salida depende directamente de la entrada al menos en la codificación temporal, es decir la salida codificada de una neurona en su ventana de tiempo puede no cubrir todo los valores de ésta, solo un sub conjunto de la misma.
- **La taza de aprendizaje se elige de forma heurística.** En las redes neuronales en general, ésta ha sido siempre un tema a tratar debido a que la forma de seleccionar este parámetro depende de estar modificando hasta obtener un resultado favorable.

Capítulo 3

Redes neuronales de tercera generación

Las redes neuronales spike o de pulsos han sido llamadas redes neuronales artificiales de tercera generación cuando la información viaja en pulsos temporales. Existen diversos tipos de modelos matemáticos que representan la dinámica de las neuronas, los cuales son categorizados en base a su nivel de abstracción, de igual forma los métodos de entrenamiento pueden ser separados en dos bloques principales, entrenamiento supervisado y no supervisado. Esta tesis se enfoca en el entrenamiento supervisado, donde una nueva arquitectura es propuesta, para el control de sistemas subactuados, basada en una retroalimentación a la red neuronal dependiente del error, generado por la diferencia de la señal de referencia y la salida del sistema.

3.1. Introducción

Las redes neuronales de tercera generación han sido una importante parte de la inteligencia artificial y han sido usadas extensamente en algoritmos de reconocimiento, diagnóstico médico, análisis de imágenes, finanzas, predicción del clima, y muchas de las ciencias computacionales y por supuesto en tareas de ingeniería, aunque a diferencia de los procesos computacionales convencionales son implementadas masivamente con una distribución paralela, para ser capaz de aprender a través de su capacidad de plasticidad sináptica. Wolfgang en uno de sus trabajos publicados [46], divide las redes neuronales artificiales dentro de tres generaciones, basadas en qué tanto se acercan éstas a las neuronas biológicas reales.

La primera generación de las redes neuronales artificiales con un modelo neuronal de umbral fue usada por McCulloch-Pits en 1943. Su modelo conocido también como perceptrón solo tenía dos estados (“alto” o “bajo”), basadas en la suma de las señales

de entrada multiplicadas por sus pesos. Este tipo de redes neuronales podía ser implementado computacionalmente por funciones booleanas.

La segunda generación de redes neuronales artificiales utiliza funciones de activación sigmoideas con cambio de umbral, en lugar de solo lógica booleana, esta segunda generación es mucho más poderosa que las neuronas de primera generación, y desde la era de la computación digital, el número de neuronas para resolver el mismo problema se ha reducido [48]. Además, en esta generación surge un nuevo algoritmo de aprendizaje supervisado para su implementación, llamado propagación del error hacia atrás, basado en el gradiente descendiente, el cual va ajustando los pesos proporcionalmente al error que existe a la salida. Desde un punto de vista biológico la segunda generación de redes neuronales artificiales es más realista que la primera generación pues una salida que se comporta de forma sigmoidea se acerca mas al disparo que se encuentra en las neuronas biológicas, al menos más que una instrucción booleana [46].

Las neuronas situadas en la corteza cerebral tienen un funcionamiento más rápido en analogía con la implementación computacional, pues el reconocimiento facial le toma al cerebro humano alrededor de 100ms, esto significa que el tiempo de procesamiento por neurona hacia sus millones de neuronas periféricas no sobrepasa los 10ms [69]. Los resultados experimentales muestran que la ventana de tiempo para la codificación en el cerebro es muy pequeña. Esto no significa que la codificación no sea significativa biológicamente hablando. Esta aseveración puede ser probada experimentalmente, por ejemplo en el caso de los receptores de presión en una sanguijuela [35], donde mayor número de pulsos produce un estímulo mayor, durante una estimulación periódica.

Investigaciones muestran que las neuronas reales se comunican por pulsos, conocidos como acción potencial, y que todos los pulsos producidos en el cerebro son muy parecidos entre sí, los cuales están contenidos en una ventana de tiempo de 1 a 2 ms de duración y con una amplitud alrededor de los 100mV [22]. La información es codificada en relación al tiempo en el que ocurren los pulsos y no por el valor de los pulsos en sí, todos estos resultados experimentales brindaron la información necesaria para que surgiera la tercer generación de redes neuronales artificiales. Las cuales se acerca aún más al comportamiento biológico real. Pues sus señales codifican información temporal, lo cual significa que manejan información espacio-temporal. Wolfgang [45], en uno de sus trabajos, muestra que es posible simular disparos sigmoideos con información de pulso temporales.

3.1.1. La neurona real

Todas las neuronas reales, normalmente están divididas en tres partes; las dendritas, la soma y el axón, además de un proceso propio de éstas llamado sinápsis. Generalmente, las dendritas son quienes reciben la información de las neuronas periféricas conectadas a ésta, la soma es donde todas las señales recibidas de las dendritas se acumulan y procesan, y finalmente el axón es el segmento de la neurona donde las señales de salida son emitidas y conducidas hacia las dendritas de las próximas neuronas. El proceso de la sinapsis ocurre siempre entre dos neuronas y es el momento justo donde la información (pulso eléctrico) de una neurona pasa a otra; si la neurona “J” emite un pulso que cruza la sinapsis hacia la neurona “I”, entonces se dice que la neurona que emite el pulso (la neurona “J”) es una neurona pre sináptica y consecuentemente la neurona que recibe el pulso (la neurona “I”) es llamada una neurona post sináptica.

Hodgkin y Huxley [30] encontraron, de manera experimental que el axón es el lugar donde la información de la soma, es codificada [47], ver la Figura 3.1

3.1.2. Potencial de la membrana

Todas las neuronas están rodeadas por iones positivos y negativos, en la superficie de la membrana existe una excesiva carga negativa y en el exterior de la membrana hay un exceso de carga positiva, ambas cargas conforman el potencial de la membrana.

El potencial de cada membrana puede ser calculado por la siguiente ecuación:

$$V_m = V_{in} - V_{out}, \quad (3.1)$$

donde: V_{in} es la carga negativa dentro de la célula

V_{out} es la carga positiva fuera de la célula

Cuando el potencial de la membrana se encuentra en reposo, es decir cuando no ha recibido ninguna señal de entrada, se le conoce como potencial de reposo y normalmente se encuentra alrededor de los -60 mV a los -70 mV.

El fenómeno en el cual el potencial de la membrana rebasa el potencial de reposo es llamado depolarización. Al fenómeno opuesto se le conoce como hiperpolarización. Cuando la depolarización rebasa el valor crítico, conocido también como umbral, la célula produce una acción potencial (un pulso) [35], Figura 1.2, Si el potencial de membrana recibe un impulso esto causará depolarización o hiperpolarización y

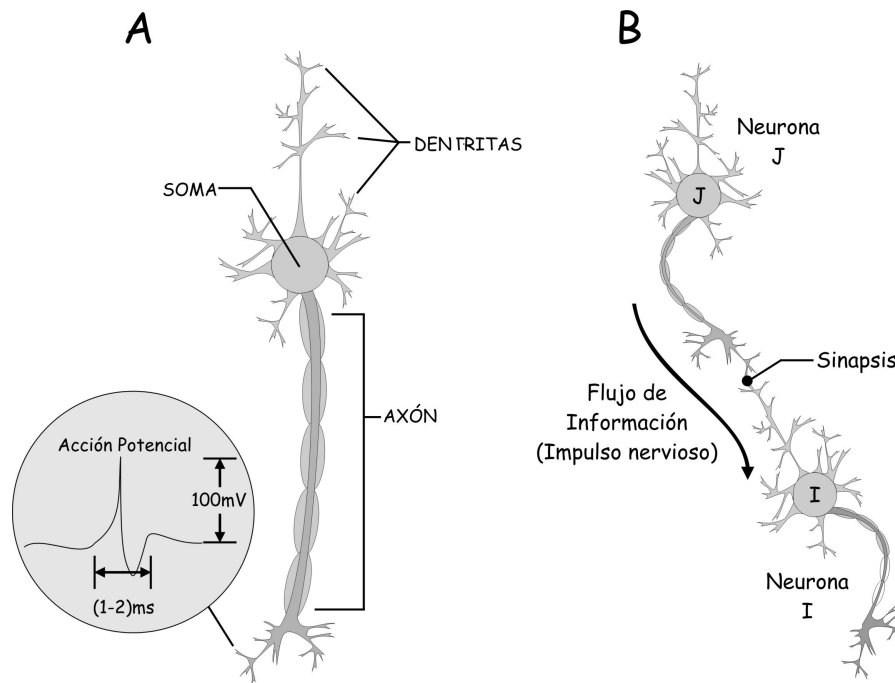


Figura 3.1: **A.** Se muestra un ejemplo de la acción potencial neuronal, la cual prácticamente es un pulso de voltaje con una duración de 1-2 ms, y una amplitud de 100 mV. **B.** Transmisión de la señal (información) de la neurona pre sináptica J a la neurona post sináptica I, donde la sinapsis entre estas neuronas es denotada por el círculo negro [22].

si después de este suceso, la neurona no experimenta ninguna señal de entrada, el potencial de la membrana regresará lentamente a su potencial de reposo, recordemos que esto ocurre en las neuronas biológicas reales y no imperiosamente en los modelos matemáticos propuestos.

3.1.3. La acción potencial

Como se ha visto anteriormente, cuando el potencial de membrana rebasa el valor crítico llamado umbral θ se emite una acción potencial, también conocido como pulso, esto es causado por el movimiento de iones que cruza la membrana a través de algún canal en una ventana de voltaje [35]. El pulso generado en una neurona es idéntico al de cualquier otra, así como la señal que se mueve de la neurona pre sináptica a la neurona post sináptica [22]. El conjunto de tiempos en los que ocurren los disparos de alguna neurona es nombrado como tren de pulsos y se puede representar por la siguiente ecuación

$$F_i = t_i^1, t_i^2, \dots, t_i^{(f)}. \quad (3.2)$$

El subíndice i define el número de neurona y el superíndice es el pulso emitido por la neurona, siendo ' n ' el pulso más reciente emitido por la n ésima neurona. Exactamente después de la transmisión del pulso de la neurona pre sináptica a la neurona post sináptica, el potencial de la membrana se desploma y ocurre una fase de alta hiperpolarización debajo del valor del potencial de reposo para después lentamente regresar al potencial de reposo. Durante ese tiempo no es posible que la neurona emita un segundo pulso, no importa que tan fuerte sea el conjunto de señales de entrada. Esto biológicamente ocurre porque los canales de iones se abren instantáneamente cuando un pulso es generado [22]. El tiempo mínimo que transcurre entre dos pulsos emitidos consecutivamente por la misma neurona se denomina periodo de refracción absoluta y el fenómeno que ocurre cuando el potencial de membrana está por debajo del potencial de relajación, esto es después de que un pulso es emitido, se conoce como SAP (Spike After Potencial), como se observa en la Figura 3.2

3.1.4. La sinapsis

Entre el axón de la neurona pre sináptica y las dendritas de la neurona post sináptica existe un pequeño espacio vacío, conocido como espacio sináptico, la operación de la sinapsis es muy compleja y una descripción más detallada se encuentra en [35], en esta tesis sólo los aspectos más relevantes se tomarán en cuenta.

El pulso de la neurona pre sináptica no puede cruzar por sí mismo el espacio sináptico, por lo que cuando la neurona pre sináptica emite un pulso, éste espacio

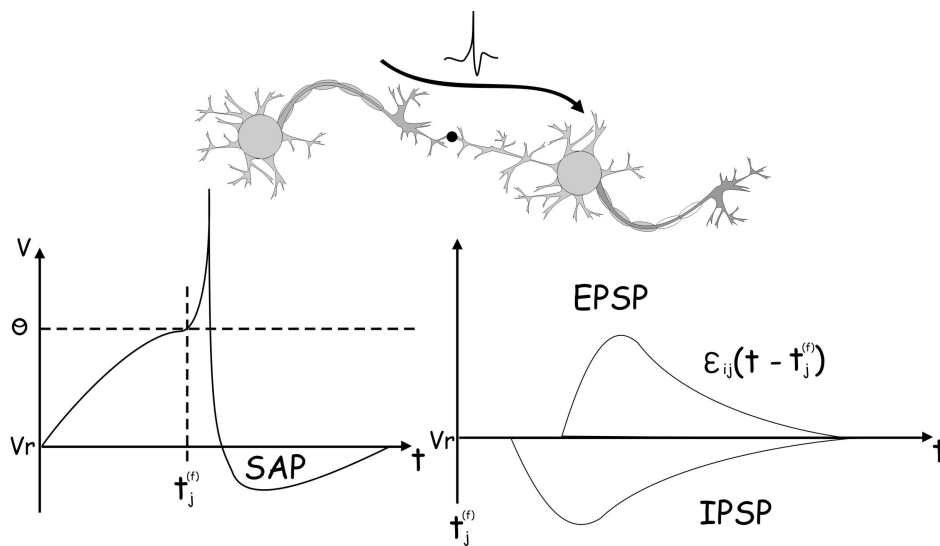


Figura 3.2: El potencial de membrana va incrementando y en el tiempo $t_j^{(f)}$ el potencial de la membrana rebasa el umbral y esto produce un pulso [47], posteriormente la neurona se relaja y su potencial de membrana se coloca debajo del potencial de reposo y va regresando lentamente al potencial de receso. El fenómeno de la sinapsis se puede dividir en un potencial de excitación post sináptico (EPSP) $E_{ij}(t - t_j^{(f)})$ y un potencial inhibitorio post sináptico (IPSP), θ es el umbral de disparo, V_r es el coltaje de relajación

sináptico es llenado de un líquido, el cual genera un potencial post sináptico (PSP) hacia las dendritas de la neurona post sinápticas [35]. Este proceso no ocurre instantáneamente; hay un pequeño retardo generado en cada sinapsis.

Existen dos tipos de potenciales post sinápticos, si el potencial post sinápticos generado es positivo se le llama potencial post sináptico de excitación (EPSP), y si el potencial post sináptico es negativo, se le denomina potencial post sináptico de inhibición, dichos potenciales se muestran en la Figura 3.3. Un IPSP disminuye el potencial de membrana de la neurona post sináptica mientras que el EPSP incrementa el potencial de membrana, y si es lo suficientemente grande, la neurona post sináptica puede disparar.

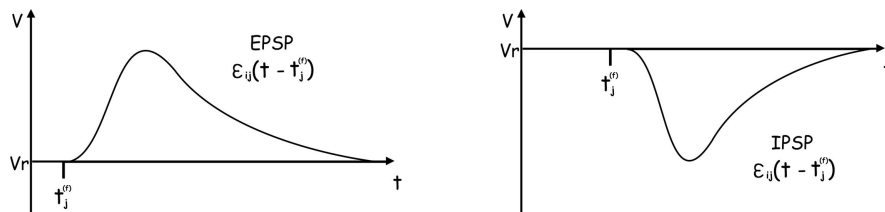


Figura 3.3: Potencial post sináptico de excitación (EPSP) y el potencial post sináptico de inhibición (IPSP), ambos de una neurona biológica [46].

3.1.5. Los pesos sinápticos

Cada sinapsis va acompañada por un factor que incrementa o decrementa la señal de salida de la neurona pre sináptica, este factor es llamado peso sináptico, el cual es el responsable de que el EPSP y el IPSP saturen o relajen a la neurona con mayor o menor rapidez, por lo que la modificación de estos pesos sinápticos perturba directamente la salida de la neurona y son directamente los responsables de que la neurona y por consiguiente la red neuronal modifique su salida, pues representan la variable más dominante en el proceso de aprendizaje.

Matemáticamente, tanto los pesos sinápticos como los retardos relacionados a cada sinapsis, son usados para entrenar a la red neuronal, en cada par de neuronas pre sinápticas “ i ” a sus respectivas neuronas post sinápticas “ j ”. Recientes resultados experimentales, han mostrado que entre todas las interconexiones de dos neuronas biológicas puede haber múltiples sinapsis con múltiples retardos [61],[14], como se ilustra en la Figura 3.4. Esta pensamiento ha sido usado en numerosos trabajos aprovechando dichos parámetros para mejorar los algoritmos de entrenamiento.

En la Figura 3.4 la capa de neuronas post sinápticas J recibe el conjunto de entradas Γ_j (Señales pre sinápticas de la j -ésima neurona). El acercamiento entre la

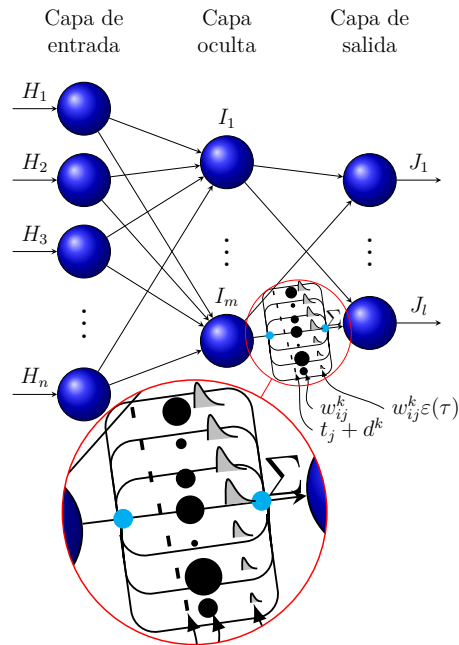


Figura 3.4: Modelo de una red neuronal simple.

inter conexión de la neurona i -ésima a la neurona j -ésima muestra que cada conexión entre dos neuronas consiste en “ m ” terminales sinápticas retardadas. Cada terminal sináptica “ k ” constituye el retardo k -ésimo (d^k) la cual es asociada con un peso sináptico k -ésimo entre la neurona i -ésima a la neurona j -ésima (w_{ij}^k) [5]. Los círculos negros dictan la magnitud de los pesos sinápticos.

3.2. Modelos matemáticos de las neuronas de tercera generación

Los modelos matemáticos que representan el comportamiento de las neuronas de tercera generación pueden ser divididos en dos grandes categorías [22] basándose en su nivel de abstracción; modelos de conducta y modelos de umbral. Los modelos de conducta incorporan a los canales de iones de la célula para describir el comportamiento, mientras que los modelos de umbral representan el comportamiento dinámico de las neuronas biológicas reales de una manera más abstracta, donde el umbral del voltaje tiene un valor fijo y la neurona dispara cada vez que el potencial de la membrana rebasa dicho umbral.

3.2.1. Modelos basados en conducta

En general, los modelos basados en conducta han sido derivados de los ganadores del premio Nobel (1963) Hodgkin y Huxley [47], basados en su experimentos ellos modelaron el funcionamiento de un gran axón de un calamar [30]. Su trabajo básicamente, describe que ocurre con los canales de iones de la célula [39].

3.2.1.1. El modelo Izhikevich

Izhikevich [33] creó un modelo matemático para describir y emular el comportamiento de las neuronas biológicas de la corteza cerebral de una rata, el cual combina la dinámica del modelo de Hodgkin-Huxley y la eficacia computacional de los modelos umbral y disparo. Esto quiere decir que se redujo el modelo de cuatro dimensiones de Hodgkin-Huxley a un modelo de dos ecuaciones diferenciales no lineales de primer orden:

$$\begin{aligned}\dot{v} &= 0.04v^2 + 5v + 140 - u + I \\ \dot{u} &= a(bv - u)\end{aligned}\tag{3.3}$$

$$\begin{aligned}\text{si; } v &= 30mV \\ \text{entonces; } v &\leftarrow c, u \leftarrow u + d\end{aligned}\tag{3.4}$$

La variable v representa el potencial de membrana de la neurona y u representa la variable de recuperación de membrana, I la cual es la corriente de iones de activación de potasio (K) y la inactivación de la corriente de iones de sodio (Na). Este modelo puede exhibir las propiedades más importantes sobre el comportamiento neuronal con las variables apropiadas a, b, c y d . Este modelo tiene un umbral dinámico que depende del estado previo del potencial de membrana anterior al pulso. A continuación se muestra los parámetros de la ecuación (3.4) [33]:

- El parámetro a describe la escala de tiempo de la variable de recuperación de membrana u , valores pequeños repercuten en recuperación lenta. El valor típico es $a = 2$.
- El parámetro b describe la sensibilidad de la variable de recuperación u del valor de reposo del potencial de membrana v . El valor típico es $b = 0.2$.
- El parámetro c determina el valor del potencial de reposo de la membrana después del pulso emitido. El valor típico que manejan las neuronas reales es de $c = -65mV$.

- El parámetro d determina el valor de reposo de la variable de recuperación u , dictado por el valor de conductancia del sodio (Na) y potasio (K), su típico valor es $d = 2$.

En la Figura 3.5 los parámetros del modelo neuronal propuesto por Izhikevich pueden ser observados, y en la Figura 3.6 se muestra el potencial de membrana, la variable de recuperación y la señal de entrada.

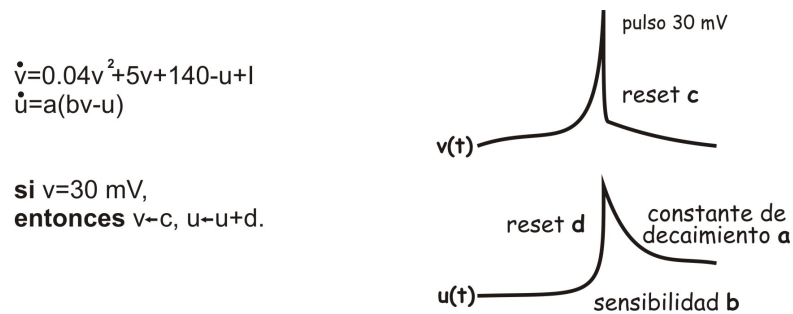


Figura 3.5: Se muestra las ecuaciones diferenciales que describen el comportamiento del modelo de Izhikevich, así como su representación gráfica.

3.2.2. Modelos umbral y disparo

Uno de los modelos más usados extensamente dentro de los modelos umbral y disparo, es el modelo de integración y disparo implementado en un extenso número de trabajos [12], esto debido a su fácil implementación y bajo costo computacional.

Los modelos umbral y disparo representan dinámicas con un alto nivel de abstracción. Estos modelos se basan en la sumatoria de todas las contribuciones de las neuronas pre sinápticas al potencial de membrana, y si el potencial de membrana rebaza el umbral fijado previamente, la neurona disparará.

Este tipo de modelos Integración y disparo se puede dividir en dos sub modelos: Integración y disparo perfecto (PIF) e Integración y disparo ideal (LIF)

La ecuación diferencial de primer orden que describe el comportamiento de los modelos de integración y disparo no describe completamente el comportamiento de las neuronas artificiales de tercera generación, pues solo comprende la dinámica desde las condiciones de reposo del potencial de membrana hasta antes del momento en que la neuronal dispara (instante en que se emite el pulso de salida), por lo que la condición de la actividad de disparo debe ser planteado. Esto es expresado en la ecuación

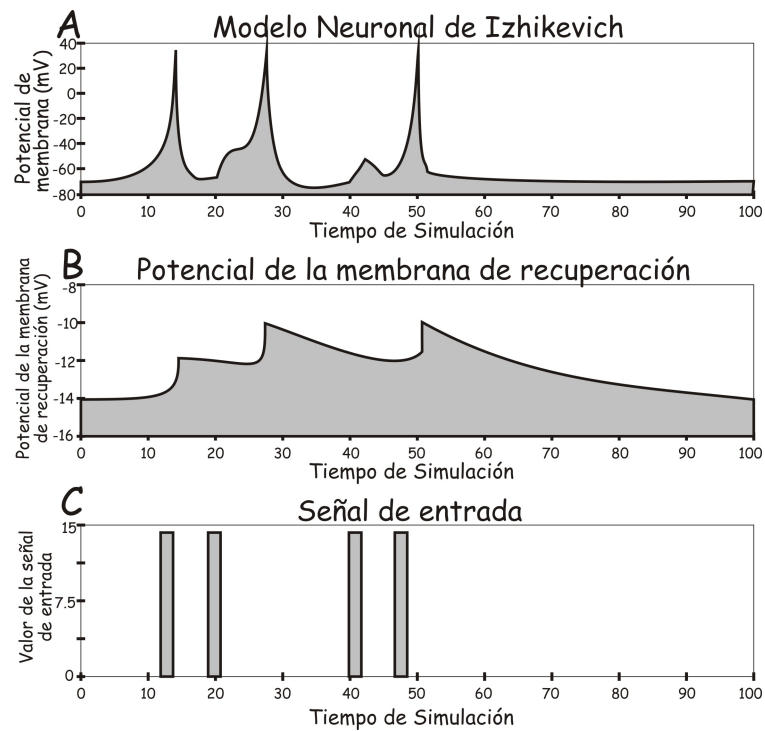


Figura 3.6: En A se presenta la respuesta del potencial de membrana del modelo de redes neuronales de tercera generación propuesto por Izhikevich, en B se grafica la variable de recuperación de la membrana y en C se muestran los pulsos y valor de entrada a la neurona.

(3.5), donde representa el momento justo en el que el valor del potencial de membrana supera el umbral θ fijado previamente, esto es descrito como tiempo de disparo

$$t^{(f)} : u(t^{(f)}) = \theta \quad \text{y} \quad \left. \frac{du(t)}{dt} \right|_{t=t^{(f)}} > 0 \quad (3.5)$$

En el modelo integración y disparo la acción potencial (pulso) no está descrito de la misma manera que en el modelo neuronal Hodgkin-Huxley. Los pulsos en el modelo de integración y disparo son caracterizados solo por sus tiempos de disparo $t^{(f)}$, figura 3.7.

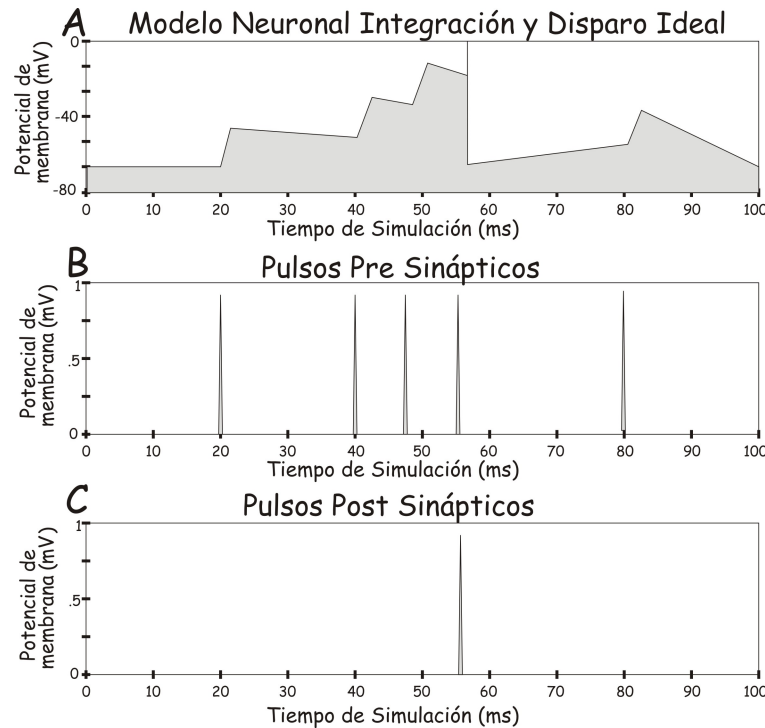


Figura 3.7: Se describe el modelo neuronal de integración y disparo ideal. En la gráfica **A** se muestra el potencial de membrana cuyo umbral se fijó en 1 volt. La gráfica **B** se muestra los pulsos que recibe la neurona post sináptica de una neurona pre sináptica. La gráfica **C** describe la señal de salida (acción potencial) de la neurona post sináptica bajo la conducta del potencial de membrana ver gráfica **A**.

3.2.2.1. Integración y disparo perfecto

Modelo representado por un circuito capacitivo, donde el voltaje de entrada se va acumulando en el capacitor y es disparado cuando se cierra el circuito, como se

representa en la Figura 3.8.

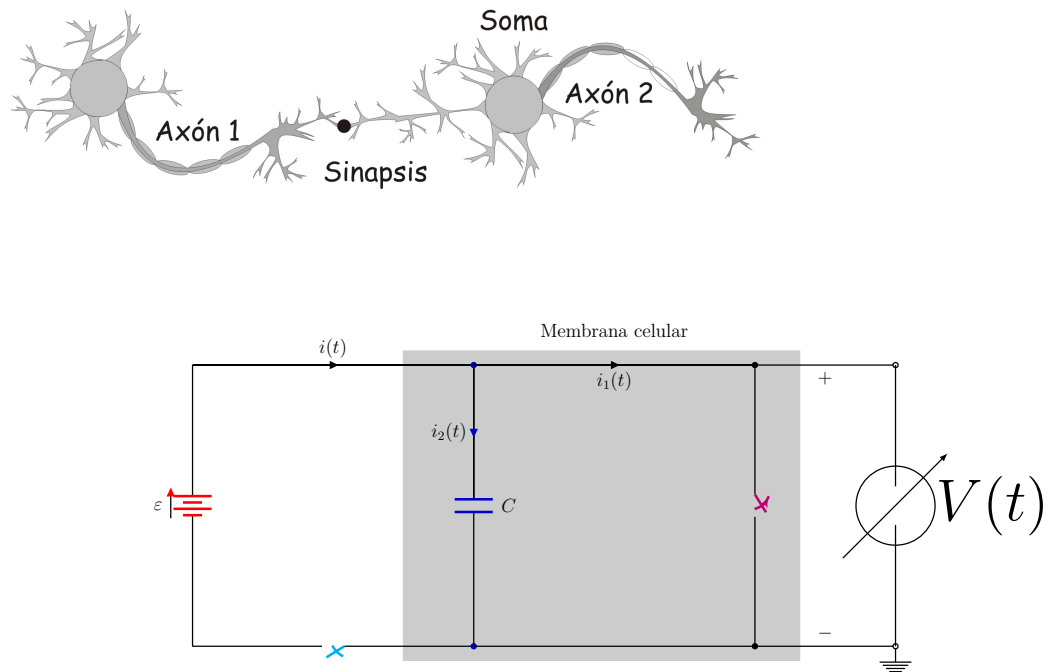


Figura 3.8: Representación del modelo de integración y disparo perfecto.

La ecuación diferencial que describe este sistema está dada por:

$$V(t) = V_r + \frac{1}{C} \int_{t_0}^t i(t) dt \quad (3.6)$$

Donde:

- $V(t)$ es el voltaje acumulado en el capacitor.
- V_r es el valor del voltaje en reposo del capacitor.
- C es el valor del capacitor en μF .
- $i(t)$ es el valor de la corriente de la fuente.

3.2.2.2. Integración y disparo ideal

El modelo LIF es representado por un circuito RC en paralelo, donde la energía acumulada en el capacitor se disipa no solo por el switch, sino también por la resistencia, ver Figura 3.9.

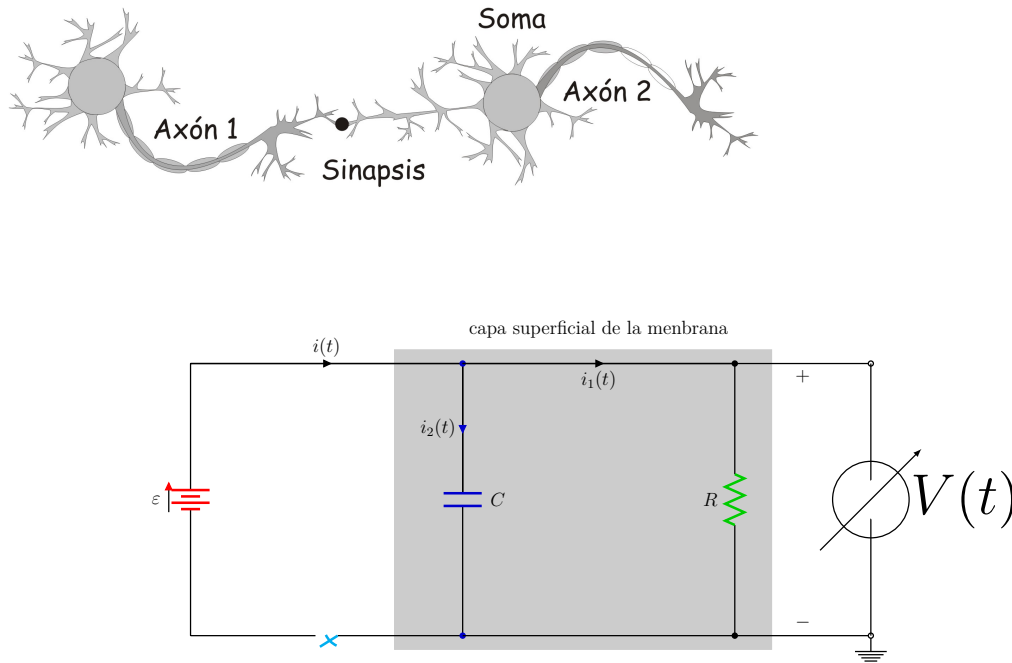


Figura 3.9: Representación del modelo de integración y disparo Ideal (LIF).

La ecuación diferencial que describe este sistema está dada por:

$$V(t) = V_r e^{-\frac{1}{RC}(t-t_0)} + \frac{1}{C} \int_{t_0}^t i(\tau) e^{-\frac{1}{RC}(t-\tau)} d\tau \quad (3.7)$$

Donde:

$V(t)$ es el voltaje del capacitor.

V_r es el valor del voltaje en reposo del capacitor.

C es el valor del capacitor en μF .

$i(\tau)$ es el valor de la corriente de la fuente.
 R es el valor de la resistencia en $m\Omega$.

3.3. Código neuronal

Una de las cuestiones importantes de la neurofisiología es el cómo las neuronas codifican la información. Aún no se ha encontrado una respuesta clara a esta pregunta y toda la información que se encuentra disponible es solo de resultados experimentales. Al principio se pensaba que la información se codificaba en tasas de disparo neuronal. Esto fue probado experimentalmente en 1926, donde se descubrió que la tasa de disparo de una pequeña terminal nerviosa muscular va directamente relacionada a la fuerza que aplica el músculo [22]. Este fue el principal modelo de codificación neuronal por muchos años.

Thrpe et al. [69] probó que los seres humanos pueden reconocer un rostro o imagen compleja en 100ms, lo cual hace imposible la codificación por tasa. Posteriormente, Bialek et al. [3], consiguió observar la codificación neuronal de la vista de una mosca y encontró que está constituida por señales dependientes del tiempo.

Las redes neuronales de tercera generación pueden codificar en forma digital [2],[59] o analógica [49]. La forma de codificación puede ser dividida dentro de tres categorías:

1. Codificación por tasa.
2. Codificación temporal.
3. Codificación por población.

3.3.1. Codificación por tasa

En la codificación por tasa, la información se codifica utilizando el promedio de pulsos en el que una neurona dispara dentro de una ventana de tiempo [22]:

$$v = \frac{\eta_{sp}(T)}{T} \quad (3.8)$$

Donde T es la ventana de tiempo, $\eta_{sp}(T)$ son el número de pulsos emitidos en la ventana de tiempo. Existen tres procedimientos para realizar el promedio [22]: Tasa por conteo de pulsos (promediar sobre el tiempo); Tasa por densidad de pulsos (promediar sobre varios trenes de pulso), y tasa por actividad de poblaciones (promediar sobre diversas neuronas).

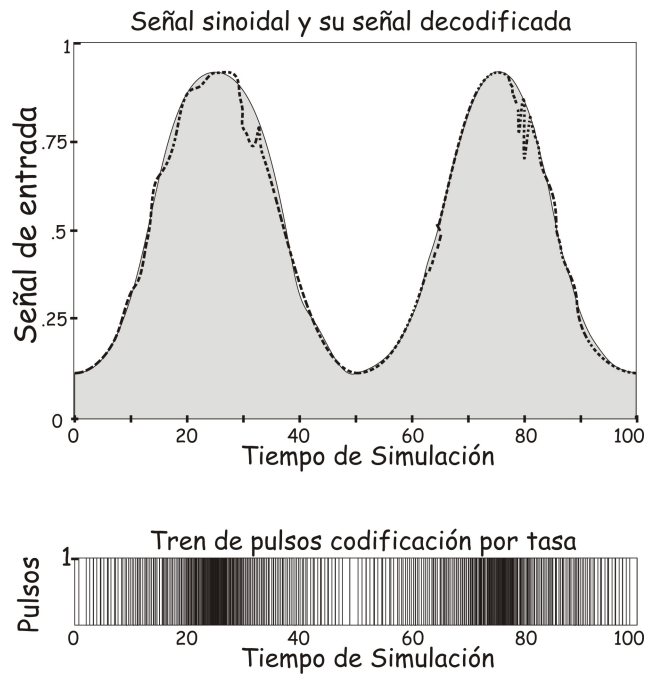


Figura 3.10: Una señal senoidal se codifica por medio del método de codificación por tasa, en la parte de abajo de la figura se muestra la codificación en un tren de pulso donde se percibe que en las crestas de la señal hay una mayor densidad de pulsos, posteriormente se decodifica dicho tren de pulsos para obtener una señal (línea punteada) muy parecida a la señal sinoidal original [60].

3.3.2. Codificación temporal

En la codificación temporal la información es codificada por pulsos temporales [4]. Hopfield [31] propuso un método de codificar señales analógicas dentro de pulsos temporales con respecto a un diseño oscilatorio, este método ha sido probado experimentalmente en un sistema robotico llamado anguila eléctrica. Además Mass [46] propuso un método para codificar señales analógicas en forma de pulsos temporales. Un diferente método ha sido sugerido por Wen y Sendhoff [34], donde la neuronas de entrada codifican directamente la información en pulsos temporales y por ende las neuronas de las capas posteriores usan esa información.

3.3.3. Codificación por población

En la codificación por población, un número de neuronas de entrada (población) codifican las entradas analógicas y producen diferentes tiempos de disparo. Bohte et al. [5] propuso una forma de representar entradas analógicas en trenes de pulsos implementando una codificación por población. Campos receptivos con múltiples funciones Gaussianas (GRF) donde se usan las entradas de las neuronas y se codifican para obtener múltiples retardos, como se muestra en la Figura 3.11.

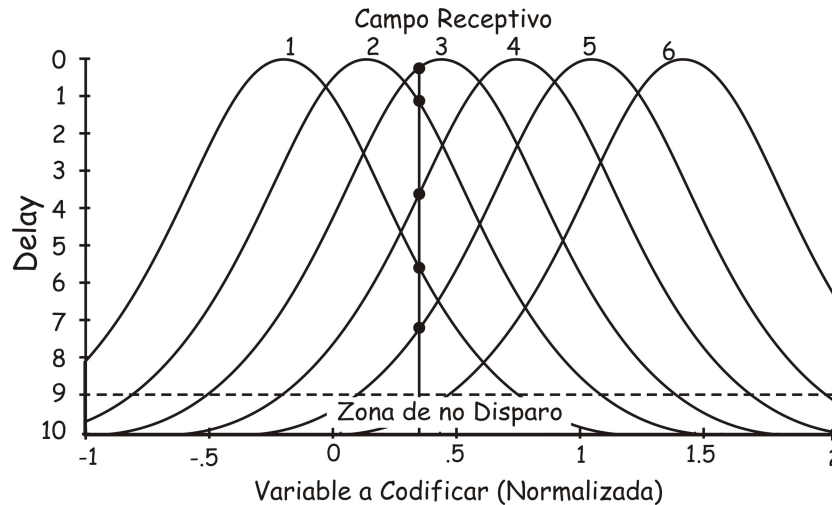


Figura 3.11: Codificación con campos receptivos Gaussianos. El eje horizontal representa la entrada de la variable real a codificar, el eje vertical representan los tiempos para armar el tren de pulsos de la variable codificada.

Para realizar este tipo de codificación, primeramente el rango de la señal de entrada debe ser calculado. Después los máximos y mínimos valores de entrada (I_{max} , I_{min}), tienen que ser definidos. Posteriormente, el número de variables representadas por las neuronas GRF debe ser precisada a través de la variable m . Finalmente, el centro de

cada neurona GRF es calculado por C_i , mientras la amplitud de cada neurona GRF es obtenido por σ_i [49]:

$$C_i = I_{min} + \left(\frac{2i - 3}{2} \right) \left(\frac{I_{max} - I_{min}}{m - 2} \right) \quad (3.9)$$

$$\sigma_i = \frac{1}{\gamma} \frac{I_{max} - I_{min}}{m - 2} \quad (3.10)$$

Donde σ es una constante, usualmente es 1.5. Se propone un umbral del cual cualquier valor por encima de éste, la neurona no dispara. En el ejemplo de la Figura 3.11, el valor analógico a codificar en disparos temporales en 6 neuronas es de 0.3, lo cual la codificación resulta de la siguiente manera: 0.250ms para la neurona 1, 1.287ms para la neurona 2, 3.783ms para la neurona 3, 5.564ms para la neurona 4, 7.741ms para la neurona 5, y la neurona número 6 no emite disparo pues se encuentra arriba del umbral.

3.4. Métodos de aprendizaje

Los pesos w_{ij}^k entre la neurona pre sináptica I y la neurona post sináptica J ver Figura 3.4 no son valores fijos. El cambio de estos valores está directamente relacionado con la salida neuronal y el manejo de información por trenes de pulsos. Si la sinapsis se fortalece se le conoce como LTP (Long Term Potentiation) y si por el contrario la sinapsis se debilita recibe el nombre de LTD (Long Term depression). El proceso en el que los pesos se actualizan se le conoce como entrenamiento. El cual puede ser dividido dentro de dos categorías: Aprendizaje supervisado y no supervisado.

3.4.1. Aprendizaje supervisado

Como se ha descrito en las secciones previas, el comportamiento de las redes neuronales de tercera generación se asemejan más a la dinámica de las redes neuronales biológicas reales, al mismo tiempo, son más poderosas que las neuronas artificiales de generaciones anteriores. Sin embargo, su poder computacional no ha sido explotado totalmente, esto se debe al número tan limitado de algoritmos de aprendizaje no supervisado que existen. Existen dos principales razones para que esto ocurra, la primera es que las redes neuronales de tercera generación a diferencia de sus antecesoras son funciones discontinuas, lo cual dificulta la creación de algoritmos de aprendizaje, la segunda razón es el poco tiempo que llevan de existencia.

3.4.1.1. Método SpikeProp

Bohte et al. [5] propuso un algoritmo de aprendizaje supervisado basado en el algoritmo BackPropagation de redes neuronales artificiales de segunda generación. Este método se denomina SpikeProp y fue diseñado para una arquitectura de redes neuronales con múltiples conexiones las cuales tiene múltiples retardos, como la mostrada en la Figura 3.4.

La arquitectura que necesarita este entrenamiento es de tipo feed-forward, lo que significa que no existe retroalimentación entre ninguna neurona de la red, aunque esto no limita a que la red neuronal tenga una o más capas ocultas. Es decir la arquitectura es similar a la Figura 3.4 H es la capa de entrada, I es la capa oculta y J es la capa de salida.

Bohte et al. [5] implementa el modelo del pulso de respuesta (SRM) para describir el potencial de membrana de cada pulso emitido por las neuronas, el kernel asociado a cada SRM es sin el fragmento de refracción de la ecuación, esto quiere decir que la ecuación queda de la siguiente forma:

$$u_j(t) = \sum_{i \in \Gamma_j} \sum_{k=1}^m w_{ij}^k \varepsilon(t - t_i - d^k) \quad (3.11)$$

El propósito de este algoritmo es que la red neuronal sea entrenada de acuerdo al error que se genera con la diferencia entre los tiempos de disparo específicos correspondientes a salidas deseadas y las entradas a la red neuronal, esto se logra con la función de error:

$$E = \frac{1}{2} \sum_{j \in J} (t_j^a - t_j^d)^2 \quad (3.12)$$

Donde t_j^a representa el tiempo de disparo de la j -ésima neurona de salida actual, t_j^d es el tiempo de disparo de la j -ésima neurona de salida deseada.

La adaptación de los pesos neuronales relacionados a la neurona de la capa de salida es:

$$\Delta w_{ij}^k = -\eta y_i^k(t_j^a) \delta_j \quad (3.13)$$

Donde η , es la tasa de aprendizaje, y δ_j se formula como:

$$\delta_j = \frac{(t_j^a - t_j^d)}{\sum_{i \in \Gamma_j} \sum_l w_{ij}^l \left(\frac{\partial y_i^l(t_j^a)}{\partial t_i^a} \right)} \quad (3.14)$$

La adaptación de los pesos neuronales relacionados a la neurona de las capas ocultas y capas de entrada es:

$$\Delta w_{hi}^k = -\eta y_h^k(t_i^a) \delta_i = -\eta \frac{y_h^k(t_i^a) \sum_j \left[\delta_j \sum_k w_{ij}^k \left(\frac{\partial y_i^k(t_j^a)}{\partial t_i^a} \right) \right]}{\sum_{n \in \Gamma_i} \sum_l w_{ni}^l \left(\frac{\partial y_n^l(t_i^a)}{\partial t_i^a} \right)} \quad (3.15)$$

donde δ_i es:

$$\delta_i = \frac{\sum_{j \in \Gamma_i} \left[\delta_j \sum_k w_{ij}^k \left(\frac{\partial y_i^k(t_j^a)}{\partial t_i^a} \right) \right]}{\sum_{h \in \Gamma_i} \sum_l w_{hi}^l \left(\frac{\partial y_h^l(t_i^a)}{\partial t_i^a} \right)} \quad (3.16)$$

La prueba matemática de este algoritmo se puede encontrar en la referencia [14], Finalmente, el método SpikeProp puede ser desarrollado en la siguiente tabla 3.1.

Tabla 3.1:

Algoritmo SpikeProp

Se calcula δ_j para todas las salidas, de acuerdo a (3.14)

Por cada capa subsecuente $I = J - 1, J - 2, \dots$

se calcula δ_i para todas las neuronas de la I -ésima capa de acuerdo a (3.16)

Para cada neurona j -ésima de la capa de salida J ,

adapta w_{ij}^k utilizando $\Delta w_{ij}^k = -\eta y_i^k(t_j) \delta_j$ (3.13),

Para cada capa subsecuente $I = J - 1, J - 2, \dots$

calcular w_{hi}^k utilizando $\Delta w_{ij}^k = -\eta y_i^k(t_j) \delta_j$ (3.15).

Una de las desventajas del algoritmo SpikeProp es que toma demasiado número de iteraciones para converger, varias técnicas han sido propuestas para mejorar este proceso; alguno de esos son los algoritmos Rprop y QuickProp [21].

3.4.1.2. Método asociado a la entrada-salida

Como lo describe A. Mohemmed et al. [52], el aprendizaje asociado a la entrada-salida es similar a otros algoritmos de aprendizaje supervisado. Los pesos sinápticos de la red son ajustados iterativamente con la finalidad de obtener un mapeo entrada/salida deseada de la SNN.

Las neuronas del tipo LIF son modelos comúnmente usados en la simulación de redes neuronales de pulso, ya que esta basado en un circuito eléctrico RC donde R y C son asumidas como constantes, y donde la dinámica de una neurona i esta descrita por:

$$\tau_m \frac{du_i}{dt} = -u_i(t) + RI_i^{syn}(t) \quad (3.17)$$

Donde $\tau_m = RC$, y es la constante de tiempo de la membrana de la neurona, u_i es el potencial de membrana que cruzó al umbral v desde abajo generando la neurona un pico para después regresar al voltaje de relajación u_r .

El aprendizaje es similar a otros aprendizajes supervisados, los pesos sinápticos son modificados iterativamente en el orden impuesto, deseando una entrada/salida mapeando la SNN. Se inicia con la regla general de Widrow-Hoff, modificando el peso sináptico i :

$$\Delta\omega_i^{WH} = \eta x_i(y_d - y_s) \quad (3.18)$$

Donde $\eta \in \mathbb{R}$ y es la tasa de aprendizaje de la red, x_i es la entrada transferida mediante la sinápsis i , y_d es la salida deseada y y_s es la salida actual de la neurona respectiva. De acuerdo a (3.18), $\Delta\omega_i^{WH}$ no está en función del tiempo, integraremos para obtener un escalar en función del tiempo como:

$$\Delta\omega_i = \eta \int x_i(t)(y_d(t) - y_s(t))dt \quad (3.19)$$

Los pesos presinápticos son modificados en un proceso iterativo, todas las muestras obtenidas son de forma secuencial y acumulativa.

Capítulo 4

Aplicación de las SNN al procesamiento digital de señales

El objetivo de este capítulo es presentar las aplicaciones de las SNN en el procesamiento digital de señales, en particular en la aproximación de señales médicas, señales heurísticas y señales determinísticas. La organización del presente capítulo es la siguiente: en la sección 4.1 se da una breve introducción sobre la importancia del procesamiento de señales médicas en el contexto de PDS, también se presenta el esquema de SNN para dicho propósito.

4.1. Introducción

En la actualidad, una de las prioridades del ser humano, es conocer el comportamiento y funcionamiento del cuerpo humano, siendo el cerebro y el corazón los órganos con mayor interés para su estudio. Las señales eléctricas que se obtienen de los órganos mencionados son conocidas como Electrocardiograma para el corazón (ECG) y Electroencefalograma para el cerebro (EEG).

El electrocardiograma de superficie, registra la diferencia de potencial de voltaje que se genera en el corazón a través de electrodos colocados en puntos preestablecidos en la superficie corporal. El potencial eléctrico registrado en las derivaciones electrocardiográficas representa la sumatoria de las fuerzas eléctricas instantáneas (vectores) que ocurren en forma secuencial durante los procesos de despolarización y repolarización.

En esta sección se presenta la introducción sobre las aplicaciones de las SNN, donde se describen los resultados que están relacionados directamente con la aplicaciones proyectadas en este capítulo.

En este capítulo, en la Figura 4.1 se muestra un esquema de SNN para la aproximación de señales con diferentes características como son: aleatorias y no estacionarias [50].

En esta tesis, a diferencia de los resultados presentados en la literatura, aquí se aplica a señales médicas, su principal característica es que son señales no estacionarias, para este estudio se forman un ECG y EEG; también se utilizan una señal heurísticas y una determinísticas para el análisis y aproximación mediante las redes neuronales de tercera generación.

En la Figura 4.1 se muestra el diagrama a bloques, donde se observa que la señal a ser aproximada es y , la señal del error e es utilizada para el entrenamiento de la SNN. Para el caso de estudio la entrada y corresponde a la señal EEC o la señal ECG, la señal determinística y la señal heurística, respectivamente. La salida de la SNN \hat{y} es la aproximación de las señales deseadas, e es el error que existe entre la señal deseada y la aproximación de la red, además, el error es la señal usada para el entrenamiento de la SNN, v es una señal de excitación persistente, para estos experimentos se considera contante, siendo esta un pulso en el instante 200, dentro de la ventana de tiempo.

La arquitectura propuesta para este trabajo, contiene una capa de entrada con una

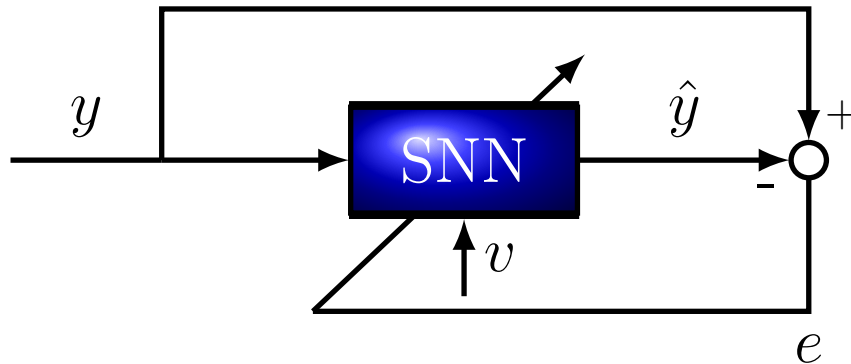


Figura 4.1: Esquema propuesto para la aproximación de señales.

neurona SN_1^1 , una capa oculta SN_n^2 donde n es el número de neuronas de la capa y una capa de salida con una neurona SN_1^3 , como se muestra en la Figura 4.2. El valor de n depende de la complejidad de la señal que se requiere aproximar, en este caso se propone diferente, para las señales determinística y heurística $n = 3$, y para las señales EEG y ECG se realizan dos experimentos uno con $n = 5$ y otro con $n = 10$.

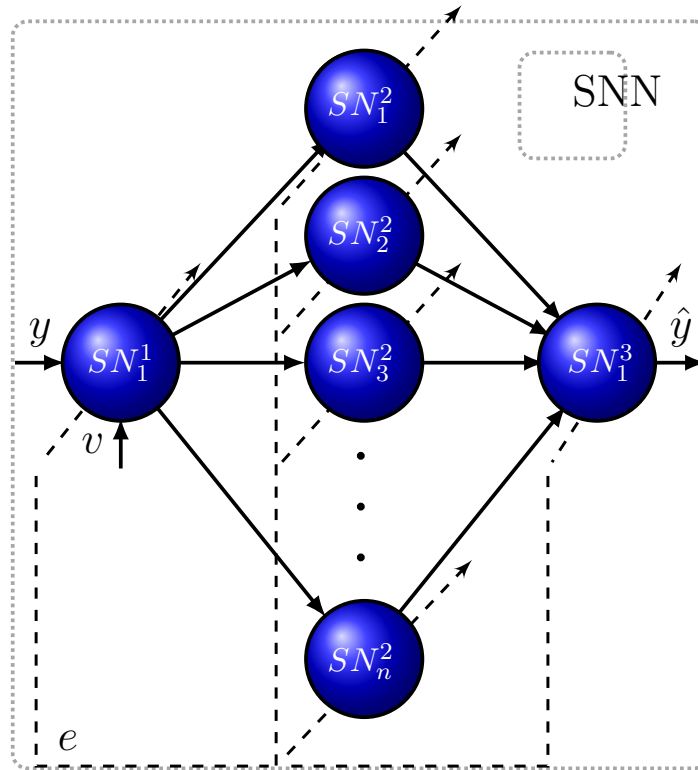


Figura 4.2: Arquitectura propuesta de la SNN para la aproximación de señales.

4.2. Señales determinísticas

Para el experimento de la señal determinística se propone una señal con cambios en sus niveles de voltaje en y , de acuerdo a la Figura 4.1, durante un número de muestras determinado, para lo cual se usó el modelo matemático siguiente:

$$y = .8\sin(Ni * .05) * .2 + .6) \text{ para } Ni < 250$$

$$y = 780mV \text{ para } 250 \leq Ni < 500$$

$$y = 422mV \text{ para } 500 \leq Ni < 750$$

$$y = 0.3\sin(Ni * .07) + 0.1\sin(Ni * .02) + 0.65\sin(Ni * .09))/3.5 + .69) * 1000) \text{ para } Ni \geq 750$$

Donde N_i es el número de muestras en un tiempo n .

Como se puede observar en la Figura 4.3 la aproximación de la señal que realiza la SNN es parecida a la señal determinística que se usa como la entrada y .

En la Figura 4.4 se muestra el error que se genera entre la señal de entrada y y la señal de salida \hat{y} , donde se aprecia que el error e se aproxima a cero en las 170 muestras y manteniéndose en ese nivel hasta llegar a la muestra 910, en donde la señal determinística realiza un cambio en su amplitud, haciendo que el error crezca.

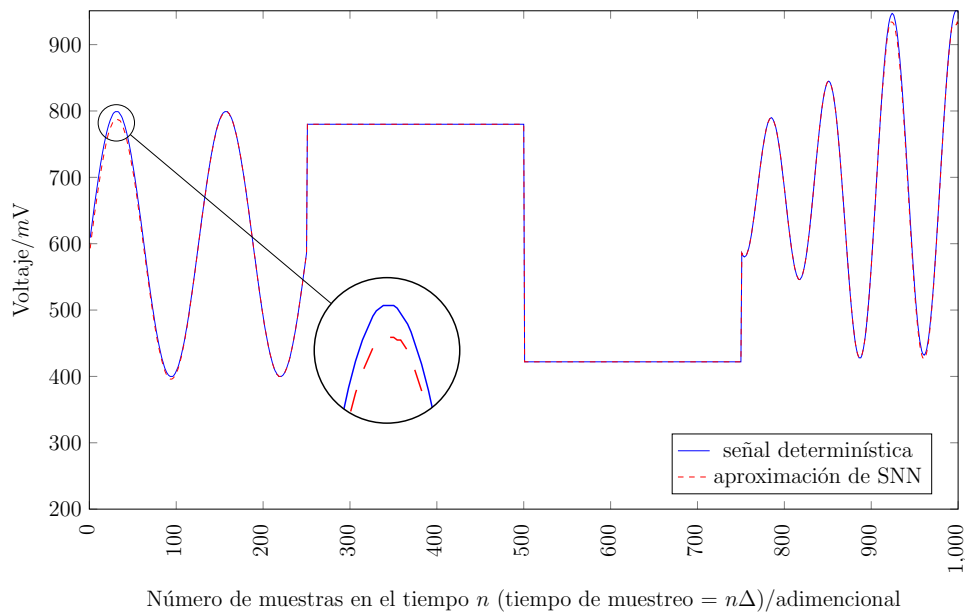


Figura 4.3: Aproximación de la señal determinística, mediante la SNN con una arquitectura de 3 SN en la capa oculta.

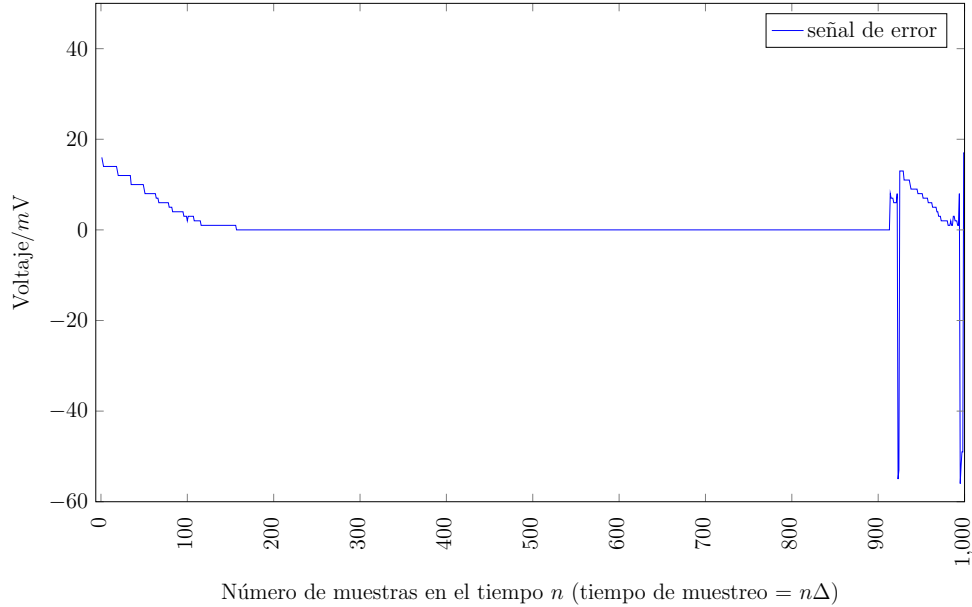


Figura 4.4: Error de aproximación de la señal determinística, mediante la SNN con una arquitectura de 3 SN en la capa oculta.

Como se puede observar para el caso de señales determinísticas la aproximación obtenida mediante la arquitectura propuesta, es muy buena para cuando se tienen cambios drásticos en la señal, sin embargo, cuando se tienen cambios en la amplitud de la señal o un cambio de señal, se incrementa el error, es debido al aprendizaje previo de la SNN. En la siguiente sección se muestran resultados para una señal que desde el inicio tiene cambios en la amplitud.

4.3. Señales heurísticas

Para el experimento de la señal heurística se considera tres SN en la capa oculta como se describe en la Figura 4.2, esta cantidad de neuronas son las necesarias para obtener una señal de error pequeña, como se muestra en 4.6. La señal heurística propuesta está dada por la siguiente ecuación:

$$y = ((0.4\sin(Ni*.07)+0.15\sin(Ni*.02)+0.7\sin(Ni*.09))/3.5)*abs(\cos(randi(Ni, 1))) * 900)$$

Donde N_i es el número de muestras en un tiempo n

En la Figura 4.5 se muestra la respuesta de la SNN con la aproximación en la línea punteada, observando que desde las primeras muestras hace la aproximación con un error pequeño.

En la Figura 4.6 se muestra la señal de error e que se genera entre la señal heurística y la aproximación de la SNN, observando como el error se aproxima a cero antes de las 100 muestras, cuando la señal heurística tiene cambios en su amplitud el error permanece acotado cerca del cero, y a medida que se incrementan las muestras se mejora la aproximación, esto debido al aprendizaje de la red. Sin embargo es importante mencionar que, si bien es cierto que la arquitectura de la red es la misma para ambos casos (señal determinística y señal heurística) las condiciones iniciales pueden ser diferentes.

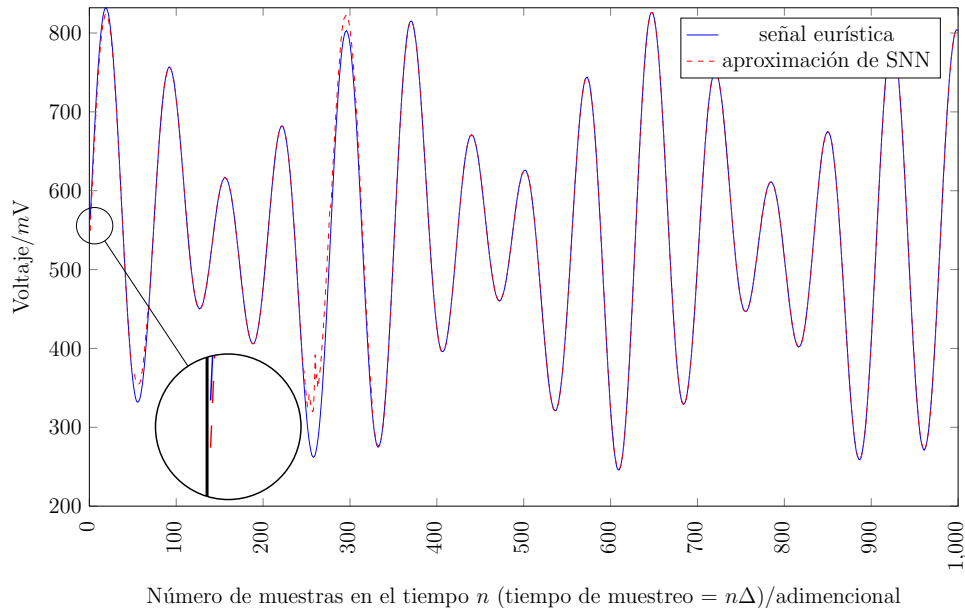


Figura 4.5: Aproximación de la señal heurística, mediante la SNN con una arquitectura de 3 SN en la capa oculta.

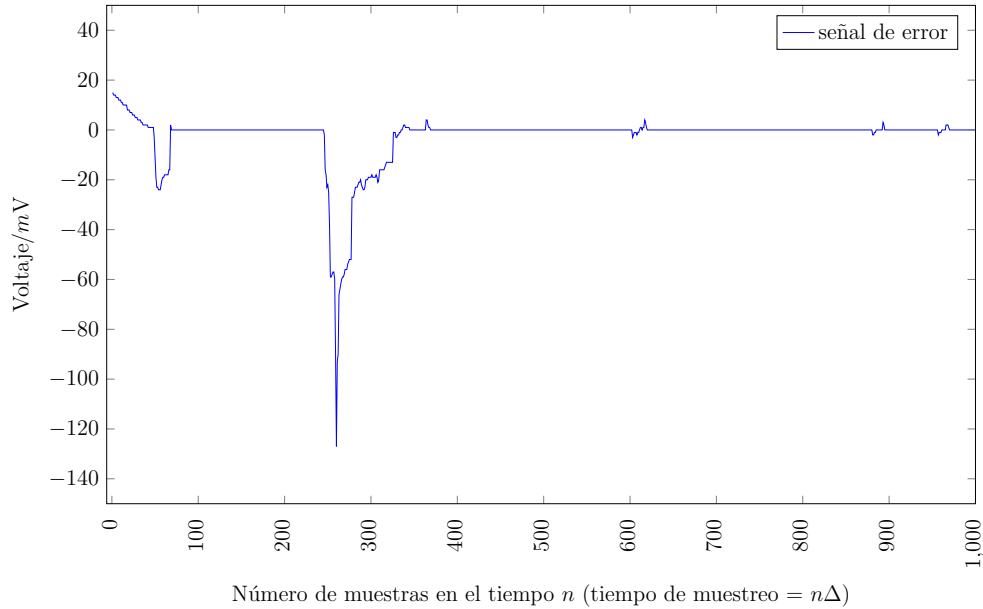


Figura 4.6: Error de aproximación de la señal heurística, mediante la SNN con una arquitectura de 3 SN en la capa oculta.

Con este experimento se muestra la capacidad que tiene las SNN para la adaptación para lograr la aproximación de las señales propuestas en la sección 4.3 y 4.2, en estos resultados sólo requirió de tres SN en la capa oculta. Sin embargo, en la naturaleza existen señales más complejas que requieren de una cantidad mayor de neuronas para su procesamiento, como son la señales no estacionarias.

4.4. Seguimiento de señales médicas mediante SNN

En el experimento para el seguimiento de la señal electroencefalograma (EEG), y el electrocardiograma se toman los datos obtenidos por el dispositivo Emotiv EPOC y por un electrodo superficial respectivamente, se propone dos arquitecturas una que contiene 5 SN en la capa oculta y otra que contiene 10 SN en la capa oculta de acuerdo a la Figura 4.2, obteniendo una aproximación como se muestra en la subsección 4.4.1 y 4.4.4, respectivamente

4.4.1. Aproximación de la señal EEG con una SNN con 5 SN en la capa oculta

En la Figura 4.7 se observa el resultado de la aproximación hecha por la SNN con 5 SN en la capa oculta sobre una señal de electroencefalograma en el que se observa cómo desde el inicio de las muestras la señal del EEG se aproxima de forma eficaz. En la Figura 4.8 se muestra la señal de error que genera la diferencia de la señal EEG original y la señal que produce la SNN, observando que el error se aproxima a cero en las 200 muestras, manteniendo niveles bajos de voltaje para las muestras siguientes.

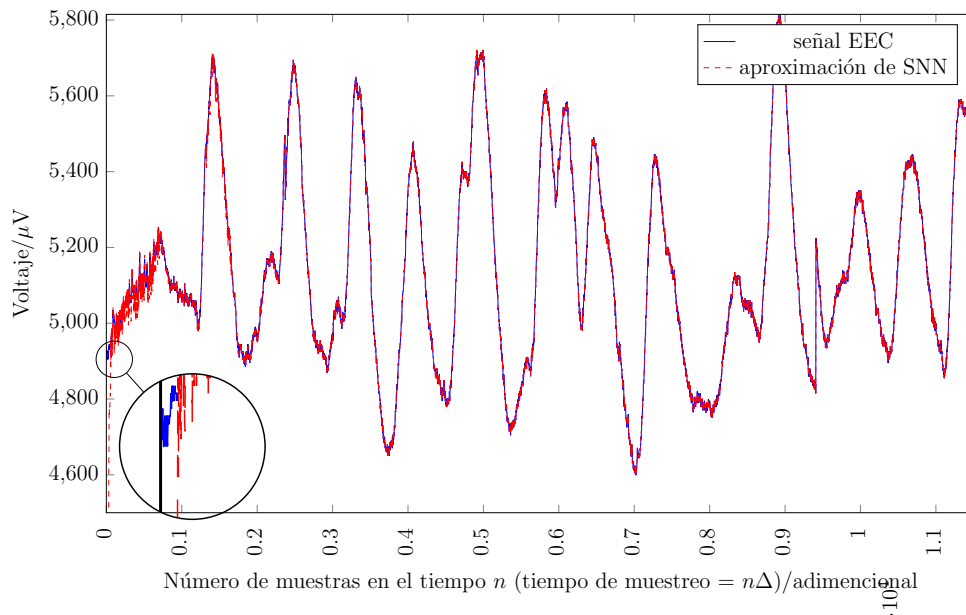


Figura 4.7: Aproximación de la señal Electroencefalograma, mediante la red neuronal de pulso con una arquitectura de 5 SN en la capa oculta.

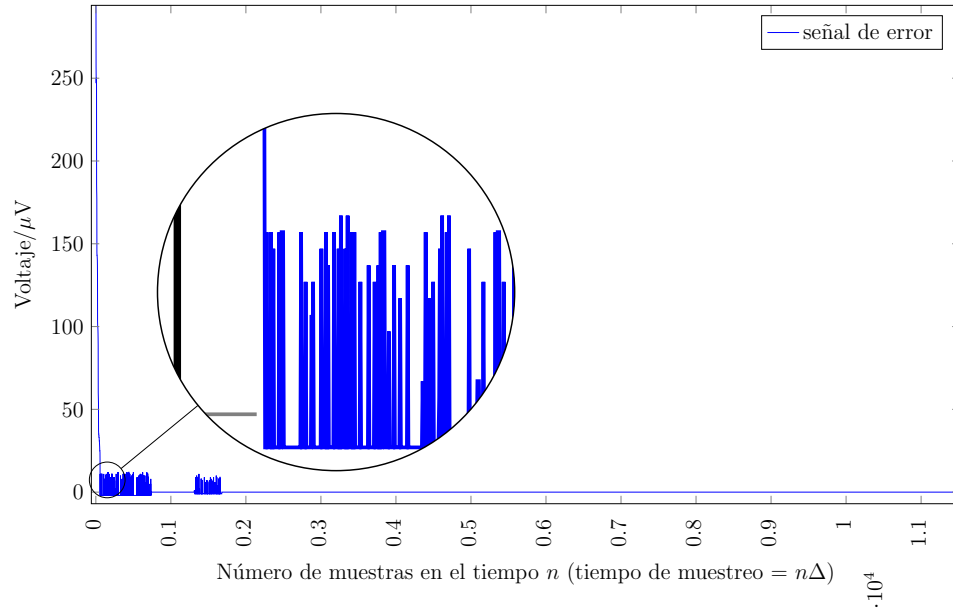


Figura 4.8: Señal de error existente entre la aproximación y la señal EEG.

4.4.2. Aproximación de la señal EEG con una SNN con 10 SN en la capa oculta

Para este caso se propone una arquitectura de diez SN en la capa oculta, con la finalidad de mejorar el comportamiento de la SNN y obtener una mejor aproximación. En la Figura 4.9, la señal que aproxima la SNN (línea punteada) se acerca a la señal EEG deseada (línea continua) en las primeras muestras, teniendo un error que se acerca a cero de forma rápida como se muestran en la Figura 4.10.

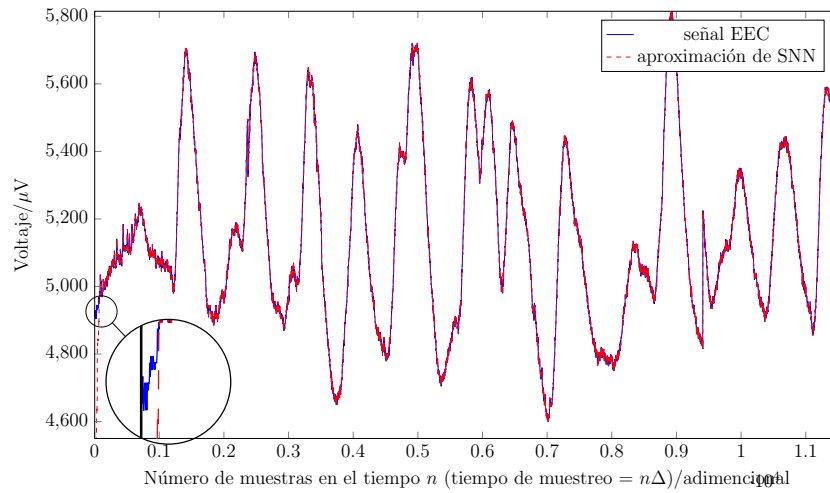


Figura 4.9: Aproximación de la señal Electroencefalograma, mediante la red neuronal de pulso con una arquitectura de 10 SN en la capa oculta.

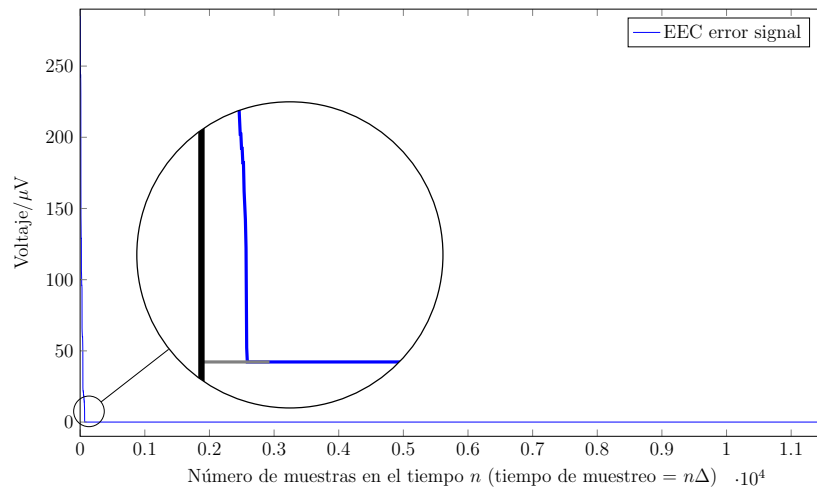


Figura 4.10: Señal de error existente entre la aproximación y la señal EEG..

4.4.3. Aproximación de la señal ECG con una SNN con 5 SN en la capa oculta

Para el otro caso, se tomó una señal proveniente de un electrocardiograma, siendo ésta una señal más compleja que un electroencefalograma, esta prueba de complejidad se realiza mediante un análisis de entropía a la señal, sin embargo, la SNN también logra aproximarla como se muestra en la Figura 4.11 y 4.13, con una arquitectura de 5 y 10 SN en la capa oculta.

En la Figura 4.14 se observa como el error se aproxima al cero en las 2400 muestras, a diferencia que el error de la Figura 4.10 que en un menor número de muestras se aproxima a cero, esta señal Figura 4.14 tarda más número de muestras en ser aproximada, y que esto es debido a la complejidad de la señal ECG, cabe hacer mención que se realizaron diferentes experimentos modificando el número de neuronas de la capa oculta.

Por ejemplo se propuso que tenga tres, cinco, diez, y veinte, estos resultados se incluyen en un artículo sometido para su revisión en un congreso internacional. En el que se expone que el número de neuronas en la capa oculta depende de la complejidad de la señal a ser aproximada, pero además de que si se aumenta el número de SN en la capa oculta el resultado no necesariamente tenderá a mejorar, como fue el caso de 20 SN en la capa oculta.

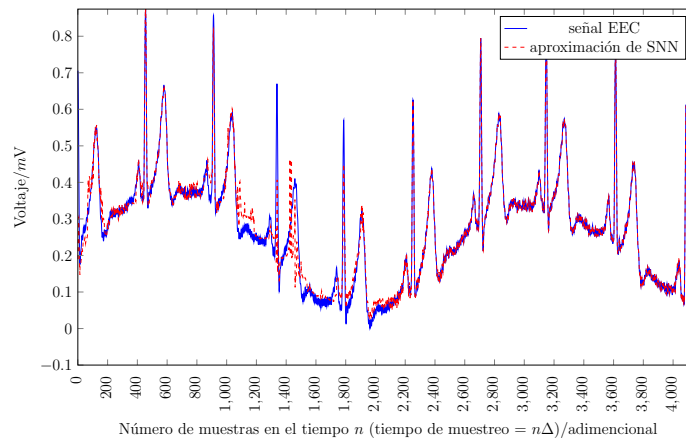


Figura 4.11: Aproximación de la señal ECG, con SNN con una arquitectura de 5 SN en la capa oculta.

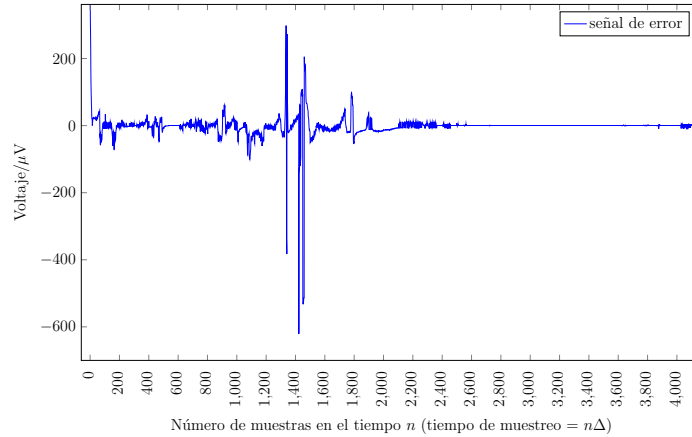


Figura 4.12: Señal de error existente entre la aproximación y la señal ECG.

4.4.4. Aproximación de la señal ECG con una SNN con 10 SN en la capa oculta

Estos experimentos de aproximación de señales complejas ha demostrado que las SNN son unas excelentes aproximadoras, dando pauta a decir que estos experimentos sirven para hacer identificación de sistemas con el solo conocimiento de la señal de salida. Este trabajo demuestra que las SNN pueden manipular señales que tengan dinámicas ricas, como los que el carro péndulo tiene en su funcionamiento, además como no existen datos de una implementación similar era obligatorio mostrar esta cualidad de las redes neuronales de pulso (SNN). Como ya se ha mencionado en el transcurso de este capítulo, el carro péndulo invertido, es un sistema mecánico no lineal del tipo subactuado que tiene una complejidad alta, las SNN tendrán la capacidad de controlarlo como se mostrará más adelante. Sin embargo, el trabajo de las SNN no se limita a lo presentado en esta tesis, si no que en el ramo de las aproximaciones se podrá trabajar como filtros en el tema de las señales cerebrales, diagnóstico de enfermedades en el corazón o en el cerebro, es decir el tema de las SNN como aproximadoras queda abierto para una infinidad de trabajos.

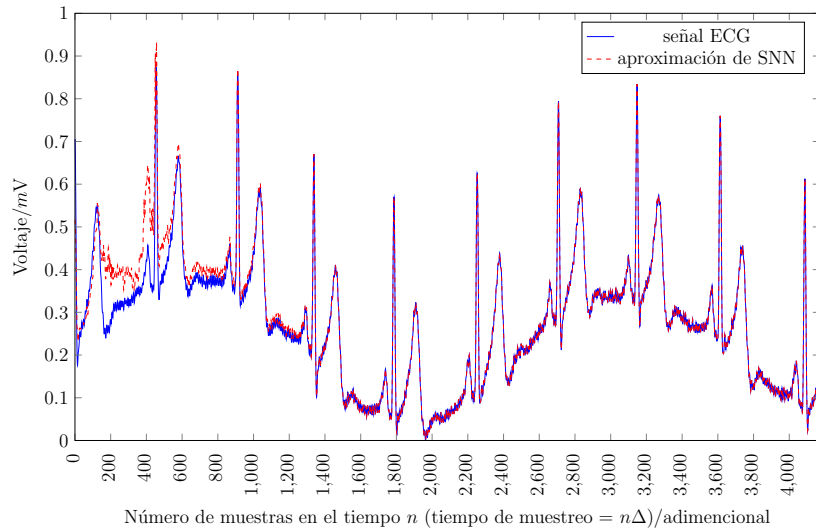


Figura 4.13: Aproximación de la señal Electrocardiograma, mediante la red neuronal de pulso con una arquitectura de 10 SN en la capa oculta.

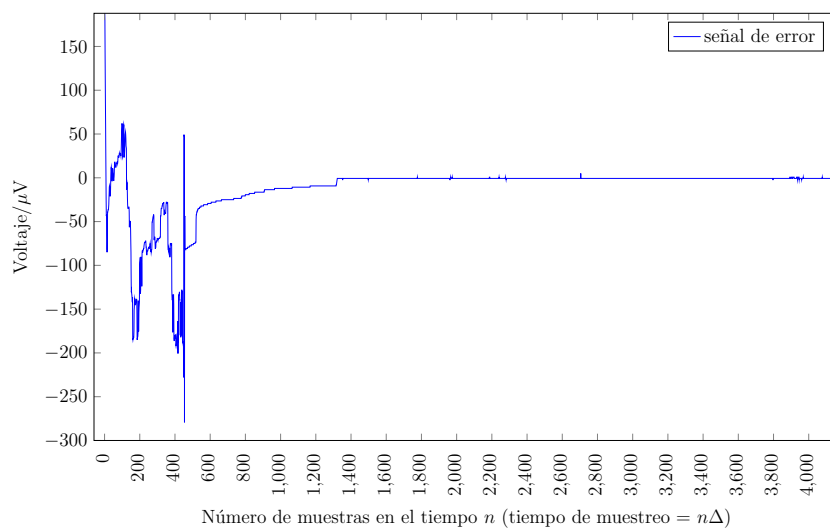


Figura 4.14: Señal de error existente entre la aproximación y la señal ECG.

Capítulo 5

Aplicación de las SNN al control automático

Las redes neuronales se han aplicado como auxiliar en los sistemas de control, es decir, para el cálculo de ganancias de otros controladores. Este trabajo es una consecuencia de la tesis realizada para la obtención del grado en la maestría en ciencias en automatización y control de la UAEH, [39] en la que hace alusión de un control con SNN sobre una planta del tipo lineal. Siendo una de los primeros trabajos en los que se usan las SNN como controladores.

Para este trabajo se opta por incrementar el grado de complejidad en la planta a controlar, siendo un sistema mecánico no lineal del tipo subactuado.

Para la implementación del control con las SNN, se requirieron de diferentes arquitecturas con la finalidad de encontrar la que nos proporcione una mejor respuesta como sistema de control. Uno de los primeros pasos para la aplicación de control; es garantizar que las SNN sean capaces de procesar todas y cada una de las señales obtenidas del sistema.

La lectura de las señales en el sistema carro péndulo invertido, posición angular θ y posición del carro p , tienen la característica de tener una gran riqueza en sus dinámicas, por tal motivo se inicia este trabajo con una aplicación de las SNN en el Capítulo 4. El carro péndulo invertido, como se mencionó en el Apéndice C es la planta no lineal que se utiliza para los experimentos. Es importante señalar que para la realización de este trabajo de simulación es necesario conocer el modelo matemático de la planta a controlar, sin embargo, en la práctica las SNN no requieren el conocimiento del modelo, ésto es una de las características que hacen que las SNN sean consideradas como un sistema de control inteligente.

5.1. La SNN como sistema de control

Las redes neuronales de pulso se han utilizado en diferentes aplicaciones por la gran capacidad que tienen. En este trabajo se propone la utilización de una SNN con una arquitectura de una SN como un sistema de control con la finalidad de mantener la planta en un punto deseado, para este objetivo se propone el diagrama de la Figura 5.1.

En el diagrama mostrado en la Figura 5.1 está propuesto un diagrama general para la utilización de la SN como un sistema de control donde: y es la salida de la planta; r es la señal de referencia a la que queremos llegar; e es el error que existe entre la salida de la planta y la referencia, es decir

$$e = r - y \quad (5.1)$$

El error es tomado para el entrenamiento de la neurona; SN es una neurona; u es la señal de control que se aplica a la planta.

SN es del tipo PIF, LIF o Izhikevich, y el entrenamiento utilizado durante este trabajo es el del tipo SpikeProp.

La neurona, su entrenamiento y la codificación son programadas y simuladas en el software computacional Matlab, en el Apéndice B se muestra el programa desarrollado.

5.2. Control del carro péndulo invertido con una arquitecturas de una SN y diferentes configuraciones en la entrada

En esta sección se muestra el comportamiento en simulación del carro péndulo cuando se coloca una neurona con diferentes entradas de alimentación, como son el colocar el error como entrada e , la derivada del error $\frac{d}{dt}e$, la integral del error $\int e$, la salida de la neurona u , la salida del sistema y y la salida de SN.

5.2.1. La SNN con una arquitectura de una SN y una entrada

Para este caso se toma el diagrama general de la Figura 5.1, donde la planta a considerar es el carro péndulo invertido, la referencia r es tomada como la posición angular deseada respecto a la vertical como se observa en la Figura C.3 y se mide en grados, el estado de la planta x está compuesta por la posición del carro p , la velocidad de la posición del carro \dot{p} , la posición angular θ , y la velocidad angular de

la posición $\dot{\theta}$ de acuerdo a (C.12), sin embargo, sólo están disponibles p y θ , ya que son las que podemos medir de nuestro sistema físico.

Para este caso sólo se considera controlar la planta en la posición angular θ , con el

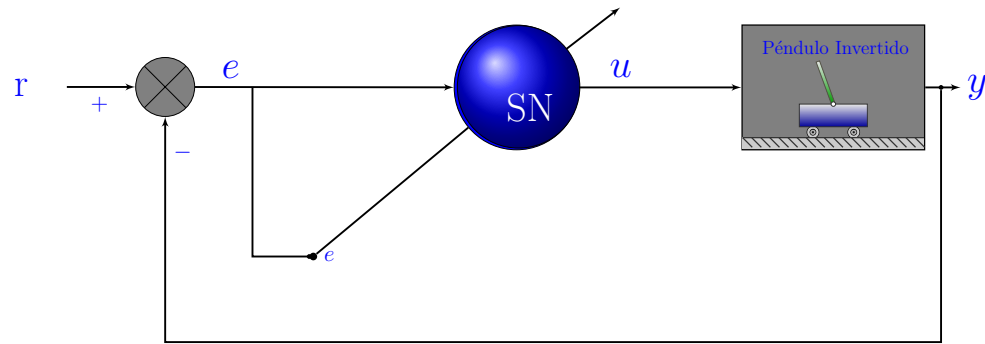


Figura 5.1: Diagrama con una arquitectura de una SN y el error como entrada.

objetivo de mantener el péndulo en posición vertical, dejando libre a p .

El error e se obtiene:

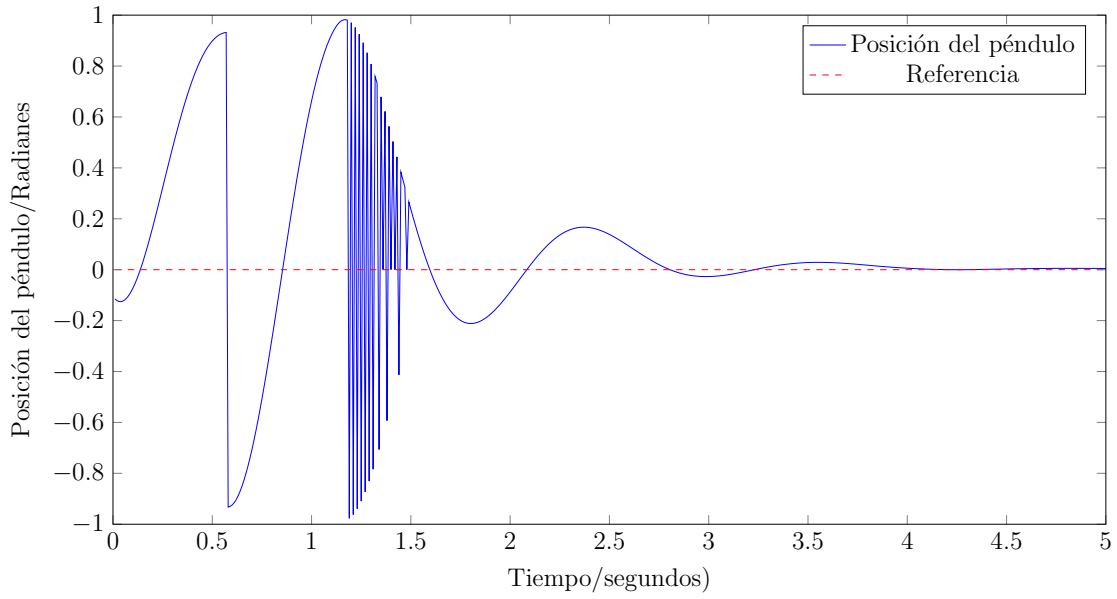
$$e = r - y$$

La neurona de pulso SN, es colocada como el sistema de control, en este caso se coloca el error e como entrada de la neurona y como parte del entrenamiento. La SN se toma como una neurona del tipo Izhikevich y su entrenamiento es tomado con el algoritmo SpikePprop.

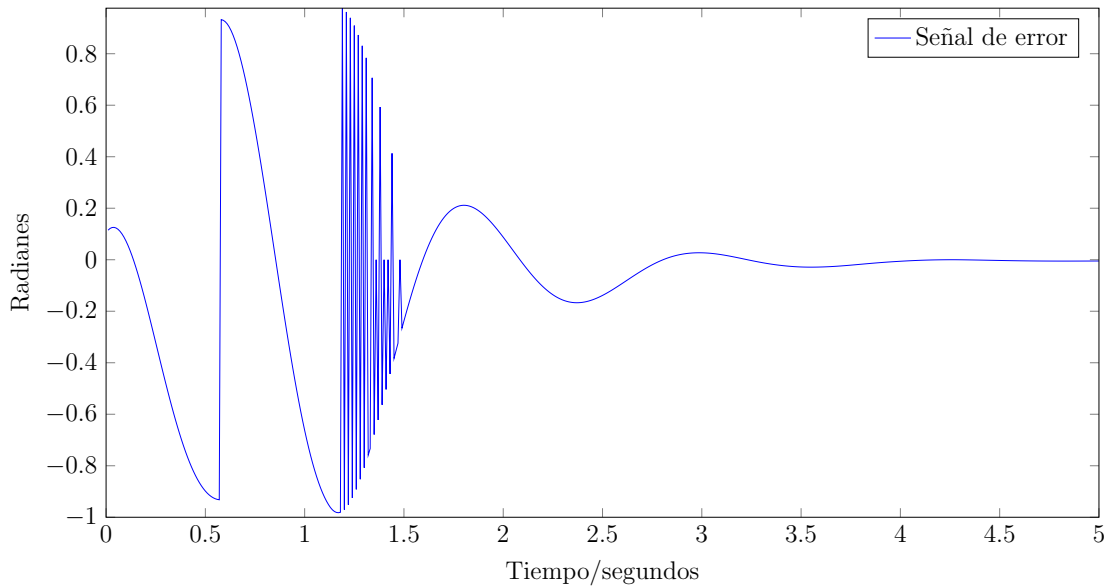
En la Figura 5.2 se muestra la respuesta del sistema carro péndulo invertido, al aplicarle un control u que se obtiene mediante la SN. Para este sistema se consideran las siguientes condiciones iniciales: $x = 0$, $\dot{x} = 0$, $\theta = .1 \text{radianes}$, $\dot{\theta} = 0$.

En la Figura 5.2(a) se muestra cómo la neurona controla y pone al péndulo en la posición vertical en un tiempo de 14 segundos. En la Figura 5.2(b) se muestra el error existente entre la referencia y la posición del péndulo. Y en la Figura 5.2(c) se muestra el control necesario para mantener al péndulo en la posición deseada.

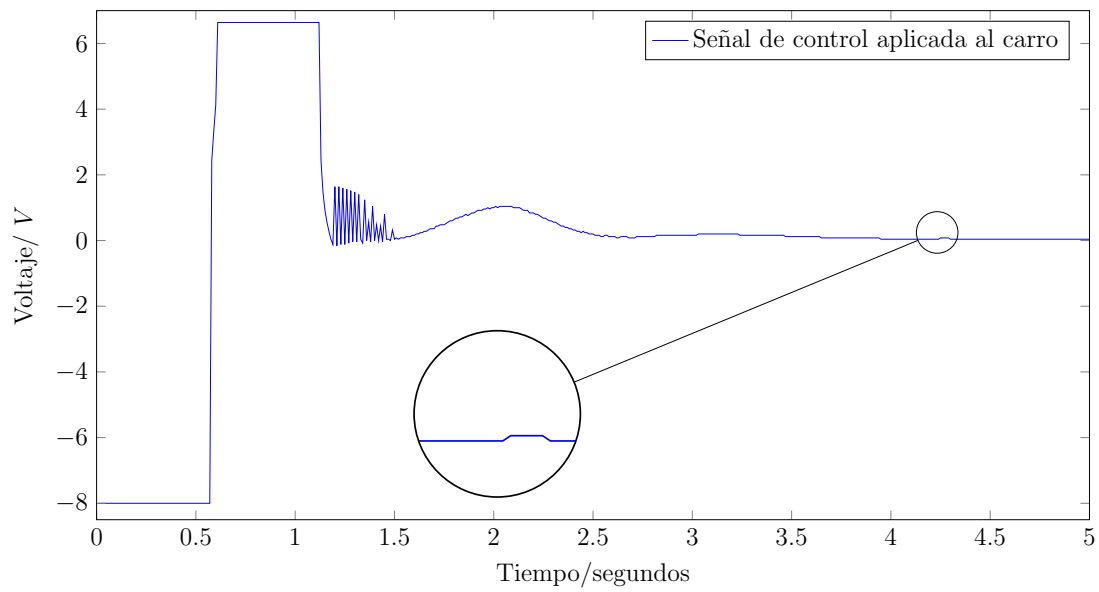
De esta arquitectura se observa que hay algunas necesidades que cumplir, como mejorar el tiempo de establecimiento, mejorar el error en estado estacionario cuando se encuentra en la posición vertical, por lo anterior se propone una arquitectura diferente.



(a) Salida del sistema



(b) Señal de error



(c) Señal de control

Figura 5.2: Respuesta del sistema con condiciones iniciales $x = 0$, $\dot{x} = 0$, $\theta = .1\text{rad}$, $\dot{\theta} = 0$, para una arquitectura de una SN y una entrada

5.2.2. La SNN con una arquitectura de una SN y tres entradas

El propósito de incrementar el número de entradas en la SN es mejorar la respuesta del sistema, disminuir el tiempo de establecimiento, que las oscilaciones en estado estacionario se reduzcan, y que las oscilaciones del péndulo sean de menor magnitud. Al proponer que en la entrada sea el error e , la derivada del error $\frac{d}{dt}e$ y la integral de error $\int e$, es con la finalidad de hacer que se asemeje el comportamiento al controlador PID, como se muestran en la Figura 5.3.

El entrenamiento de la SN se sigue realizando mediante la señal de error e , ya que

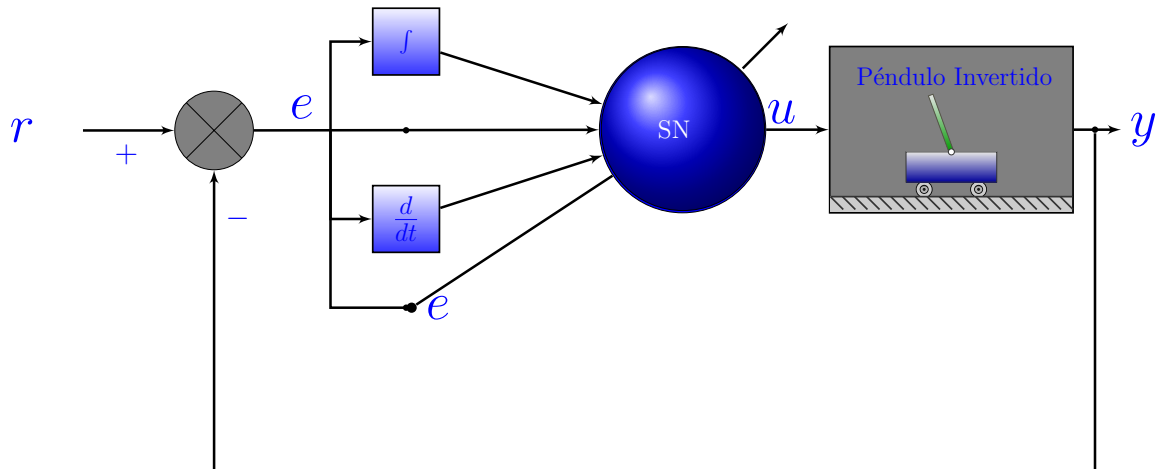
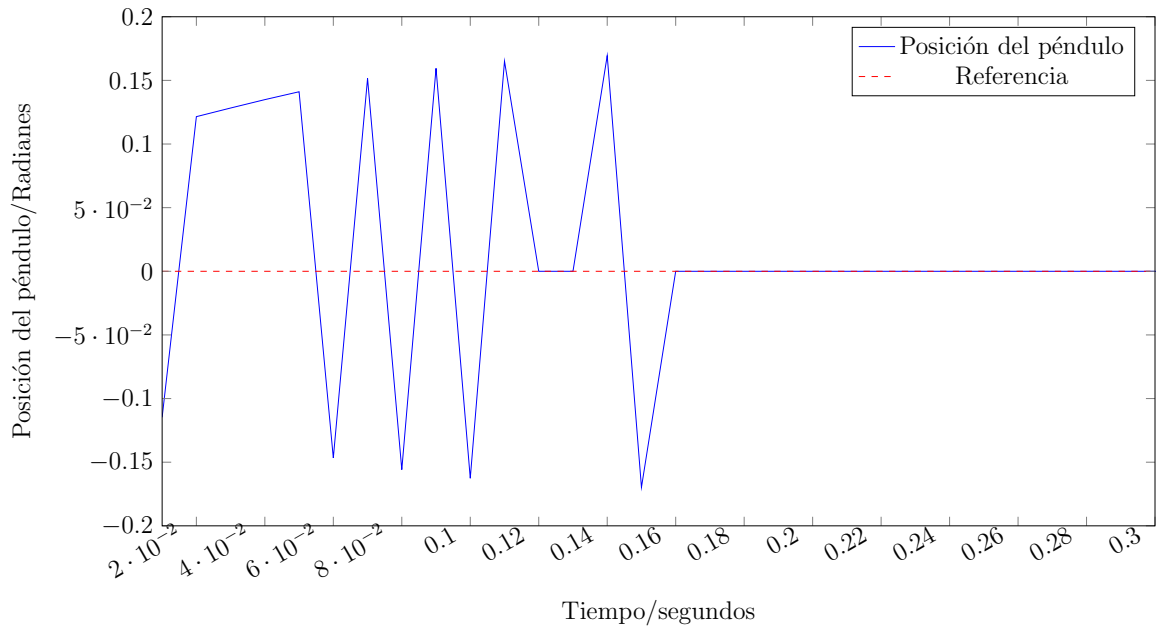


Figura 5.3: Diagrama con una arquitectura de una SN, el error, derivada del error e integral del error como entradas.

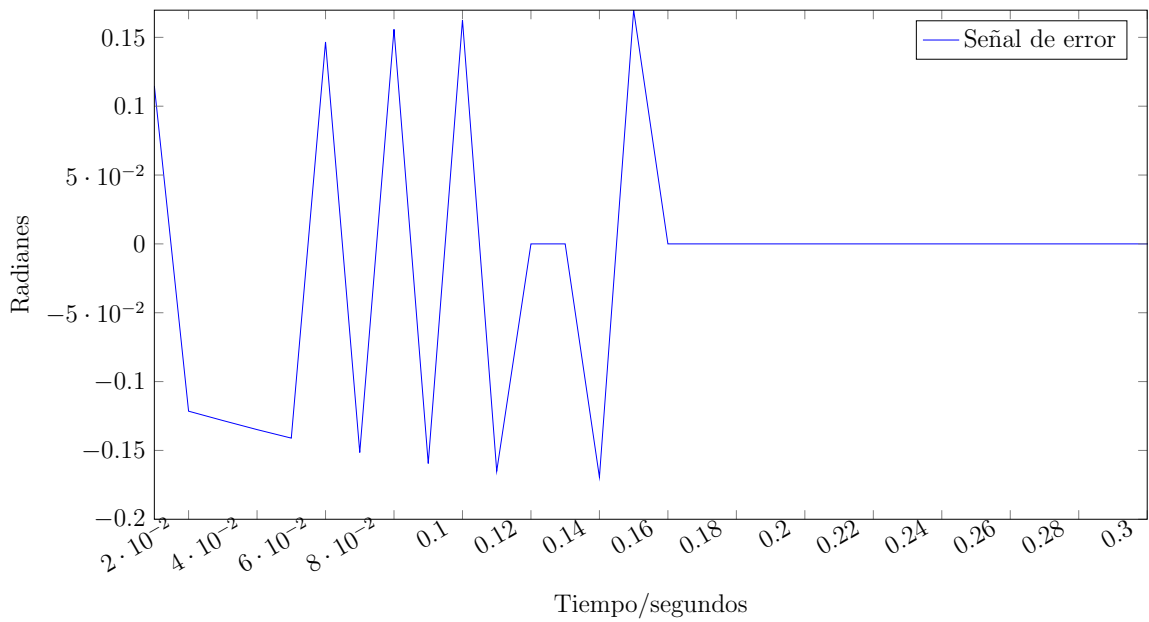
la finalidad del sistema es que el error se acerque al cero (posición vertical del péndulo)

En este caso las condiciones iniciales son cero en $\dot{\theta}$, p , \dot{p} y en $\theta = .11rad$.

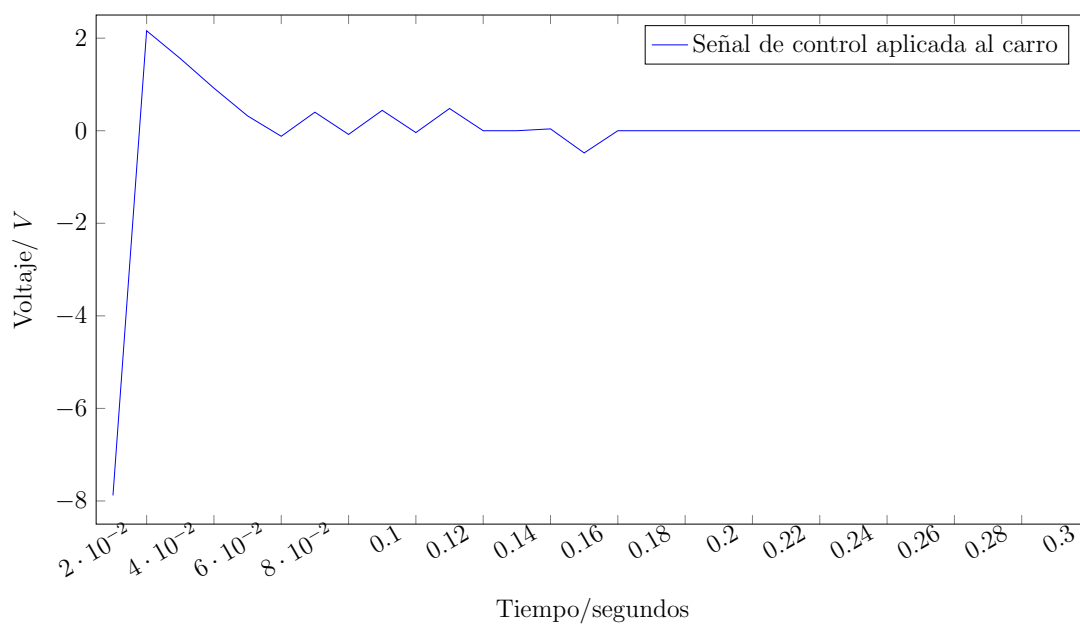
En la Figura 5.4 se muestra el comportamiento del sistema a las condiciones iniciales antes mencionadas, observando que en 5.4(a) se muestra que el péndulo hace repetidas oscilaciones que llegan hasta los 0.15 radianes, posterior a eso, hace un alto repentino cuando llega a la referencia.



(a) Salida del sistema



(b) Señal de error



(c) Señal de control

Figura 5.4: Respuesta del sistema con una arquitectura de una SN con tres entradas, con condiciones iniciales $p = 0$, $\dot{p} = 0$, $\dot{\theta} = 0$, y en $\theta = .1rad$.

Lo anterior idealmente y en simulación parece ser adecuado, sin embargo en el sistema físico es imposible por diferentes circunstancias como la inercia la gravedad, etc. En la Figura 5.4 (b) se muestra el error que existe entre la referencia y la posición. En la Figura 5.4 (c) se muestra la señal de control que se aplica al sistema para mantener el péndulo en la posición vertical, observando que el voltaje necesario es menor que se requiere en el diagrama de la Figura 5.1.

La arquitectura mostrada en la Figura 5.3 fue utilizada en el trabajo de titulación [39], aplicándola a un sistema lineal como es un motor de corriente continua, obteniendo buenos resultados.

En arquitectura, aunque proporciona mejores resultados que la anterior, sólo se está controlando una de las variables, la posición angular θ , y se deja libre la otra variable, posición lineal p , así mismo es necesario que la llegada a la referencia sea más suave haciendo necesario proponer otra arquitectura.

5.2.3. La SNN con una arquitectura de una SN y cinco entradas

La arquitectura que tiene una neurona con 5 entradas ha mejorado el desempeño de la planta en lazo cerrado, en la Figura 5.5 se muestra esta arquitectura.

Una de las características para la arquitectura con cinco entradas radica en colocar como parte de las entradas, las señales obtenidas de la salida del sistema y y la señal obtenida de la misma neurona SN. Al agregarle mayor número de entradas se genera una dinámica más rica a la entrada de la neurona, esto beneficia en gran medida al entrenamiento de la misma, ya que existe un número más grande de pesos presinápticos que se modifican para lograr la encomienda de la neurona.

La decisión de colocar estas entradas es para mejorar la dinámica de la entrada de corriente que necesita la neurona, sin embargo, estas entradas se eligen por que forman parte del mismo sistema.

Para esta arquitectura se tomó también la señal del error e para el entrenamiento de la neurona.

Con esta arquitectura se logra mejorar la respuesta del sistema como se observa en la Figura 5.6(a) la amplitud de las oscilaciones del péndulo se reducen con el transcurso del tiempo hasta llegar a estabilizarse en 0.27 seg . El tiempo de muestreo utilizado es de 10 ms , con ello, el tiempo necesario para la estabilización son menores

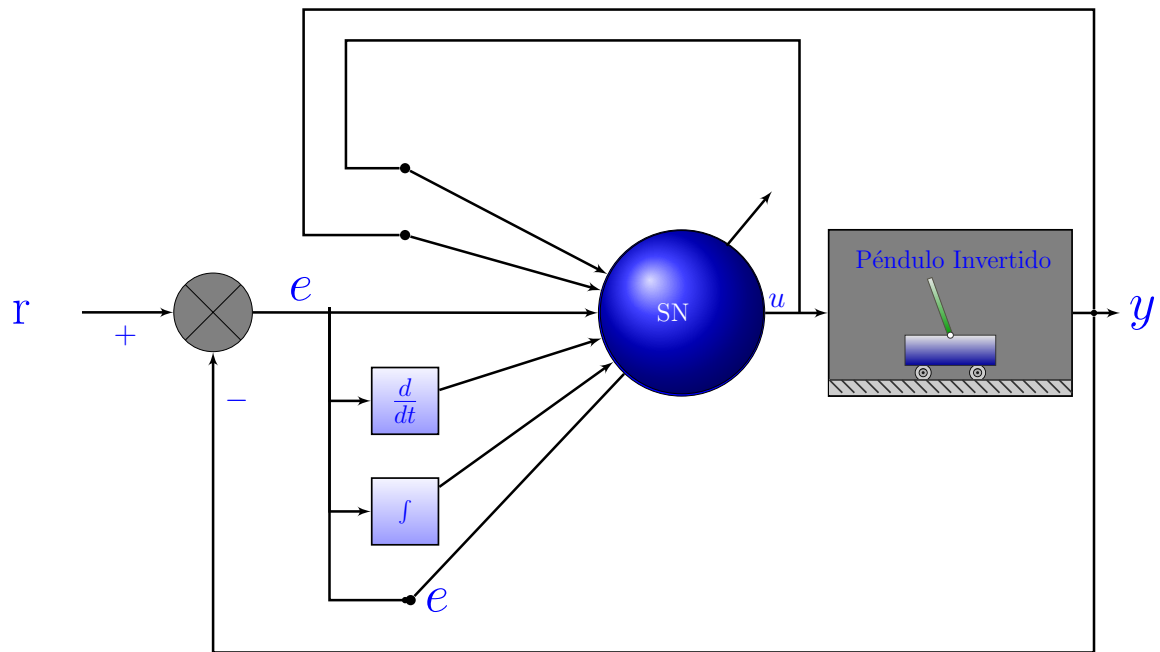
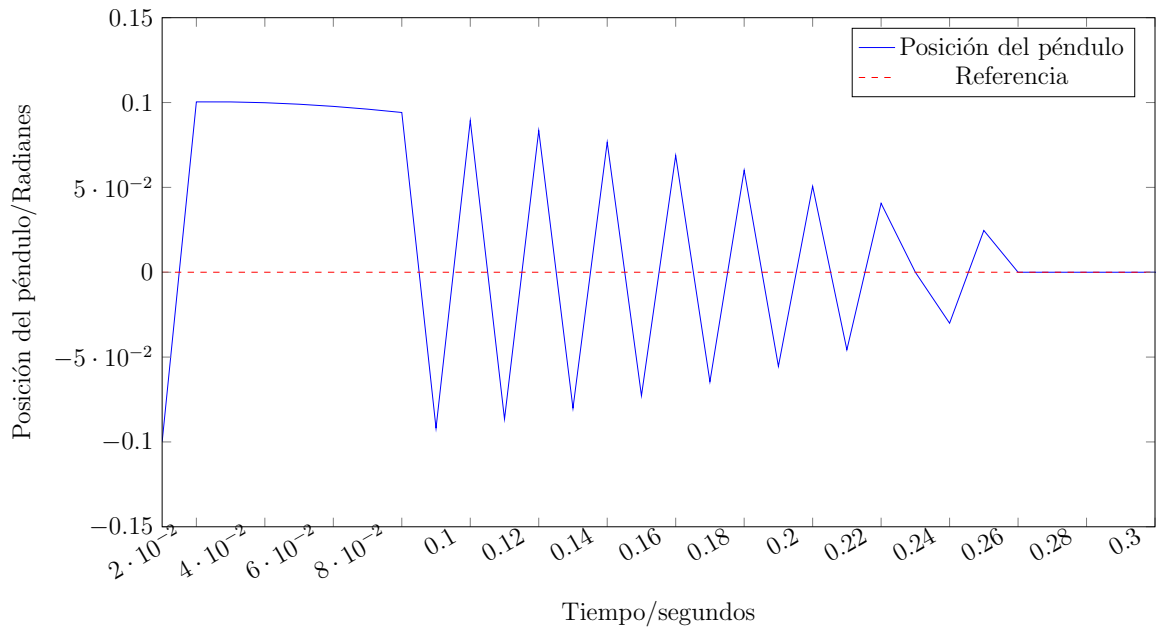


Figura 5.5: Diagrama con una arquitectura de una SN, y cinco entradas.

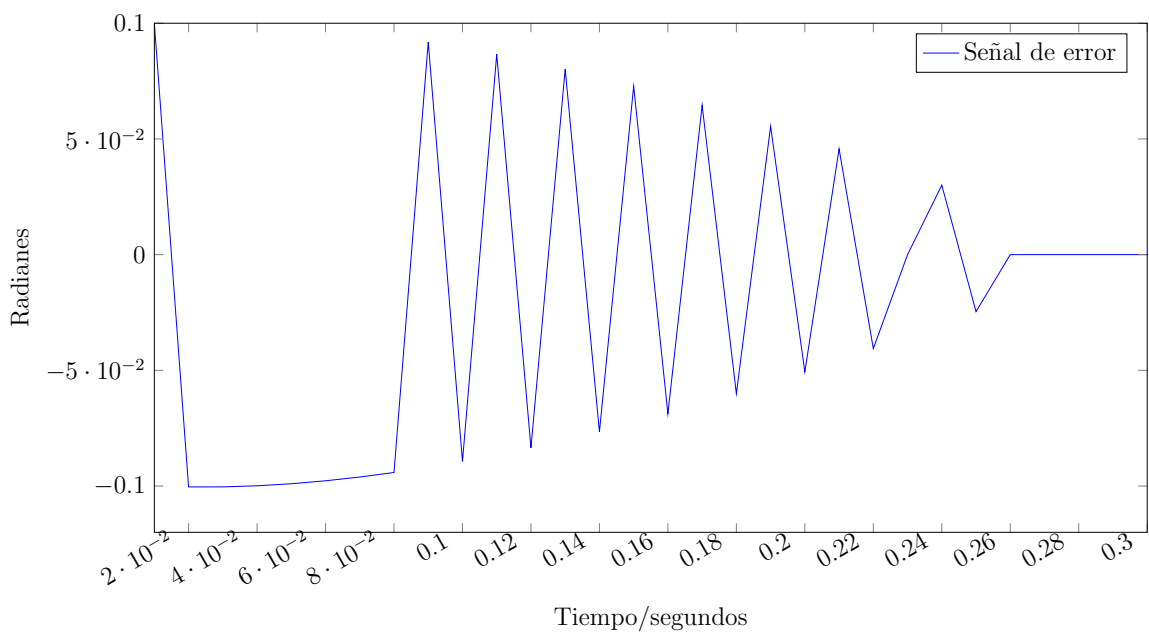
de 1 *seg.* En la Figura 5.6(b) se muestra la señal de error. En la Figura 5.6(b) se muestra la señal de control teniendo un pico voltaje máximo de -1.5 volts.

En la subsección 5.2.1 se describen tres arquitecturas diferentes, aunque no se limita a estas tres, en la Tabla 5.1 se muestra una comparativa entre las diferentes arquitecturas propuestas en este trabajo de investigación.

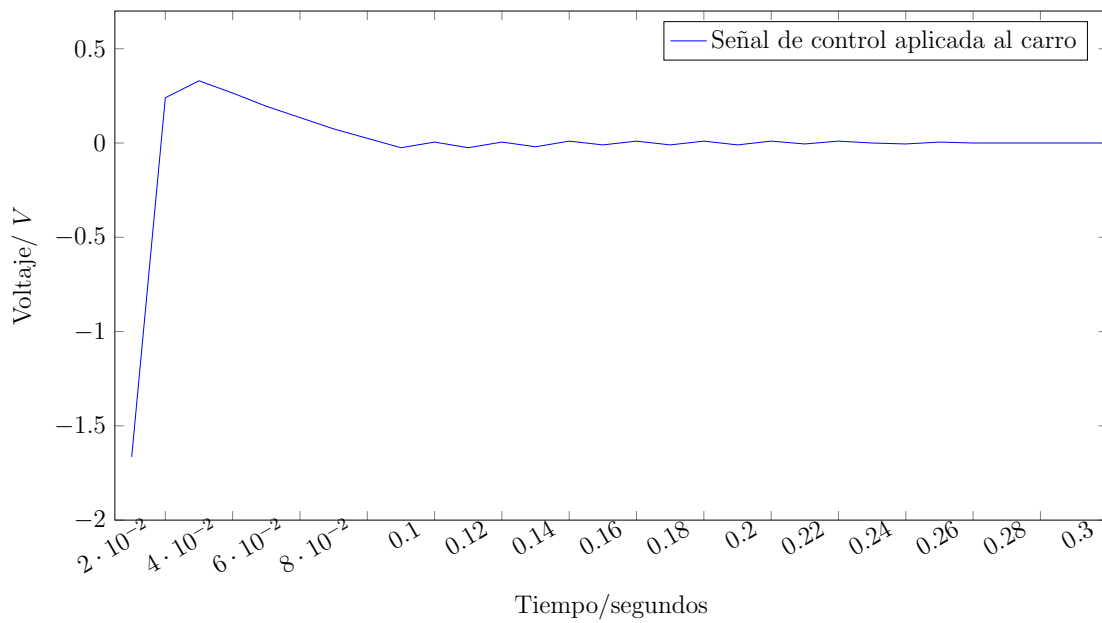
Donde la arquitectura se define como la combinación de algunas de las entradas que tiene la SN:



(a) Salida del sistema



(b) Señal de error



(c) Señal de control

Figura 5.6: Respuesta del sistema con una arquitectura de una SN con cinco entradas y con condiciones iniciales $p = 0$, $\dot{p} = 0$, $\dot{\theta} = 0$, y $\theta = .1rad$.

Variable	Descripción
e	Señal de error
D	Derivada del error
I	Integral del error
S	Salida de la neurona
Y	Salida del sistema

Tabla 5.2: Descripción de las entradas de la SN.

En la Tabla 5.1 se muestran una serie de resultados numéricos con los que se puede concluir cuál de las arquitecturas tienen un mejor comportamiento en el control del carro péndulo invertido, cabe hacer mención que las condiciones iniciales son las mismas para todos los experimentos; $x = 0$, $\dot{x} = 0$, $\theta = .1$ rad, $\dot{\theta} = 0$, y que la finalidad sólo es controlar la posición del péndulo, dejando libre la posición del carro.

El primero de los resultados obtenido es el promedio del error, lo que esperamos obtener es un promedio de la señal del error que solo nos dice qué arquitectura tiene el menor error, y en este caso la arquitectura eDSY tiene el promedio menor, este parámetro está medido en radianes.

Otro término que se analiza es la desviación estándar del error, que nos ayuda a observar que señal está más cerca del cero, y como para este trabajo el cero es la posición vertical de péndulo, es una forma de mostrar este acercamiento, siendo la arquitectura eIY la que tiene el mejor desempeño.

Estos dos procedimientos, el promedio y la desviación estándar se aplicaron a la señal de control identificando a la arquitectura que mejor se desempeña. Sin embargo, el procedimiento de obtener la integral de la señal de control se utiliza para identificar a la arquitectura que genera menor cantidad de energía, siendo la arquitectura de cinco entradas eDISY la que menos produce energía.

5.3. Control de carro péndulo invertido con una arquitectura de tres SN

En esta sección se propone una SNN con una arquitectura con tres SN creando así una red de neuronas SNN, como se muestra en la Figura 5.7, con una capa de entrada con dos SN y una capa de salida con una SN.

La finalidad de esta arquitectura es controlar la posición del carro p , sobre la horizontal como se ve en la Figura C.2, y la posición angular θ con respecto a la vertical como se muestra en la Figura C.3. En la Figura 5.7 se muestran dos señales de error e_1 y

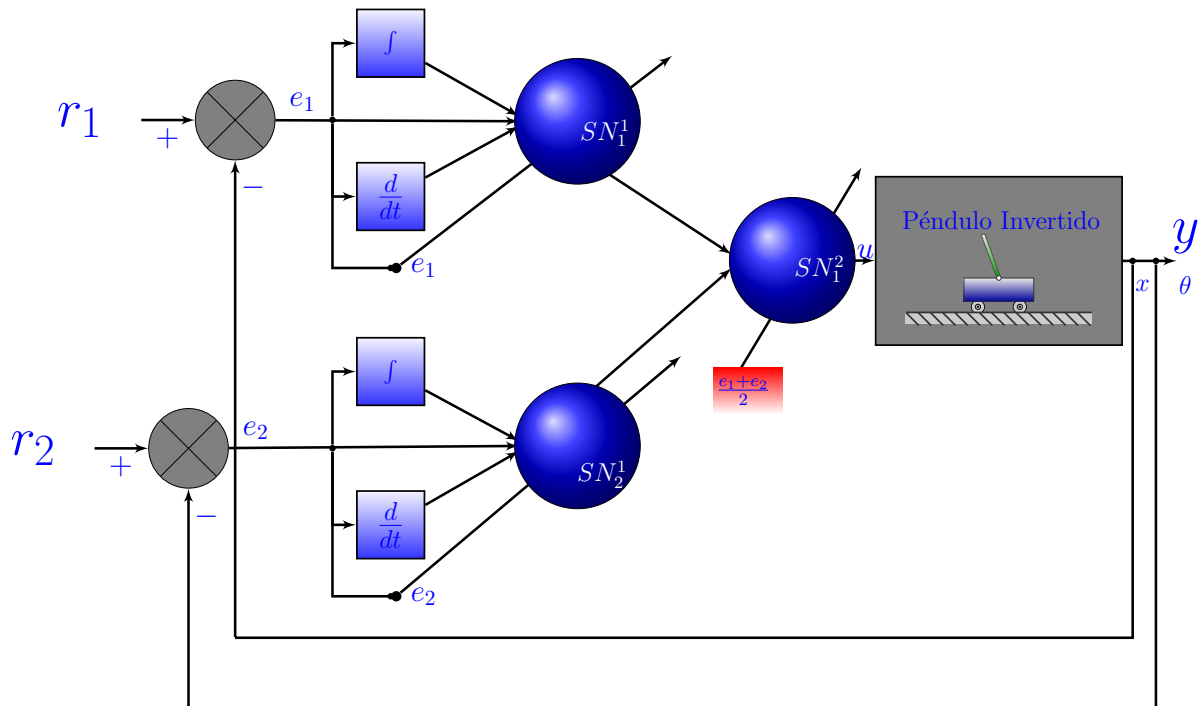


Figura 5.7: Diagrama de una SNN con una arquitectura con dos SN en la capa de entrada, y una SN en la capa de salida.

e_2 , que se obtiene mediante las siguientes expresiones:

$$e_1 = r_1 - x$$

$$e_2 = r_2 - \theta,$$

donde:

r_1 es la posición lineal deseada del carro

r_2 es la posición angular deseada del péndulo

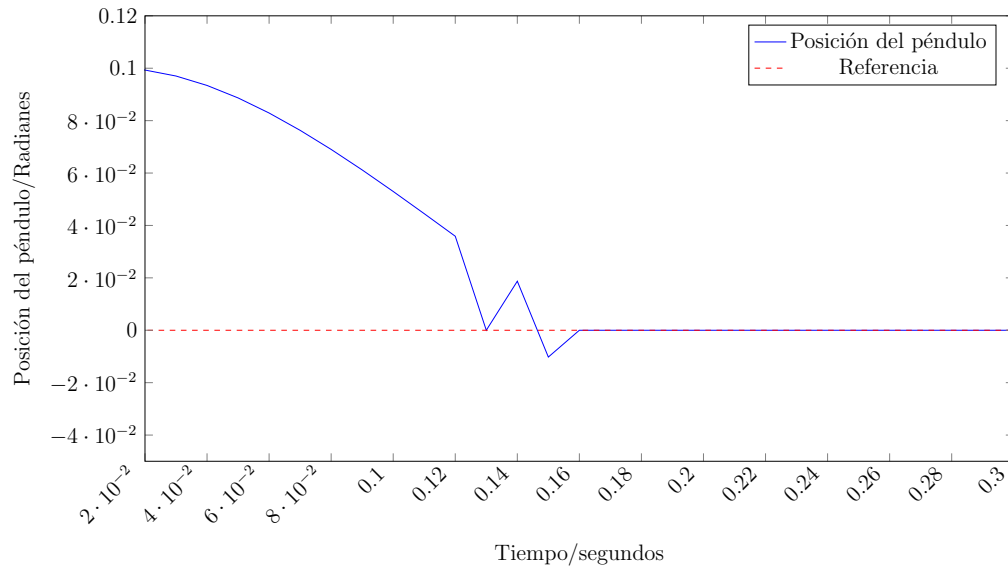
La arquitectura propuesta tiene una neurona para procesar la señal de error de la posición del carro y otra neurona para procesar la señal de la posición angular, y una tercer neurona para darle la salida a la señal de control.

La neurona SN_1^1 es entrenada mediante la señal de error e_1 y una tasa de aprendizaje de $\eta = 3$, la neurona SN_2^1 es entrenada mediante la señal de error e_2 y una tasa de aprendizaje $\eta = 3$, y la neurona SN_1^2 es entrenada con el promedio entre los errores $(e_1 + e_2)/2$ y una tasa de aprendizaje de $\eta = .3$.

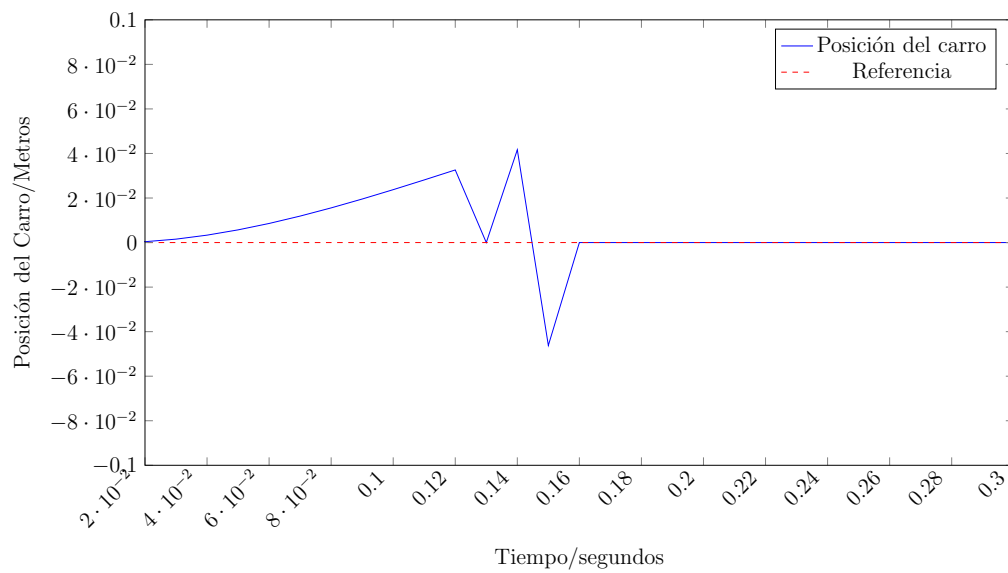
En las Figuras 5.8 se muestran resultados del comportamiento del sistema carro péndulo invertido, cuando las condiciones iniciales $p \neq 0$, $\theta \neq 0$, $\dot{p} = 0$ y $\dot{\theta} = 0$.

En la Figura 5.8 las condiciones iniciales propuestas $p = 0$, $\theta = .1rad$, $\dot{p} = 0$ y $\dot{\theta} = 0$; donde la Figura 5.8(a) muestra la posición angular del péndulo respecto a la vertical, observando que en un tiempo de 0.16 *seg.* se establece en la posición vertical, llegando a $r_2 = 0$. En la Figura 5.8(b) se muestra la posición lineal del carro, observando que en el tiempo 0.16*seg.* el carro llega a la posición fijada por la $r_1 = 0$, en la Figura 5.8(c) se muestra la señal de control requerida para llevar al sistema a las referencias r_1 y r_2 deseadas, y en la Figura 5.8(c) se muestra la señal de error para las dos variables, observando que el error en ambas, es cero en el mismo tiempo de 0.16 *seg.*

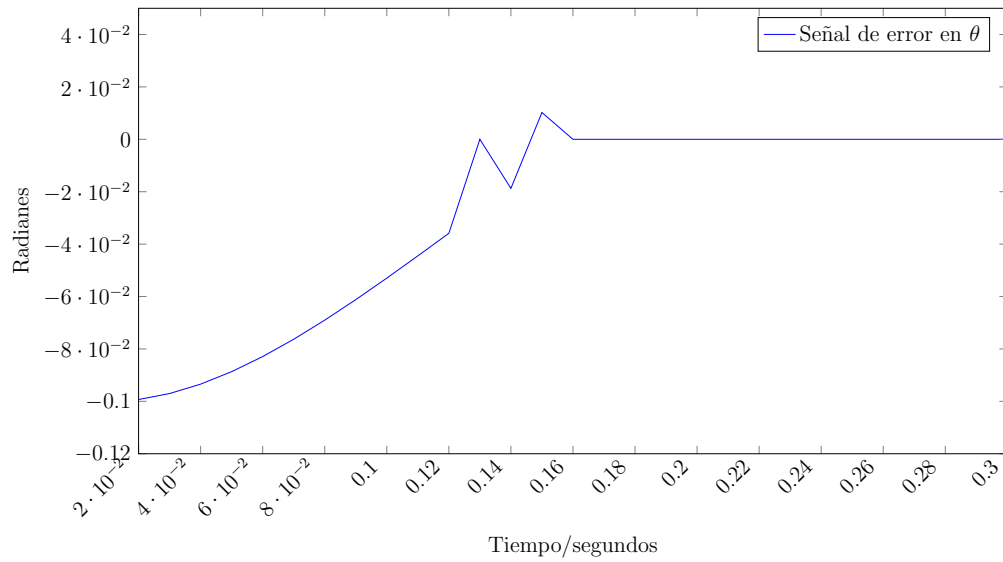
En esta sección se mostró el comportamiento de una SNN con tres neuronas, con un entrenamiento individual para cada neurona, con la finalidad de mantener el control sobre las dos variables p y θ , con esta arquitectura se realizaron diferentes experimentos, con condiciones iniciales diferentes obteniendo resultados en simulación que se aproximan a una respuesta de un sistema físico, sin embargo, esta arquitectura no es del todo práctica, por que es necesario colocar mas de una entrada en las SN, además de que el entrenamiento individual no es práctico, ya que se pierde la esencia de las redes neuronales, en donde todas las neuronas comparten la misma señal de entrenamiento, aunado a ésto, como se mencionó en capítulos anteriores la tasa de aprendizaje η no tiene un método para calcularla, y en este caso, son tres valores que se tienen que proponer, siendo esto un poco complicado. Por lo descrito anteriormente, se propone una arquitectura diferente.



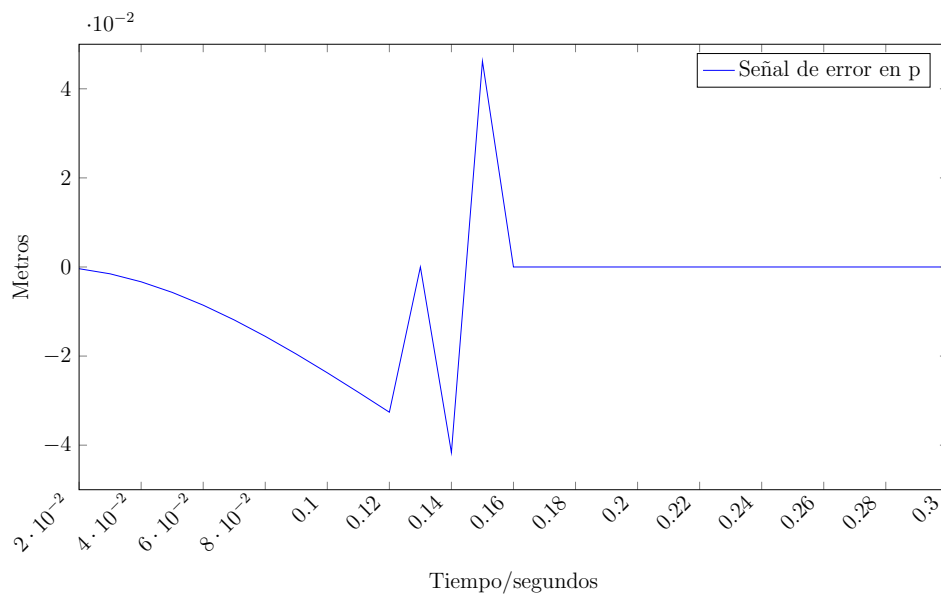
(a) Posición angular



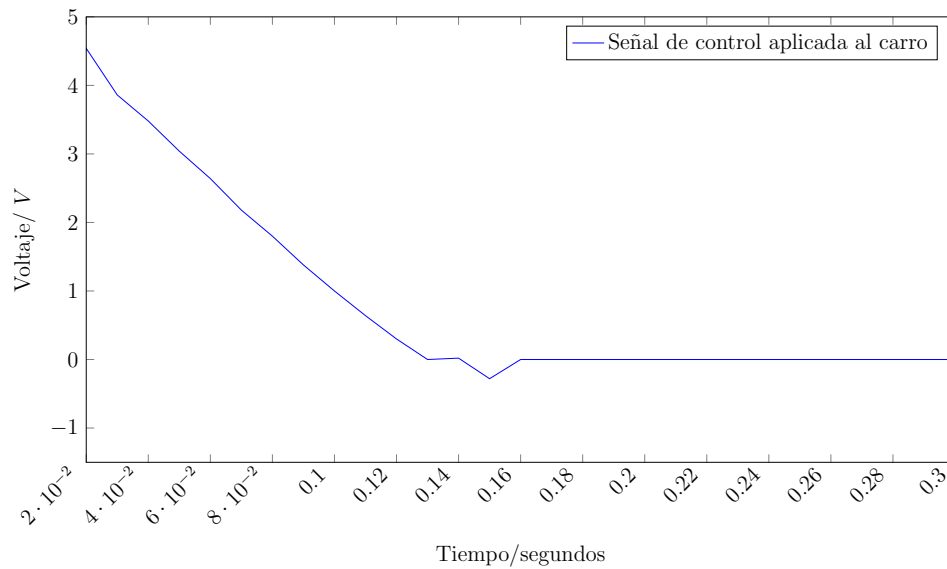
(b) Posición lineal



(c) Señale de error para θ



(d) Señale de error para p



(e) Señal de control

Figura 5.8: Respuesta del sistema con condiciones iniciales $p = 0$ y $\theta = .1rad$ con una arquitectura de tres neuronas y entrenamiento separado.

5.4. Control de carro péndulo invertido con una SNN

En la Sección 5.3 se propone una arquitectura en la que se requiere un entrenamiento individual para cada neurona, sin embargo, ésta arquitectura no es práctica y no se tiene la estructura de una red neuronal tradicional.

Para esta sección, se propone otra SNN con una arquitectura de tres capas; una capa de entrada, una capa oculta, y una capa de salida, como se muestra en la Figura 5.9. La capa oculta tiene una neurona SN_1^1 , la capa de salida tiene una neurona SN_1^3 y la capa oculta tiene i número de neuronas SN_i^2 , donde $i = 1, 2, 3, \dots$

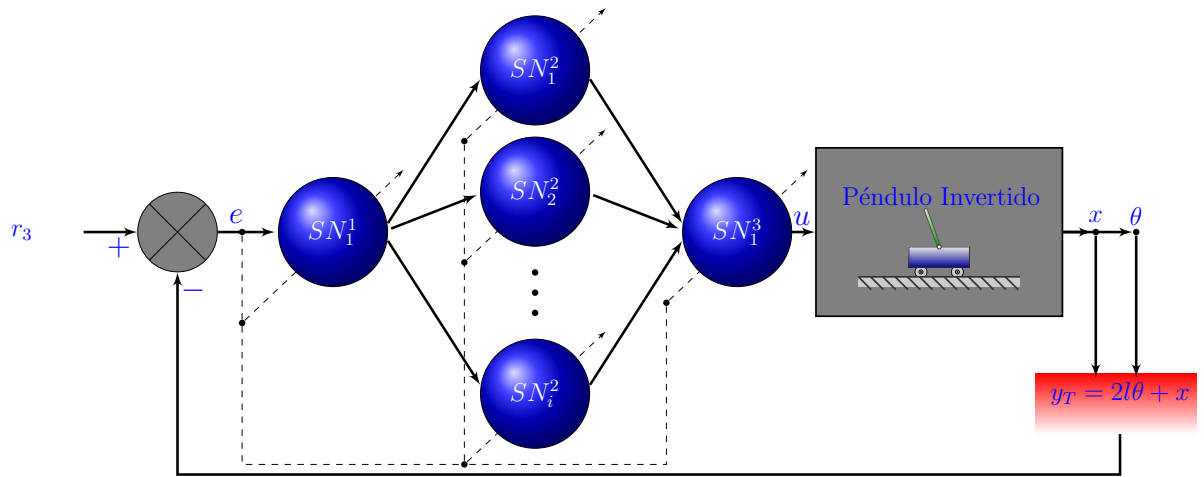


Figura 5.9: Diagrama general de control con una SNN.

El entrenamiento de esta red está basado en la señal de error e misma que se obtiene de la diferencia entre la referencia r_3 y la señal y_T , es decir:

$$e = r_3 - y_T \quad (5.2)$$

Esta señal es empleada para realizar el entrenamiento de la SNN, para esto es necesario combinar las dos variables del sistema p y θ en una sola variable llamada y_T .

Como se observa en la Figura 5.9, y_T se obtiene de la combinación de las dos señales de interés, p y θ , como se observa en la Figura 5.10.

Como el objetivo de control es mantener el péndulo en la posición vertical, la distancia en el extremo superior del péndulo queda como:

$$2l \sin \theta \quad (5.3)$$

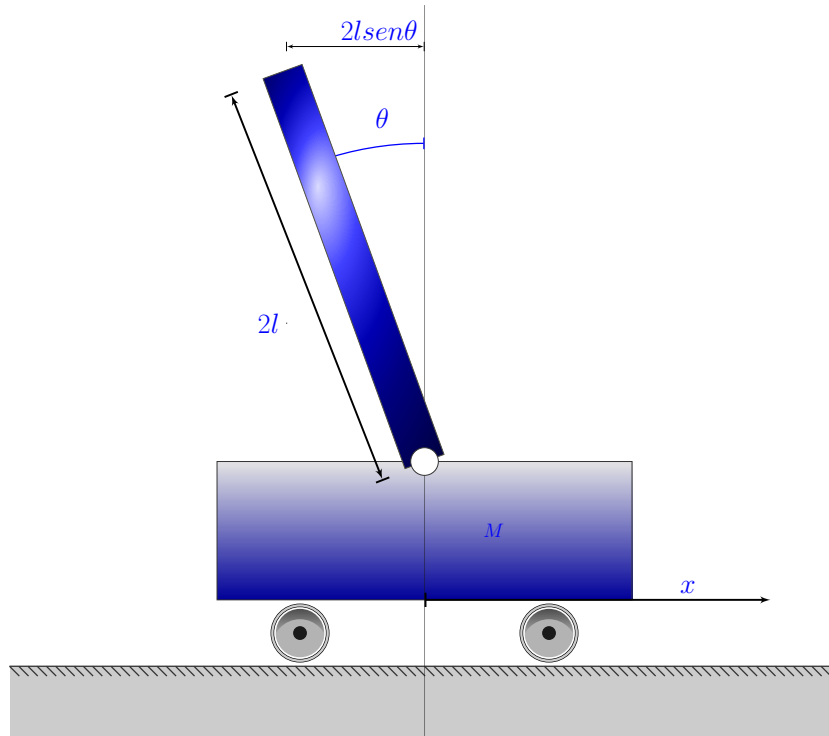


Figura 5.10: Representación de la combinación de las variables x y θ .

La ecuación (5.3) es del tipo no lineal por la función *seno*, que por su naturaleza y dado que se encuentra como la retroalimentación del sistema, hace que el carro péndulo invertido no se comporte de forma deseada, para evitar esto, se usa la serie de Taylor para aproximar a la función $\text{sen}\theta$ al rededor del origen, obteniendo:

$$2l\theta \quad (5.4)$$

Por lo tanto, la nueva salida del sistema se define como:

$$y_T = 2l\theta + x$$

Misma que se utiliza para calcular el error como se dio en la ecuación (5.2) siendo la entrada de la neurona SN_1^1 y la señal con la que se entrena a la red.

Para realizar los experimentos con esta arquitectura se decide realizar diferentes cambios como son:

1. La tasa de aprendizaje de la SNN η , en los experimentos anteriores se maneja un valor fijo para toda la simulación, para esta arquitectura se propone que el valor de η sea variable y que dependa del error, es decir que si el error es grande

el valor de η es grande, con la finalidad de que se acerque a la referencia con rapidez, y si el error es pequeño el valor de η es pequeño, ésto con el objetivo de darle precisión cuando está cerca de la referencia. En la simulación con la arquitectura propuesta se consideran tres valores: $\eta = 1$ sí $e \geq 1$ rad, $\eta = 0.5$ sí $0.1 \leq e < 1$, y $\eta = 0.2$ sí $e < 0.1$.

2. El modelo de la planta que en las simulaciones anteriores se maneja un modelo discreto, en ésta arquitectura se propone utilizar el modelo de la planta en continuo, teniendo que utilizar una herramienta de software de MatLab como auxiliar para la solución de las ecuaciones diferenciales como se muestra a continuación.

$$tspan = [Tm * (ent1 - 1), ent1 * Tm];$$

$$dx = @(t, x) \begin{bmatrix} x(3); \\ x(4); \\ \frac{(-m*m*g*l*\sin(x(2))*\cos(x(2))-(m*l*J)*m*l*x(4)*x(4)*\sin(x(2))-(m*l*J)*u)}{(-M+m)*(m*l*J)+m*m*l*\cos(x(2))*\cos(x(2))}; \\ \frac{(-m*g*l*(M+m)*\sin(x(2))+m*m*l*x(4)*x(4)*\sin(x(2))*\cos(x(2))+m*l*\cos(x(2))*u)}{(-M+m)*(m*l*J)+m*m*l*\cos(x(2))*\cos(x(2))}; \end{bmatrix}$$

$$[et, ex] = ode45(dx, tspan, x0);$$

$$theta = ex(end, 2);$$

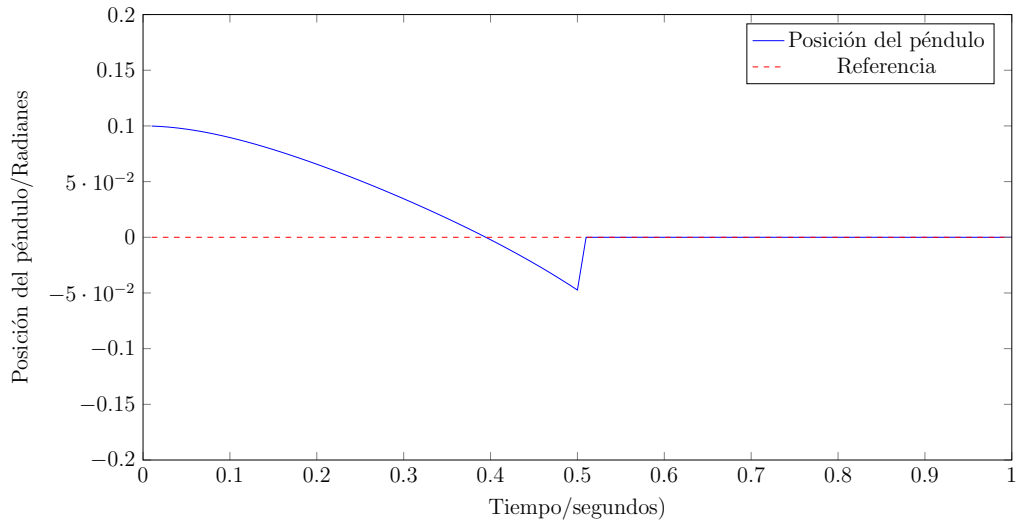
$$exis = ex(end, 1);$$

$$thetadot = ex(end, 3);$$

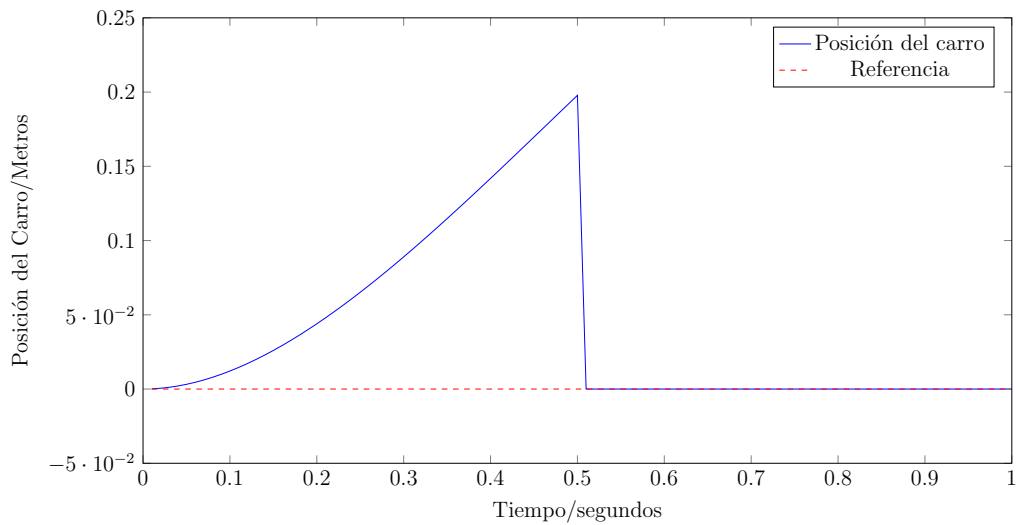
$$exixdot = ex(end, 4);$$

En esta arquitectura se simulan cuatro casos diferentes, con diferentes condiciones iniciales del carro péndulo invertido, donde se obtuvieron los siguientes resultados. Para iniciar los experimentos se consideran las condiciones iniciales $[\dot{x} = 0, \dot{\theta} = 0, x = 0$ mts, $\theta = 0.1$ rad].

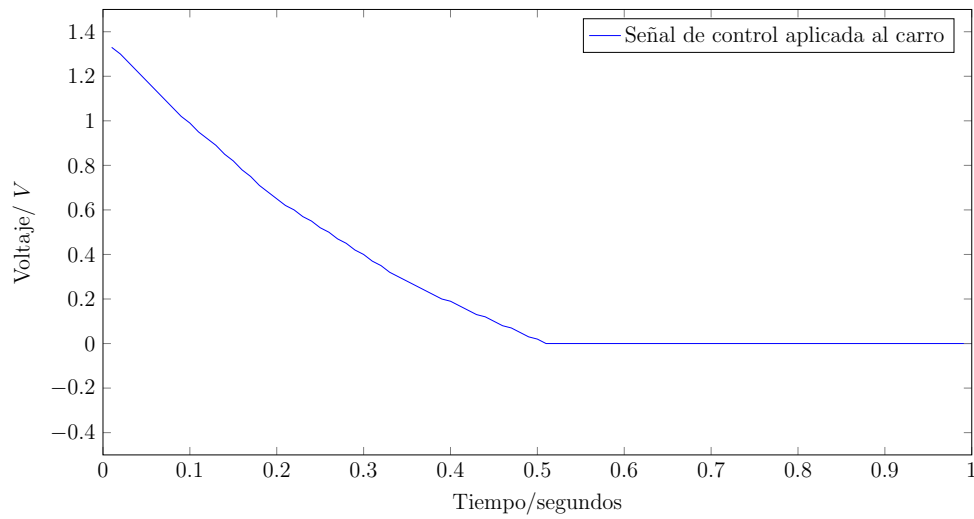
En la Figura 5.11 se observa el comportamiento del sistema carro péndulo invertido, la Figura 5.11(a) se muestra la posición del péndulo invertido que inicia en 0.1



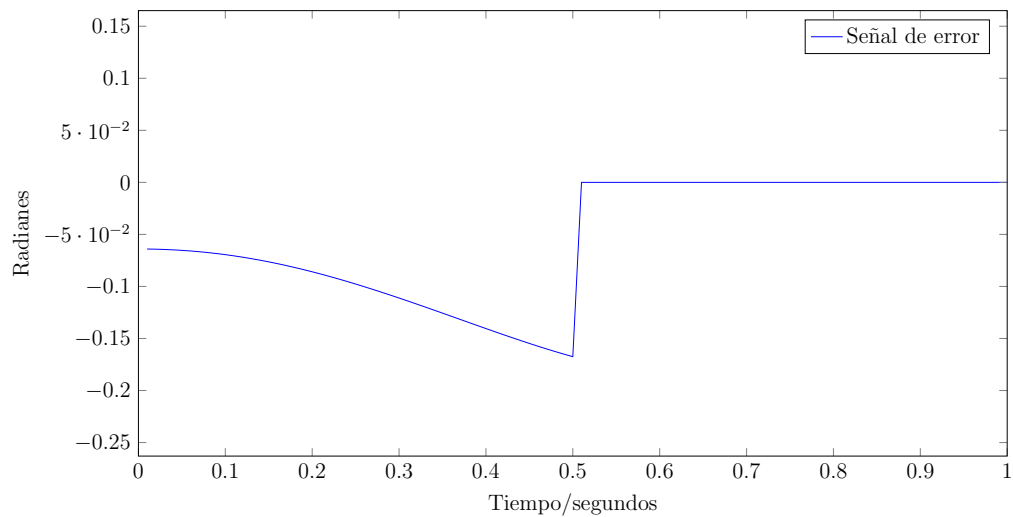
(a) Posición angular



(b) Posición lineal



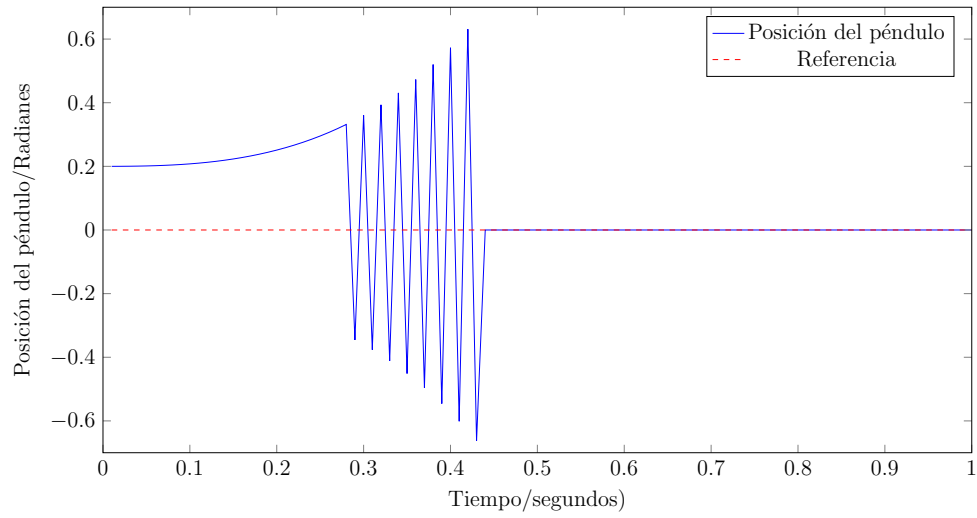
(c) Señal de control



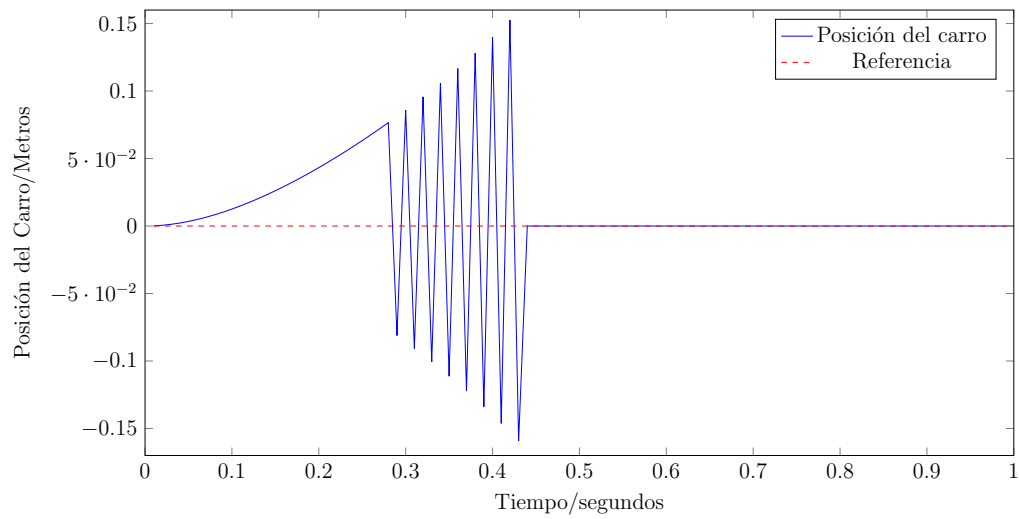
(d) señal de error

Figura 5.11: Respuesta del sistema con condiciones iniciales diferentes de cero con una arquitectura de tres capas, una de entrada, una capa de salida y una capa oculta con i número de neuronas.

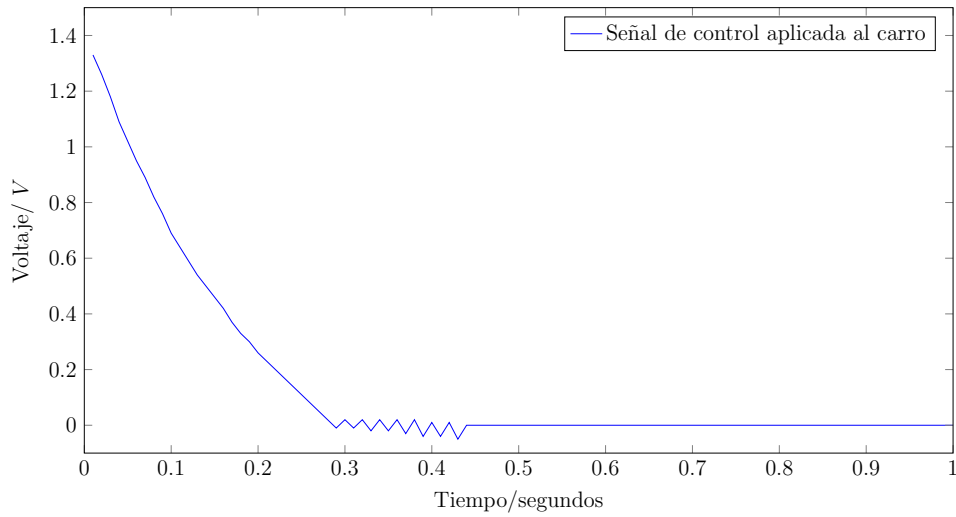
rad, y que busca estabilizarse en la referencia de 0 rad, siendo ésta la posición vertical. Llevar el péndulo a la posición deseada tarda 0.5 segundos. En la Figura 5.11(b), se muestra la posición del carro, iniciando en cero metros y teniendo una referencia de 0 metros, sin embargo es necesario que el carro se mueva ya que el péndulo es subactuado y depende del movimiento del carro. En la Figura 5.11(c) se muestra la señal de control, misma que se aplica al motor para lograr el objetivo. En la Figura 5.11(c), se muestra la señal de error que se obtiene de la combinación de las dos variables x y θ . En las Figuras 5.11, 5.13-5.12, 5.14, se observa el comportamiento del sistema con diferentes condiciones iniciales, mostrando que el tiempo que tarda en estabilizar al sistema es menor a 0.5 segundos. Ésto nos da pauta a proponer experimentos físicos con la planta carro péndulo invertido.



(a) Posición angular



(b) Posición lineal



(c) Señal de control

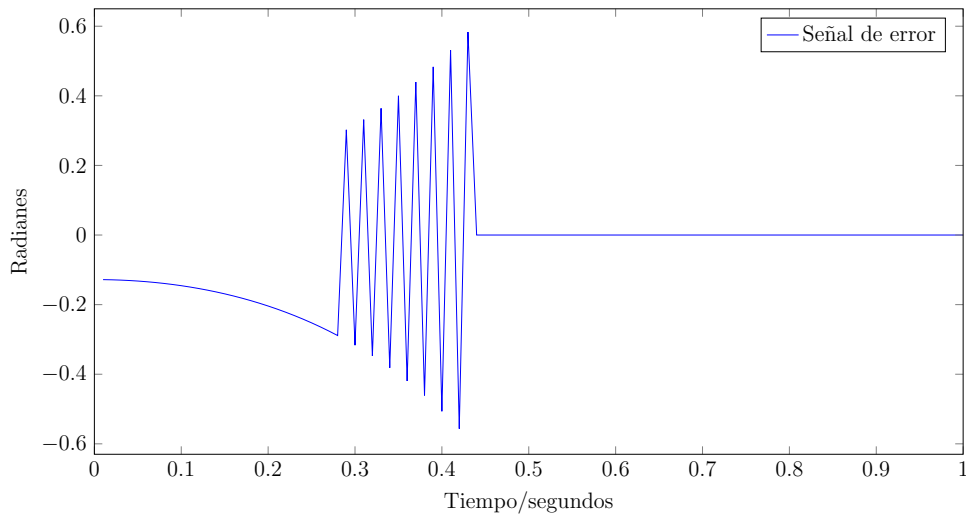
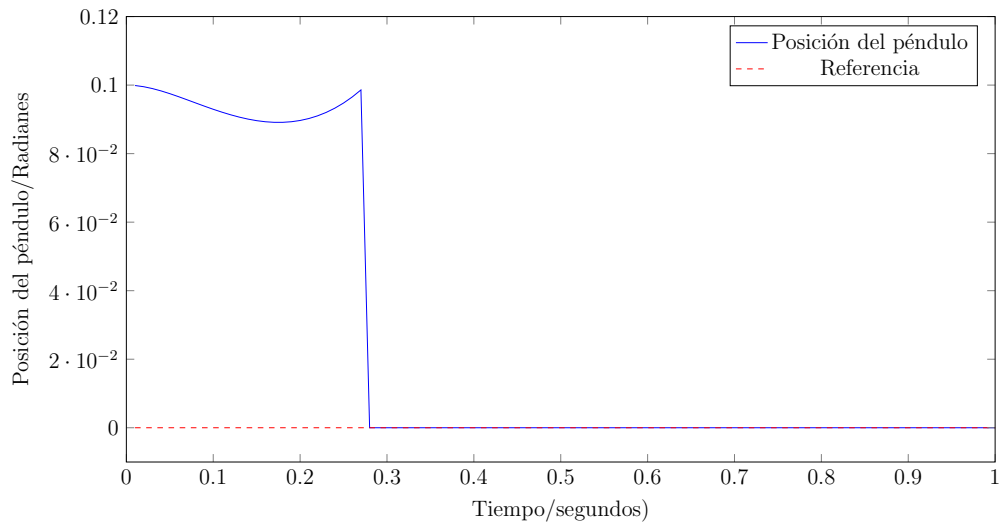
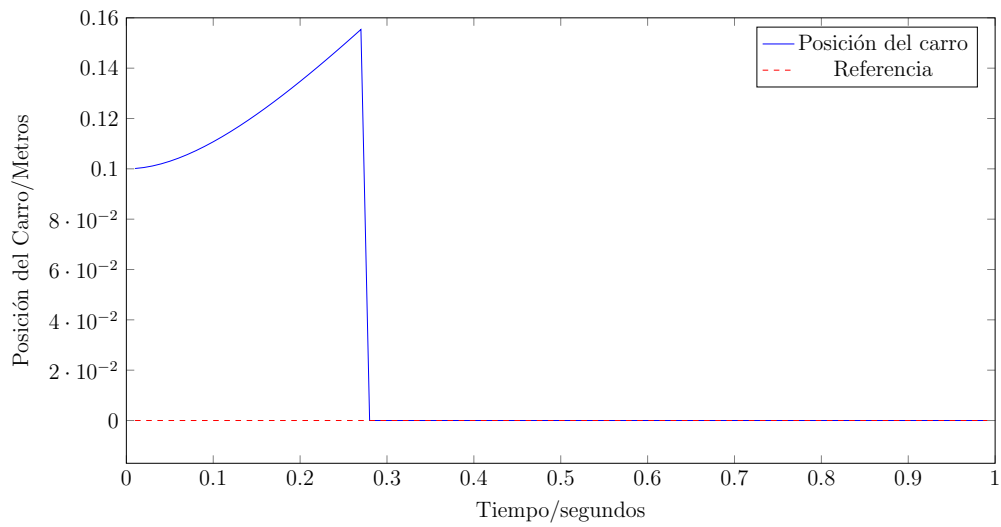
(d) señal de error e

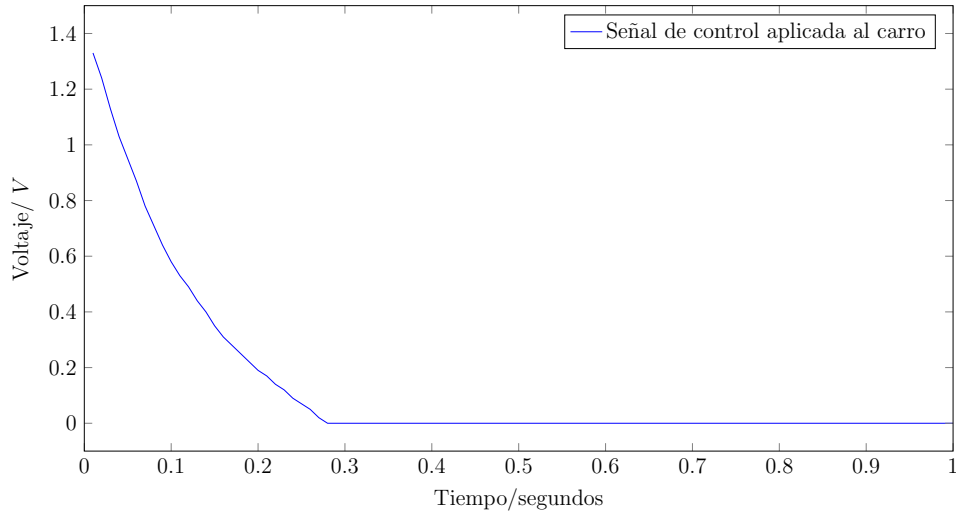
Figura 5.12: Respuesta del sistema con condiciones iniciales $[\dot{x} = 0, \dot{\theta} = 0, x = 0 \text{ mts}, \theta = 0.2 \text{ rad}]$ con una arquitectura de tres capas, una de entrada, una capa de salida y una capa oculta con i número de neuronas.



(a) Posición angular



(b) Posición lineal



(c) Señal de control

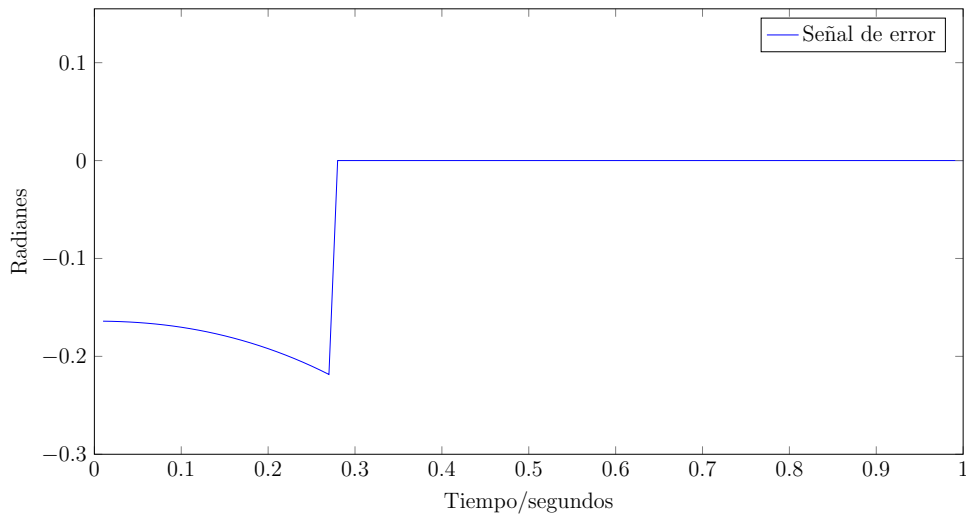
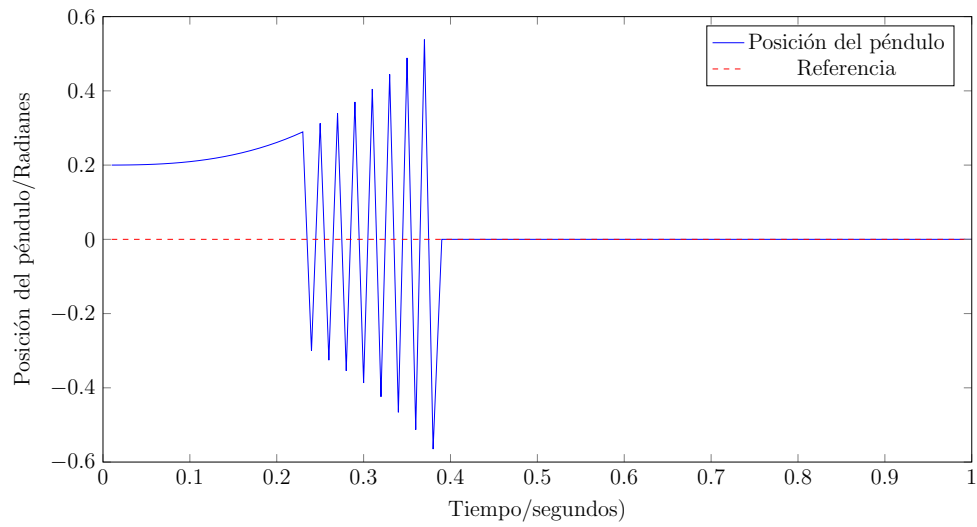
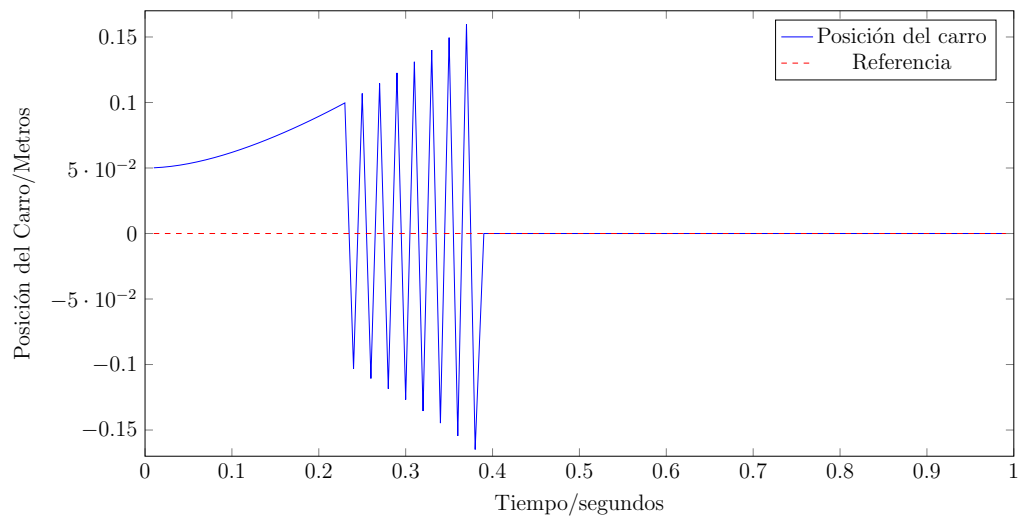
(d) señales de error e

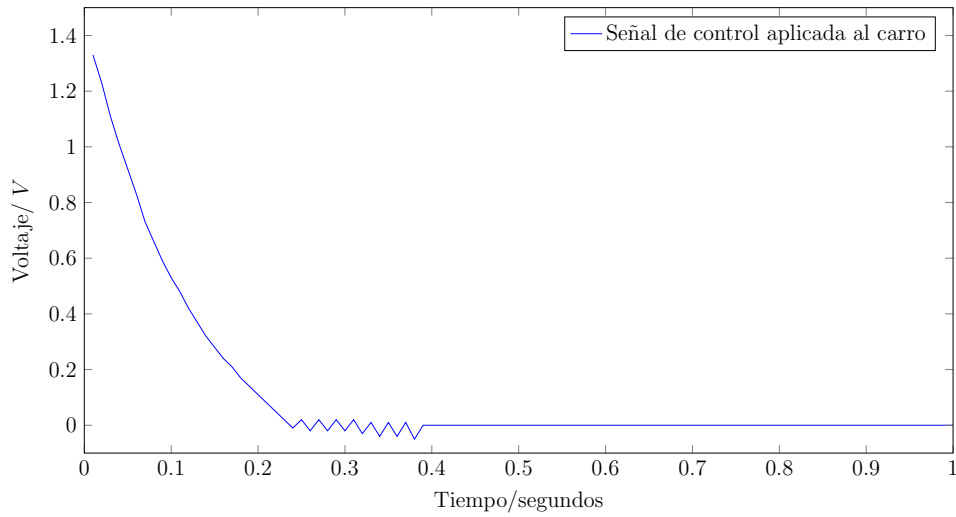
Figura 5.13: Respuesta del sistema con condiciones iniciales $[\dot{x} = 0, \dot{\theta} = 0, x = 0.1$ mts, $\theta = 0.1$ rad] con una arquitectura de tres capas, una de entrada, una capa de salida y una capa oculta con i número de neuronas.



(a) Posición angular



(b) Posición lineal



(c) Señal de control

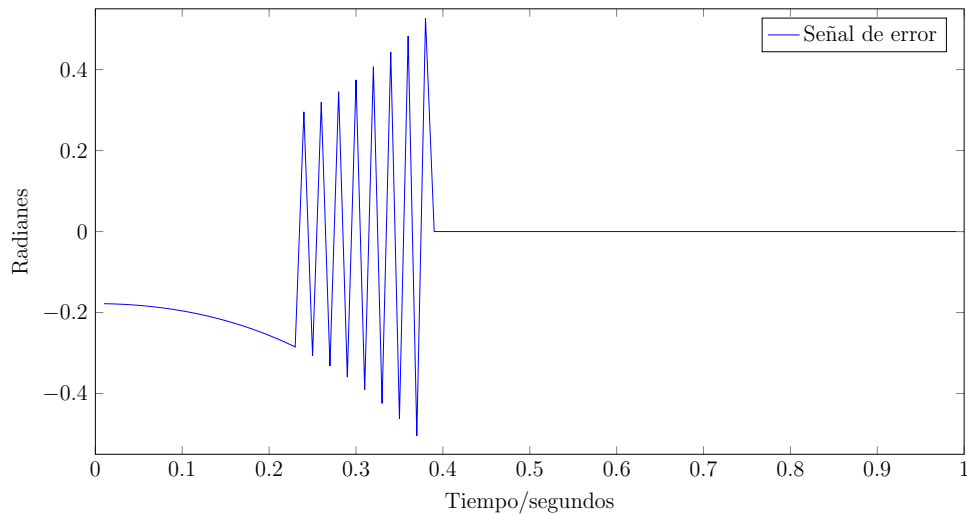
(d) señales de error e

Figura 5.14: Respuesta del sistema con condiciones iniciales $[\dot{x} = 0, \dot{\theta} = 0, x = 0.05$ mts, $\theta = 0.2$ rad], con una arquitectura de tres capas, una de entrada, una capa de salida y una capa oculta con i número de neuronas.

5.5. Conclusiones

Las SNN, son redes neuronales que hasta hace algunos años se empleaban para diferentes actividades diferentes al control, sin embargo en este trabajo se muestran diferentes arquitecturas que funcionan como un sistema de control, todas las arquitecturas mostradas no son únicas, ni en el número de capas ni en el número de neuronas, sin embargo, la arquitectura siempre dependerá de la complejidad del sistema a controlar.

La contribución original de este trabajo de investigación es proponer un controlador diferente a los que existen en la actualidad, usando la redes neuronales de tercera generación que tienen la característica de adaptar los pesos en línea en un tiempo reducido, es decir teniendo un aprendizaje de los eventos pasados, para obtener los resultados deseados. De lo anterior, se concluye que dicha hipótesis es verdadera y por lo tanto, el objetivo del trabajo de investigación se cumplió satisfactoriamente. La aplicación de las SNN al procesamiento digital de señales, en especial al seguimiento de señales médicas tiene un gran impacto ya que estas tiene la capacidad de reproducir con una excelente aproximación las señales deseadas

5.6. Trabajos futuros

- Realizar la implementación física del controlador SNN para el sistema subactuado carro-péndulo invertido.
- Realizar un estudio comparativo detallado en implementación física entre otros controladores.
- Realizar pruebas con un sistema SIMO con dichos controladores.
- Realizar pruebas con un sistema MIMO con las SNN como controladores.
- Realizar prueba y análisis de estabilidad para SNN.

Tabla 5.1: Arquitectura propuestas

Arquitectura	Promedio del error	Promedio del control	Desviación estándar del error	Desviación estándar del control	Integral del control
eDISY	0.0181	0.0044	0.0343	0.1762	0.3875
eDIS	0.0134	0.0283	0.0333	0.1867	3.525
eDIY	0.0083	0.008	0.0347	0.1812	1.575
eDSY	0.0061	0.0056	0.0369	0.1609	1.255
eISY	0.0065	0.0076	0.0358	0.1637	1.455
eDY	0.0098	0.0265	0.0337	0.1521	3.0775
eSY	0.0078	0.0293	0.0342	0.1538	3.3
eIY	0.0102	0.291	0.0315	0.1581	3.3475
eDI	0.1479	0.958	0.1342	0.5318	95.51

Bibliografía

- [1] F. Amini, H. M. Chen, G. Z. Qi, y J. C. S. Yang. *Generalized neural network based model for structural dynamic identification*. Analytical and Experimental Studies, IEEE, 1997.
- [2] M. A. Bhuiyan, R. Jaliasutram, y T. M. Taha. *Character recognition with two spiking neural network models on multicore architectures*. IEEE Symposium Computational Intelligence for Multimedia Signal and Vision Processing, 2009.
- [3] W. Bialek, F. Rieke, R. van Steveninck de Ruyter, y D. Warland. *Reading a neural code*. International Conference on Neural Networks-Conference Proceedings, 1991.
- [4] S. M. Bohte. *The evidence for neural information processing with precise spike-times: A survey*. Natural Computing, vol. 3, pp. 195-206, 2004.
- [5] S. M. Bohte, La Poutre H, y J. N. Kok. *Unsupervised clustering with spiking neurons by sparse temporal coding and multilayer RBF networks*. Neuroscience and Biobehavioral Reviews, vol. 22, pp. 355-370, 2002.
- [6] Robert L. Boylestad y Louis Nashelsky. *Electrónica teoría de circuitos*. Cuarta Edición, 1989.
- [7] S. C. Brown y K. M. Passino. *Intelligent control for an acrobot*. Journal of Intelligent and Robotic Systems, vol. 18, pp. 209-248, 1997.
- [8] S. C. Brown y K. M. Passino. *Intelligent control for an acrobot*. Journal of Intelligent and Robotic Systems, vol 18 pp. 209-248, 1997.
- [9] Carlos E. Castaneda y P. Esquivel. *Direct current motor control based on high order neural networks using stochastic estimation*. IEEE, 2010.
- [10] R. Chibani y M. Chtourou. *IMC based automatic tuning method of PID controllers using neural networks*. 6th International Multi-Conference on Systems, 2009.

-
- [11] C. C. Chung y J. Hauser. *PNonlinear control of a swinging pendulum*. Automatica, vol 31 pp. 851-862, 1995.
- [12] A. Delorme, J. Gautrais, R. Van Rullen, y S. Thorpe. *SpikeNET a simulator for modeling large networks of integrate and fire neurons*. Neurocomputing, vol. 26-27, pp. 989-996, 1999.
- [13] Gérard Dreyfus. *Neural Networks*. Springer verlag GMBH, 2005.
- [14] J. J. Eggermont. *Is There a Neural Code*. Neuroscience and Biobehavioral Reviews, vol. 22, pp. 355-370, 1998.
- [15] Jean-Jacques E.Slotine y Weioing Li. *Applied Nonlinear Control*. New Jersey, 1991.
- [16] Brian W. Evans. *Arduino notebook A beginners reference written and compiled*. Primera Edicion, 2007.
- [17] I. Fantoni y R. Lozano. *Global stabilization of the cart-pendulum system using saturation functions*. Conference on Decision and Control vol. 5, pp. 4393-4398, 2003.
- [18] I. Fantoni, R. Lozano, y M. Spong. *Energy based control of the pendubot*. IEEE Trans. on Automatic Control, vol 45, pp.725-729, 2000.
- [19] I. Fantoni, R. Lozano, y M. Spong. *Energy based control of the pendubot*. IEEE Trans. on Automatic Control, vol. 45, pp. 725-729, 2000.
- [20] R. Fierro, F. L. Lewis, y A. Lowe. *Hybrid control for a class of underactuated mechanical systems*. Transactions on Systems, Man and Cybernetics, vol. 29 pp. 649-654, 1998.
- [21] K. Furuta, M. Yamakita, y S. Kobayashi. *Swing up control of inverted pendulum using pseudo-state feedback*. Journal of System and Control Engineering, vol. 206 pp. 263-269, 1992.
- [22] Wolfram Gerstner y Werner M. Kistler. *Spiking neuron models single neurons, population, plasticity*. Cambridge University Press, 2002.
- [23] F. Gordillo y J. Aracil. *A new controller for the inverted pendulum on a cart*. International Journal of Robust and Nonlinear Control, vol 18, pp. 1607-1621, 2008.
- [24] F. Gordillo, J. Aracil, y J. A. Acosta. *Control strategy for the cart-pendulum system*. In Proceedings of Control 2004, vol. 6, pp. 214-219, 2004.

-
- [25] Rahib H. Abiyev, Okyay Kaynak, y Yesim Oniz. *Spiking neural networks for identification and control of dynamic plants*. International Conference on Advanced Intelligent Mechatronics, 2012.
- [26] Demuth Hagan y Beale. *Neural network design*. PWS Publishing Company, 1995.
- [27] J. Hauser, S. Sastry, y P. Kokotovic. *A. R. Teel. Using saturation to stabilize a class of single-input partially linear composite systems*. IFAC NOLCOS'92 Symposium, pp. 269-374, 1992.
- [28] J. Hauser, S. Sastry, y P. Kokotovic. *Nonlinear control via approximate input-output linearization: the ball and beam example*. Transactions on Automatic Control, vol 37, pp. 392-398, 1992.
- [29] J. Hauser, S. Sastry, y G. Meyer. *Nonlinear control design for slightly nonminimum phase system*. Application to v/stol aircraft. Automatica, vol 28, pp. 665-679, 1992.
- [30] A. L. Hodgkin y A. F. Huxley. *Quantitative description of membrane current and its application to conduction and excitation in nerve*. J. Physiol. (Lond.), vol. 117, pp. 500-544, 1952.
- [31] J. J. Hopfield. *Pattern recognition computation using action potential timing for stimulus representation*. Nature, vol. 376, pp. 33-36, 1995.
- [32] P. V. Kokotovic I. Kanellakopoulos y A. S. Morse. *Systematic design of adaptive controller for feedback linearizable systems*. Transaction on Automatic Control, vol. 36 pp. 1241-1253, 1991.
- [33] E. M. Izhikevich. *Simple model of spiking neurons*. Transaction Neural Networks, vol. 14, pp. 1569-1572, 2003.
- [34] Y. Jin, R. Wen, y B. Sendhoff. *Evolutionary multi-objective optimization of spiking neural networks*. Artificial Intelligence and Lecture Notes in Bioinformatics, vol. 4668, pp. 370-379, 2007.
- [35] E. R. Kandel, J. H. Schwartz, y T. M. Jessell. *Principles of neural science*. Health Professions Division New York, 2000.
- [36] Miroslav Krstic, Ioannis Kanellakopoulos, y Petar V. Kokotovic. *Nonlinear and adaptive control design*. John Wiley & Sons, Inc., 1995.

- [37] Ramos Velasco L. Enrique, Domínguez Ramírez O. Arturo, y V. Parra Vega. *Wavenet fuzzy PID controller for nonlinear MIMO systems: Experimental validation on a high-end haptic robotic interface*. Applied Soft Computing, vol. 40 pp199-205,, 2016.
- [38] Anton Labraña, Aurora Plata, y Celia Peña. *Álgebra Lineal. Resolución de sistemas lineales*. Minnesota, 1995.
- [39] Luis R. Lechuga. *Control PID SNN*. Transactions on neural networks, 2003.
- [40] Lennart Ljun. *System identification: Theory for the user*. Signal Analysis and Prediction, 1998.
- [41] Lennart Ljung y Torkel Glad. *Modeling of dynamic systems*. Dit Mosco, 1994.
- [42] R. Lozano, I. Fantoni, y D.J. Block. *Stabilization on the inverted pendulum around its homoclinic orbit*. System and Control Letter, vol 40 pp 197-204, 2000.
- [43] N. J. Aranda López, M. G. Diaz Montiel, y V. H. Galván Zavala. *Control de un sistema subactuado usando visión artificial*. Tesis de Licenciatura, UAEH, 2005.
- [44] K. Furuta M. Iwashiro y K. J. Astrom. *Energy based control of pendulum*. International Conference on Control Applications, pp. 715-720, 1996.
- [45] W. Maass. *Fast sigmoidal networks via spiking neurons*. Neural Computational., vol. 9, pp. 279-279, 1997.
- [46] W. Maass. *Networks of spiking neurons: The third generation of neural network models*. Neural Networks, vol. 10, pp. 1659-1671, 1997.
- [47] W. Maass y C. M. Bishop. *Pulsed neural networks*. Cambridge, Mass., 1999.
- [48] W. Maass, G. Schnitger, y E. D. Sontag. *On the computational power of sigmoid versus boolean threshold circuits*. Proceedings of the 32nd Annual Symposium on Foundations of Computer Science, 1991.
- [49] B. Meftah, O. Lezoray, y A. Benyettou. *Segmentation and edge detection based on spiking neural network model*. Neural Process Letters, vol. 32, pp. 131-146, 2010.
- [50] Nirode Mohanty. *Random signal estimation and identification*. Van Nostrand Reinhold Company, 1986.

- [51] Ammar Moheemmed, Satoshi Matsuda, Stefan Schliebs, Kshitij Dhoble, y Nikola Kasabov. *Optimization of spiking neural networks with dynamic synapses for spike sequence generation using PSO*. Proceedings of International Joint Conference on Neural Networks, 2011.
- [52] Ammar Moheemmed y Stefan Schliebs. *Method for training a spiking neuron to associate input-output spike train*. Engineering Applications of Neural Networks vol. 363 pp. 219-228, 2011.
- [53] Katsuhiko Ogata. *Ingeniería de Control Moderno*. Madrid, 2007.
- [54] R. Olfati-Saber. *Fixed point controllers and stabilization of the cart-pole and rotating pendulum*. the 38th Conf. on Decision and Control, vol. 2, pp. 1174-1181, 1999.
- [55] R. Olfati-Saber y A. Megretski. *Controller design for the beam-and-ball system*. IEEE Conference on Decision and Control, vol. 4, pp. 4555-4560, 1998.
- [56] Filip Ponulak. *ReSuMe - new supervised learning method for spiking neural networks*. Poznan, 2005.
- [57] Innovete Educate Quanser. *Linear Motion Servo Plants IP01*. www.quanser.com, 2000.
- [58] L.E. Ramos, B. Castillo Toledo, y S. Negrete. Nonlinear regulation of a seesaw-inverted pendulum. *International conference on control applications, vol 41, pp.1399-1403*, 1998.
- [59] K. L. Rice, M. A. Bhuiyan, T. M. Taha, C. N. Vutsinas, y M. C. Smith. *FPGA implementation of izhikevich spiking neural networks for character recognition*. International Conference on ReConFigurable Computing and FPGAs, 2009.
- [60] Benjamin Schrauwen y Jan Van Campenhout. *BSA, a Fast and accurate spike train encoding scheme*. Transactions on neural networks, 2003.
- [61] T. J. Sejnowski. *Time for a new neural code*. Nature, vol. 376, pp. 21-22, 1995.
- [62] D. Seto y J. Baillieul. *Control problems in super-articulated mechanical systems*. Transactions on Automatic Control, vol. 39, pp.2442-2453, 1994.
- [63] F. Shahrakia, M.A. Anaeib, y A.R. Arjomandzadeha. *Adaptive system control with PID neural networks*. University of Sistan and Baluchestan,, 2003.
- [64] M. W. Spong. *Control Problems in Robotics and Automation*. Chapter Underactuated mechanical systems, vol. 230, pp 135-150., 1998.

- [65] M. W. Spong y D. J. Block. *The pendubot: a mechatronic system for control research and education*. Conference on Decision and Control, vol. 1, pp 555-556, 1995.
- [66] Karl J. Åström y Tore Hägglund. *Control PID avanzado*. Pearson Educación, 2009.
- [67] A. R. Teel. *A nonlinear small gain theorem for the analysis of control system with saturation*. Transaction on Automatic. Control, vol 41, pp. 1256-1270, 1996.
- [68] A. R. Teel y L Praly. *Tools for semiglobal stabilization by partial state and output feedback*. SIAM Journal of Control and Optimization, vol. 33, pp. 1443-1488, 1995.
- [69] S. Thorpe, A. Delorme, y R. Van Rullen. *Spike-based strategies for rapid processing*. Neural Networks, vol. 14, pp.715-725, 2001.
- [70] Roberto A. Vazquez. *Training spiking neural models using cuckoo search algorithm*. Transactions on neural, 2011.
- [71] L. E. Ramos Velasco, B. Castillo Toledo, y J. Alvarez. Nonlinear regulation of a underactuated system. *International conference on robotics and automation*, vol. 49, pp, 3288-3293., 1997.
- [72] John J. Wade, Liam J. Mc Daid, Jose A. Santos, y Heather M. Sayers. *SWAT: A spiking neural network training algorithm for classification problems*. Transactions on neural networks, 2010.
- [73] K.Tanaka Wei Li y H. O. Wang. *Acrobatic control of a pendubot*. IEEE Transactions on Fuzzy Systems, vol.12 pp. 549-554, 2004.
- [74] Bogdan M. Wilamowski. *An Algorithm for fast convergence in training neural networks*. International Joint Conference. Vol. 3, 2001.
- [75] R. Xu y Umit Ozguner. *Sliding mode control of a class of underactuated systems*. Automatica, vol. 44 pp. 233-241, 2008.
- [76] Zhun Yu, Xie Ying-Bai, Jing You-Yin, y Lu Xu-Ao. *Applying neural networks to pid controllers for timedelay systems*. Proceedings of the Fifth International Conference on Machine Learning and Cybernetics, 2006.
- [77] H. Yuehua, X. Yang, N. Hang W. Lei, y X. Jiujiang W. Huirong. *Research on direct current motor PID control system based on BP neural networks*. Transactions on neural networks, 2012.

- [78] Y. Zheng y Y. Jing. *Fuzzy variable structure control for acrobot based on T-S model*. International Conference Mechatronics and Automation, vol. 2, pp. 842-846, 2005.

Apéndice A

Artículos sometidos

Spiking Neural Networks as Approximators of EEG and ECG Medical Signals

Omar Samperio-Vázquez*, J. Suárez-Cansino, and Luis E. Ramos-Velasco

Autonomous University of the State of Hidalgo
Basic Sciences and Engineering Institute
Information and Systems Technologies Research Center
Ciudad del Conocimiento, Mineral de la Reforma, Hidalgo
México, C.P. 42184
{omarsamvaz, jsuarez, lramos}@uaeh.edu.mx

Abstract

This paper proposes a novelty spiking neural network architecture, which approximates EEG and ECG medical signals with a very small error. The architecture avoids the presence of singularities along the training process by means of the application of a persistence input signal.

Keywords: LIST OF 2-4 KEYWORDS SEPARATED BY COMMAS

1 Introduction

Currently, the knowledge of the behavior and functioning of the human body is one of the main priorities in areas such as medicine and robotics. The brain and the heart are two important and vital organs to which the corresponding scientific community focus the attention for their analysis and study [1][2][3][4][5]. The beats of the heart and the internal neural interactions in the brain are transformed into electrical signals that are respectively observed by means of electrocardiogram (ECG) and electroencephalogram (EEG) devices.

For the surface electrocardiogram, two electrodes posed on their respective, and properly chosen, locations over the heart surface detect the difference of potential between the insertion points [1][2][3]. In the electrocardiographic derivations, the recorded electric potential represents the sum of the instantaneous electrical forces (vectors) that occur in a sequential form during the depolarization and repolarization processes, such as the Figure 1 shows.

*Corresponding author, Telephone: 01 771 7172000, extension 6738.

Passive electrodes located over the scalp are useful to detect the brain signals that the electroencephalogram reproduces for visual observation. This is a non invasive technique and there are several reliable devices for obtaining the measurements. For example, the Emotiv Epoc device [5] uses sixteen electrodes located over the scalp such as the Figure 2 shows. These electrodes are useful to detect diverse brain activity such as, for example, brain activity when the experimental subject is listening, paying attention or imagining movements, among others [4][5].

The EEG data for the analysis of signal approximation that this paper deals with are based on the brain activity for imagining movements, which is detected by the eight electrodes AF3, AF4, FC5, FC6, P7, P8, O1 and O2 that the Figure 2 shows, but considering the data coming from the electrode AF4, only. The data are reported by the authors of reference [5].

The reproduction of the corresponding EEC and ECG signals is very helpful to provide an interpretation, simulation or emulation of the functioning of the two organs, like usually a virtual human patient does [reference]. Although there are different techniques to approximate the signals, they usually have the inconvenience of involving the fitting of complex mathematical models by means of computationally expensive nonlinear regression methods or Fourier or Wavelets techniques which, additionally, are not able of learning.

Third generation artificial neural networks, also called spike neural networks, are defined by considering the coding and decoding of the information that one neuron respectively receive or transmit to another one, or to the environment [14]. There are different methods of encoding and decoding, and the selection of one particular procedure depends on the kind of communication and synchronization that the neurons have [reference]. In particular, this paper considers the frequency encoding method, which affirms that the information that one neuron transmits to another is just encoded by the firing frequency.

There do exist several types of mathematical models to study the neural dynamic and they are categorized in accordance to how much they resemble the behavior of the biological neuron. Similarly, the learning or training methods are usually separated in two main classes; namely, supervised and unsupervised training methods. Specifically, this paper considers the case of an artificial neuron whose behavior is given by the Perfect Integrate and Fire (PIF) model [9], while the SpikeProp algorithm defines the supervised training [8].

1.1 Problem statement

1.2 Hypotesis

2 Materials and Methods

2.1 Proposed scheme

The Figure 3 shows the main components of the proposed filter to approximate the signal y . The spiking neural network is defined by a black box, to be

explained later on, with some of the inputs given by the signal y and the absolute error e , which is computed through the difference between the desired output y and the computed neuron output \hat{y} ; in other words, $e_k = y_k - \hat{y}_k$, at the iteration k th of the training process. Particularly, the paper proposes that the input signal y be defined by the EEC or the ECG signals. Notice that an additional input exists and is represented by the constant parameter v , which is just an instantaneous pulse at time 200 ms inside the time window.

The black box in the Figure 3 is defined by a spiking neural network with a feedforward architecture containing four layers of spiking neurons, such as Figure 4 shows. The input layer is defined by the single value y , while two hidden layers are defined by just one single spiking neuron and n spiking neurons, respectively. Finally, the output layer is given by just one single spiking neuron, as well. The experimental setting in this paper considers the cases where $n = 5$ or $n = 10$, although the correct value depends on the nonlinearity of the signal to be approximated.

In the Figure 4, $s_i^{(1)}$ represents the i th spiking neuron at the first hidden layer of the SNN, $1 \leq i \leq n_1$, $s_j^{(2)}$ represents the j th spiking neuron at the second hidden layer of the same SNN, $1 \leq j \leq n_2$, and $s_r^{(3)}$ denotes the r th neuron at the output layer. Of course, $y = s_1^{(0)}$ and $v = s_2^{(0)}$ are the components of the SNN input. Remember that

$$v(t) = \begin{cases} \varepsilon & \text{if } t = t_0 \\ 0 & \text{otherwise} \end{cases}$$

where ε is a positive real number and t_0 is an instant of time within the time window. So that the input signal is defined by a bidimensional vector in the space of stimulus.

The PIF model is given by the equations related to a capacitive circuit, which is illustrated by the Figure 5. The deduction of the PIF model considers the measurement of the potential $V(t)$ at time t , assuming some considerations about the times of opening and closing the switches in the figure.

The model is given by the following simple ordinary linear, non homogeneous and first order differential equation

$$C \frac{dV(t)}{dt} = I(t)$$

whose solution is given by the equation

$$V(t) = V_r + \frac{1}{C} \int_{t_0}^t I(t) dt$$

where C is the capacitance of the condenser, V_r is the resting potential, t_0 and t respectively are the initial and current times of observation, and $I(t)$ is an external electrical input to the circuit. The experimental setting in this paper assumes that $V_r = -65\text{mV}$, $C = 50\mu\text{F}$. Then, the training process is made

under these set of experimental premissas.

2.2 Spike–Prop learning method

Bohte *et al.* propose a supervised learning algorithm based on the orthodox backpropagation training method of the feedforward architecture, in the second generation of artificial neural networks [10]. The method is called SpikeProp, and its definition considers a feedforward artificial neural network architecture with spiking neurons making multiple connections, whose signal transmission in every connection is also made with delays. The Figure 6 illustrates the general topology of the architecture.

The adaptation rule of the weights in a connection between two neurons of the SNN is given by the following training equation

$$\Delta\omega_{ij}^k = -\eta \frac{\partial e}{\partial \omega_{ij}^k}$$

where i denotes to the presynaptic neuron, j to the postsynaptic neuron and k denotes the k th weight in the set of connections from neuron i to neuron j , the parameter η is the learning rate and e is the absolute error already defined in the Section 2.1.

Since the absolute error depends on ...

$$\Delta\omega_{ij}^k = -\eta \frac{y_i^k(t_j)e}{\sum_{i,l} \omega_{ij}^l \left(\frac{\partial y_j^l(t_j)}{\partial t_i} \right)}$$

where $\Delta\omega_{ij}^k$ provides the weight correction to get the optimal values, such that \hat{y} matches the desired output y . The experimental setting in this paper assumes that $\eta = 1.5$. This value is obtained in an heuristic form based on the best performance of the training process and the fast convergence rate.

3 Results

3.1 EEC signals experimentation

This part of the experiment considers the reported data coming from the reference [reference], whose authors used the Emotiv Epoc device as a measurement tool. The data is approximated by means of SNN under two different experimental conditions defined by the number of used neurons in the hidden layer, each. The first condition uses an SNN architecture with ten spiking neurons in the hidden layer ($n = 10$), while the second condition uses an architecture with five spiking neurons in the hidden layer ($n = 5$).

The Figure 7 shows how the function defined by the SNN approximates the real EEC signal. The experimental results correspond to the case where $n = 10$ at the hidden layer, and the Figure 8 remarks the fact that the error becomes

zero in a time span less than 100 sampling, which is the required time by the Emotiv Epoc device for providing the EEC data, assuming that the sample frequency is about one sample per milisecond. Since the data source [reference] does not provide the sample time, then the Figure 7 and the Figure 8 show only the sampling number over the time domain, meaning that for the m th sample a time span of $m\Delta$ has occurred, where Δ is the unknown sample time.

On the other hand, the Figure 9 illustrates the function approximation in the case where the hidden layer has $n = 5$ spiking neurons. The result of the experiment shows that the number of neurons in the hidden layer has an effect on the network's response. In this particular case, the effect is negative when compared with the previous experimental result where the number of spiking neurons in the hidden layer is $n = 10$. This is so because the error becomes zero until the 1,500 sampling, such as the Figure 10 shows.

The normality test of the error based on the One Sample Kolmogorov–Smirnov test shows that there are not systematic errors, which means that the error has a standard normal distribution (zero mean and standard deviation equal to one) with a 5% of significance level.

3.2 ECG signals experimentation

The second part of the experiment considers an ECG signal with a sampling entropy value of 1.222, which makes it a time series more complex than that given by the ECC data of the Section 3.1, with a sampling entropy value of 0.9. Again, the experimental results show that the SNN, with $n = 10$ spiking neurons at the hidden layer, produces an acceptable approximation with an error that becomes close to zero at the sampling 1,500. The Figure 11 and the Figure 12 illustrate the situation.

Finally, the Figure 13 shows the SNN approximation with $n = 5$ spiking neurons at the hidden layer, while the Figure 14 shows the behavior of the error. The error becomes close to zero at the sampling 2,500, which means that the SNN training process converges more slowly than the case $n = 10$.

The normality test of the error based on the One Sample Kolmogorov–Smirnov test shows that there are not systematic errors, which means that in these cases the error has a standard normal distribution (zero mean and standard deviation equal to one) with a 5% of significance level, as well.

4 Conclusion

1. Bob–Jenkins approximation and other methods
2. Confiability of the experiments
3. Dynamics Adaline
4. The number of layers and neurons per layer

5. Test and error method to determine the correct number of neurons per layer
6. Sample size

5 Acknowledgements

The author Omar Samperio-Vázquez likes to thank the Science and Technology National Council (CONACyT) of Mexico, for the financial support along the whole period of time required to get the master of science degree, it is greatly appreciated.

References

- [1] Burdon Sanderson J. (1878). *Experimental results relating to the rhythmical and excitatory motions of the ventricle of the frog heart*. Proc Roy Soc Lond vol. 27, pp. 410-14
- [2] Waller AD. (1887). *A demonstration on man of electromotive changes accompanying the heart's beat*. J Physiol (Lond) vol. 8, pp. 229-34
- [3] Klabunde, R.E.. (2005). *Electrical activity of the heart*. Cardiovascular physiology concepts. Lippincott Williams and Wilkins. ISBN 0-7817-5030-X.
- [4] Olejniczak P. (2006). *The ten-twenty electrode system of the International Federation*. Electroencephalogram Clin Neurophysiologic; vol. 10 pp. 371-375
- [5] Jasper HH. (1958). *Neurophysiological Basis of EEG* Clinical Neurophysiology; vol. 23, pp. 186-189
- [6] Stytsenko K, Jablonskis E, Prahm C. (2011). *Evaluation of consumer EEG device Emotiv EPOC*. Proceedings of the MEi, CogSci Conference OpenURL
- [7] G. Pozas Garza. (2010). *Implementacin de una tecnica estandar para la adquisicin del electrocardiograma*. Avances; vol. 20: pp. 52-56.
- [8] Joost N. Kok Sander M. Bohte and Han La Poutre.(2002)*Errorbackpropagation in temporally encoded networks of spiking neurons*. *Neurocomputing*, Elsevier, vol.48, pp. 17-37
- [9] Werner M. Kistler Wulfram Gestner.(2002) *Spiking neuron Models Single Neurons,Population, Plasticity*. Cambridge University Press
- [10] L. Dingding S. McKennoch and L. G. Bushnell.(2006) *Fast modifications of the Spike-Prop algorithm*. IEEE International Conference on Neural Networks-Conference Proceedings, Vancouver, BC

- [11] Levin A. y K. Narendra. (1993) *Control of Nonlinear Dynamical Systems Using Neural Networks - Part II: Observability, Identification and Control*. IEEE Transactions on Neural Networks vol. 7, pp. 30-42.
- [12] Samanwoy Ghosh-Dastidar and Hojjat Adeli(2009).*Spiking Neural Networks*. International Journal of Neural Systems. vol.19 ISSUE 04.
- [13] L.R. Lechuga.(2014).*Control Inteligente de Mquinas Elctricas Emplorado Redes Neuronales de Tercera Generacin*. Tesis de Grado de Maestra en Automatizacin Y Control, UAEH.
- [14] Wolfgang Mass(1996).*Networks Spiking Neuron: The Third Generation of Neural Networks Models*. International Journal of Neural Networks. vol.10 No. 9 pp1659-1671.
- [15] Nicolangelo Iannella et al (2001).*A spiking neural networks architecture for nonlinear function approximation*. International Journal of Neural Networks. vol.14 No. 9 pp 933-1939.

6 Tables

7 Figure captions

Figure 1: Electrocardiogram

Figure 2: Sixteen electrodes of the EMOTIV EPOC are posed over the scalp [5]

Figure 3: Proposed scheme for the approximation of medical signals

Figure 4: General scheme of an SNN with four layers

Figure 5: Capacitive electrical circuit from where the PIF model arises

Figure 6: Weights representation in the training process of the proposed SNN [13]

Figure 7: EEC signal approximation by means of the SNN with ten neurons at the hidden layer

Figure 8: Approximation error to the EEC signal by means of the SNN with ten neurons at the hidden layer

Figure 9: EEC signal approximation by means of the SNN with five neurons at the hidden layer

Figure 10: Approximation error to the EEC signa by means of the SNN with five neurons at the hidden layer

Figure 11: ECG signal approximation by means of the SNN with ten neurons at the hidden layer

Figure 12: Approximation error to the ECG signal by means of the SNN with ten neurons at the hidden layer

Figure 13: ECG signal approximation by means of the SNN with five neurons at the hidden layer

Figure 14: Approximation error to the ECG signal by means of the SNN with five neurons at the hidden layer

8 Figures

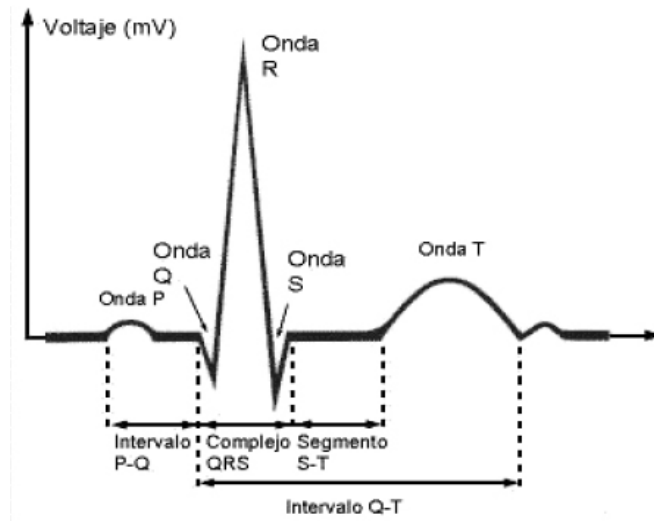


Figure 1:

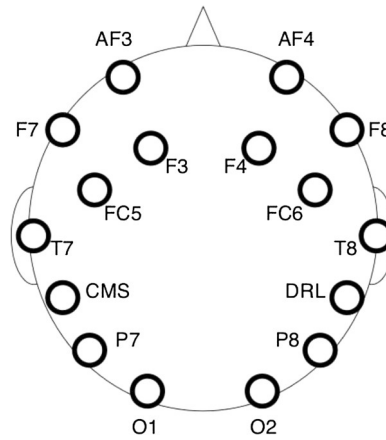


Figure 2:

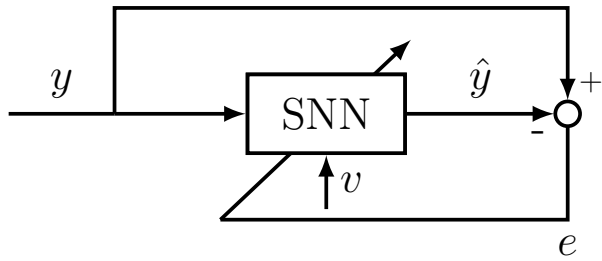


Figure 3:

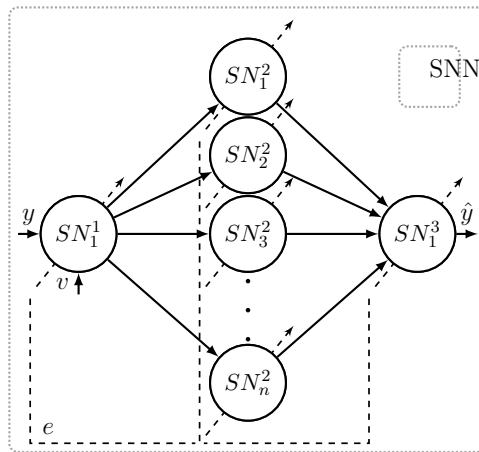


Figure 4:

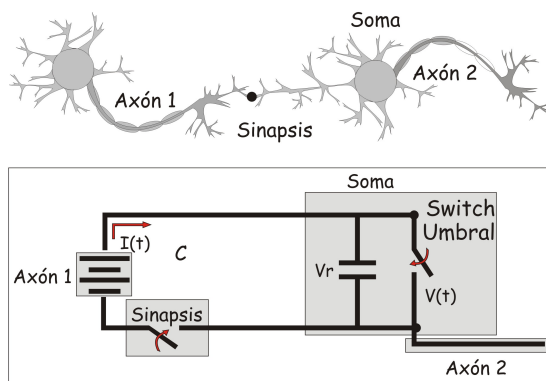


Figure 5:

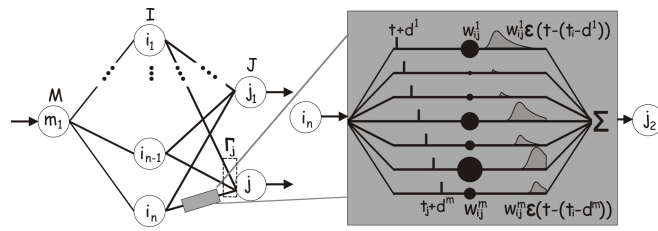


Figure 6:

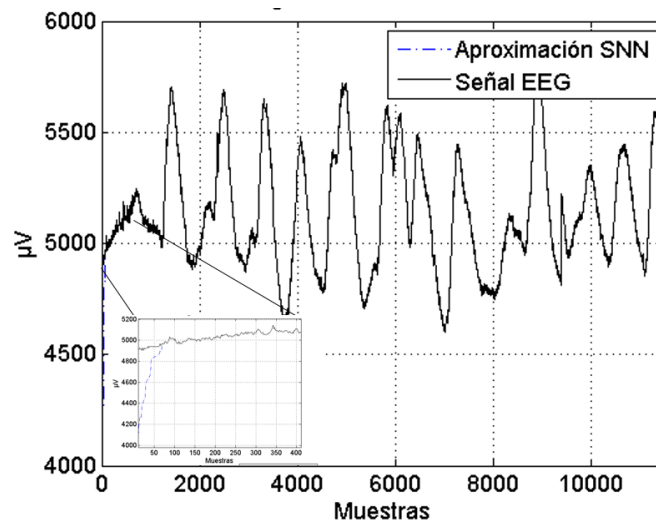


Figure 7:

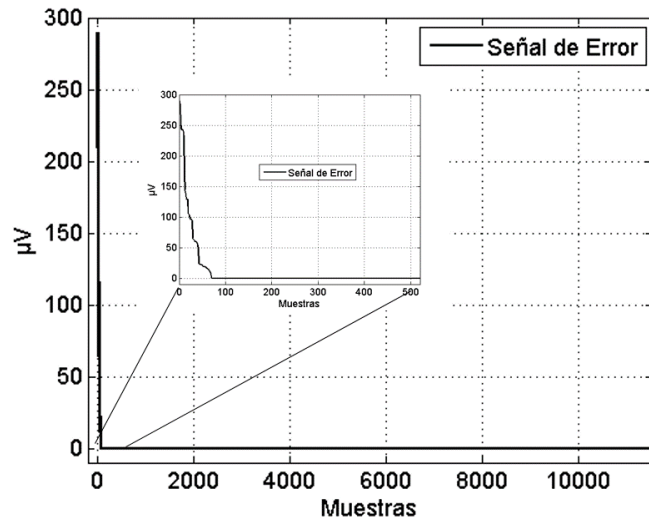


Figure 8:

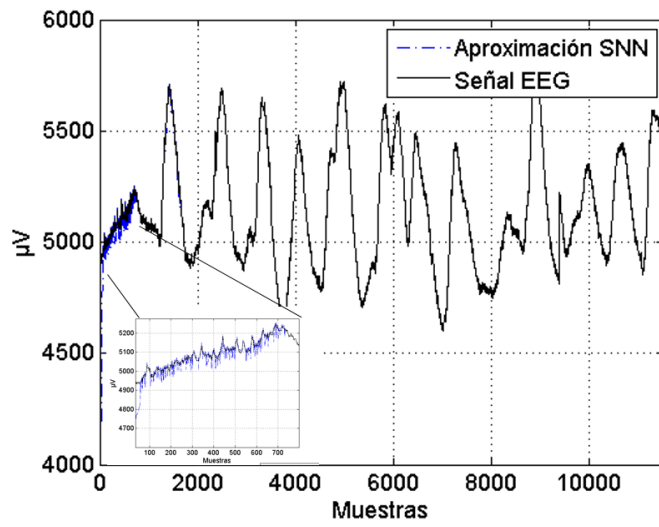


Figure 9:

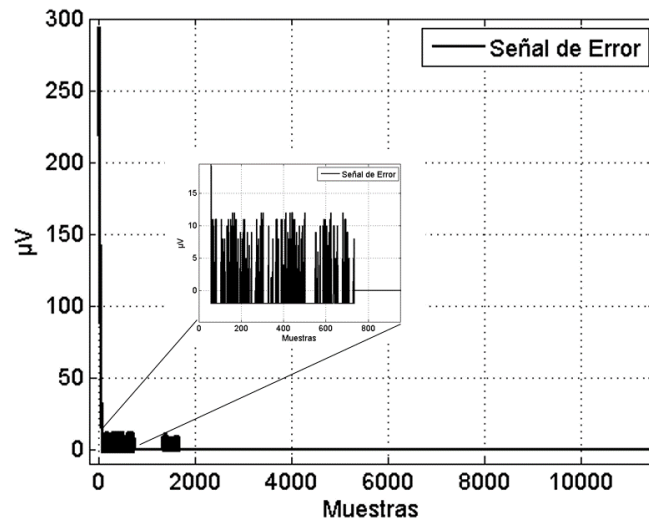


Figure 10:

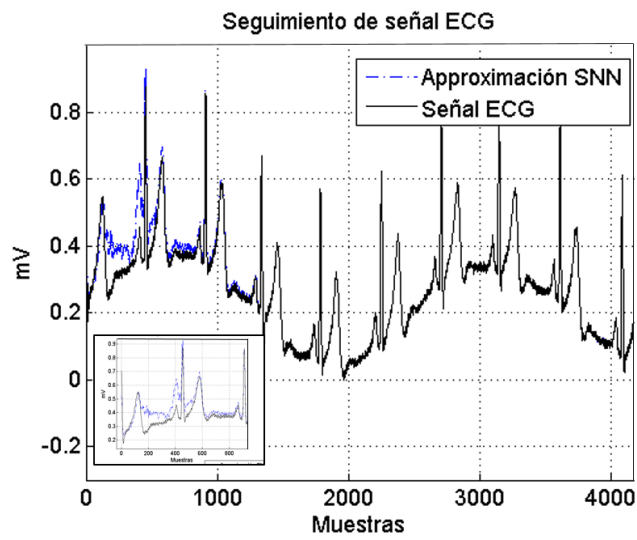


Figure 11:

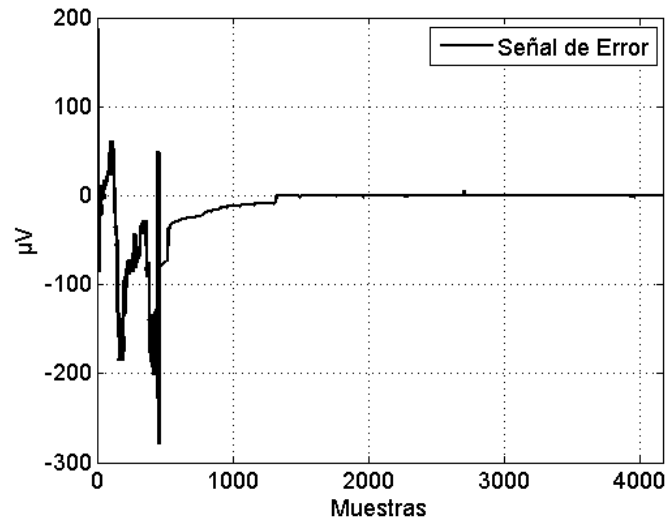


Figure 12:

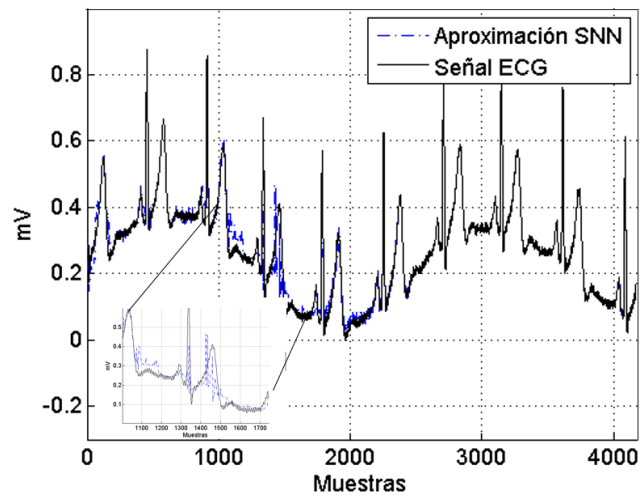


Figure 13:

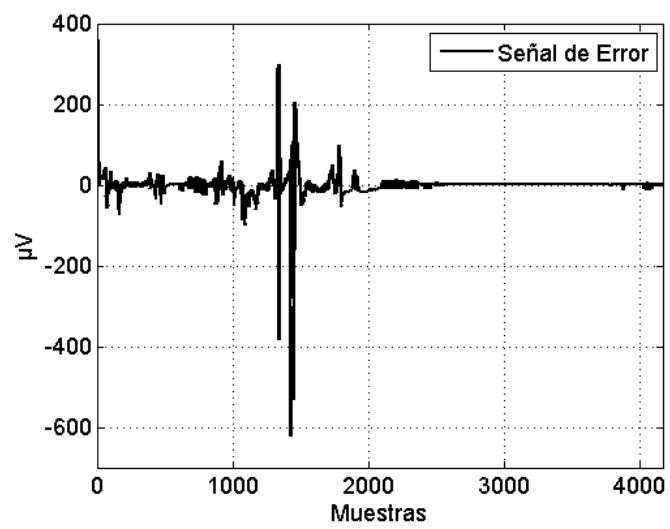


Figure 14:

Apéndice B

Algoritmos programados en MATLAB

El objetivo de este apéndice es de mostrar los algoritmos empleados para la implementación de la simulación del control de la planta carro péndulo invertido, programados en MATLAB 2013a.

B.1. MATLAB

B.1.1. Programa principal

```
1  Programa 3; Control de p\'eldulo con multicapa, 1 neuronas de  
   entrada 3 de capa oculta y 1 de salida  
2  As neuronas de la capa uno y dos solo tienen una sola entrada,  
3  clc  
4  clear all  
5  close all  
6  %Definición de parametros generales  
7  NoEntrna=100;  
8  n1=.1; %0  
9  n2=.01; %001  
10 n3=.001;  
11 x0 = [0.05;1; 0;0]; %condicion inicial en theta radianes, en X  
   unidades  
12 tSpan=[0,NoEntrna];  
13 L=.64;  
14  
15 for i=1:1  
16 tiN=0; %tiempo de inicio
```

```

17 tfN=1000; % tiempo final de la neurona
18 AN=1; % Factor de incremento de muestreo
19 pulso=1; % amplitud del pulso en milivolts
20 Wft=0;
21 Vr=-60; % voltaje de relajación
22 tN=tiN:AN:tfN;
23 ft=zeros(1,length(tN));
24 % ***** Parametros del pendulo *****
25 l = .32; % Longitud del Ándulo
26 m = .23; % Masa del pÁndulo
27 M=.52; % Masa del Carro
28 % Momento de inercia referido al eje
29 j=.007;
30 J=j; % momento de inercia de la barra
31 B = 1; % Coeficiente de fricción
32 g = 9.81; % Aceleración de la gravedad
33 Tm=.01;
34 ent1=1;
35 % Parametros Comportamiento Neurona
36 C=20; aI=.02;
37 R=50; bI=.2;
38 umbral=-30; cI=Vr;
39 dI=2;
40 % Parametros función Pulso
41 ts=30; % Tiempo de establecimiento del pulso
42 vp=1; % factor de escalamiento de la función pulso
43
44 % Matriz de Defasamiento
45 d=40; % tiempo tao
46 k=4; % numero de delays
47
48 end % Parametros
49 for i=1:1
50 spike31=500;
51 spike21=500;
52 spike22=500;
53 spike11=500;
54
55 E1=500;
56 u=0;
57 yx=0;
58 ecac1=0;

```

```

59 ecac2=0;
60 Ic1=500;
61 Ic2=500;
62 Dc1=500;
63 Dc2=500;
64 ystx=500;
65 ystc=500;
66 e=0;
67 e1=0;
68 e2=0;
69 no1=1; %numero de entradas de las neuronas de la primer capa
70 no2=1;
71 no3=1; % numero de entradas de la neurona de salida
72 no4=2;
73 %<<<<CAPA DE ENTRADA 11 >>>>>>
74 In1=zeros(no1,length(tN));
75     In1(1,500)=pulso;
76     In1(2,500)=pulso;
77     In1(3,500)=pulso; %<<<<<<CAPA OCULTA 21>>>>
78         In2=zeros(no2,length(tN));
79             In2(1,500)=pulso;
80
81         %<<<<<<CAPA OCULTA 22
82             In3=zeros(no3,length(tN));
83                 In3(1,500)=pulso;
84
85                 %<<CAPA SALIDA>>>>
86                     In4=zeros(no4,length(tN));
87                         In4(1,500)=pulso;
88                             In4(2,500)=pulso;
89
90 %
91 [INDM1 wdM1] = SpikePropM (In1,d,k,pulso); %inicializaci3n de
           pesos neurona 1
92 W1=ones(1,length(wdM1))*0.3875;
93 Aw1=zeros(1,length(wdM1));
94 [INDM2 wdM2] = SpikePropM (In2,d,k,pulso); %inicializacion de
           pesos neurona 2
95 W2=ones(1,length(wdM2))*1.161;
96 Aw2=zeros(1,length(wdM2));
97 [INDM3 wdM3] = SpikePropM (In3,d,k,pulso); %inicializacion de
           pesos neurona 2

```



```

210  exixdot=ex(end,4);
211  tiempo=et;
212  THETA(ent1)=theta;
213  EXIS(ent1)=exis;
214  tiem(ent1)=ent1;
215  THETADOT(ent1)=thetadot;
216  EXIXDOT(ent1)=exixdot;
217  x0=[exis; theta; thetadot; exixdot];
218  % % Ecuación de salida
219  %
220  %
221      yst=SNsig*theta;
222      ysx=SNsig*exis;
223  % end
224  %
225  %
226  %          ystc=floor(1000*(yst+8)/16); %salida theta
codificada a pulsos
227  %          ystx=floor(1000*(ysx+8)/16); %salida exis
codificada a pulsos
228  sp;
229      ytx2=1*ysx+.64*yst;          %calculo del error exix
230      e=sp-ytx2
231      ea=e; %calculo del error theta
232      E1=floor(1000*(ea+20)/40) %error uno codificado
233      IE1=ecac1+ea; %integral del
error
234      Ic1=floor(1000*(IE1+8)/16) ; %integral del error
codificado
235      De1=(ecac1-ea)/ ent ; %Derivada del
error
236      Dc1=floor(1000*(De1+50)/100); %Derivada del error
codificado
237      cac1=ea;
238      if abs(e)>1
239          n1=.7;
240      elseif abs(e)<=1&&abs(e)>.01
241          n1=0.5;
242      else
243          n1=0.2;
244      end
245      n2=n1/2;

```



```

288 toc
289 figure
290 % plot(entV, e1v, 'b', entV, n1v, 'r')
291 plot(entV, e1v, 'b')
292 grid on
293 title( 'Error_entre_la_referencia_y_la_posiciÃ³n_del_carro', '
    FontSize',13)
294 xlabel( 'Tiempo/segundos', 'FontSize',13)
295 ylabel( 'Metros', 'FontSize',13)
296 legend( 'error' )
297 % %
298 figure
299 plot(entV, control, 'b')
300 grid on
301 title( 'Control_carro_pÃ©ndulo', 'FontSize',13)
302 xlabel( 'Tiempo/segundos', 'FontSize',13)
303 ylabel( 'Voltage', 'FontSize',13)
304 legend( 'Control' )
305
306 figure
307 %subplot(2,1,1)
308 plot(entV, SP2V, 'b', entV, theta1, 'r')
309 grid on
310 title( 'PosiciÃ³n_del_pendulo_y_del_carro', 'FontSize',11)
311 %axis([0 200 -6 1])
312 %xlabel('Number of training', 'FontSize',13)
313 ylabel( 'Radianes', 'FontSize',13)
314 legend( 'Referencia', 'Theta' )
315 xlabel( 'Tiempo/segundos', 'FontSize',13)
316 figure
317 %subplot(2,1,2)
318 plot(entV, SPV, 'b', entV, exis1, 'r')
319 grid on
320 title( 'Response of the system with different inputs', 'FontSize
    ',13)
321 %axis([0 NoEntrna -2 2])
322 xlabel( 'Tiempo/segundos', 'FontSize',13)
323 ylabel( 'Metros', 'FontSize',13)
324 legend( 'Referencia', 'PosiciÃ³n_del_carro' )

```

B.1.2. Etapa de entrenamiento

```

function [INDM wdM] = SpikePropM (In01 ,d ,k1 , pulso)
2   nmIn01= size (In01) ;
3   for n=1:nmIn01(1)
4       Md=1;
5       suma=0;
6       for m=1:nmIn01(2)
7           Nd=1;
8           suma=In01(n,m)+suma;
9           if In01(n,m)==pulso
10              for nd=-k1:k1 % aki
11                  md=m+(nd*d);
12                  if (md>=1 and md<=nmIn01(2))
13                      if (Nd==1 and Md==1)
14                          Ind(Nd,:)=zeros(1, nmIn01(2));
15                              end
16                              Ind(Nd,md)=pulso;
17                              Nd=Nd+1;
18                              Md=Nd+1;
19                          end
20                      end
21                  end
22              end
23              if suma==0;
24                  Ind01{n}=zeros(1, nmIn01(2));
25                  nmInd=size(Ind01{n});
26                  wd{n}=zeros(1, nmInd(1));
27                  clear Ind
28              else
29                  Ind01{n}=Ind;
30                  nmInd=size(Ind); % Ind01
31                  wd{n}=ones(1, nmInd(1));
32                  clear Ind
33              end
34          end
35
36      INDM=[];
37      wdM=[];
38      for i=1:length(Ind01)
39          INDM=[INDM; Ind01{i}];
40          wdM=[wdM wd{i}];

```

```

41         end
42     end

```

B.1.3. Sumatorias

```

1
2
3 function [SumaIn SumaInM SumaInD]=InNeuro(ti,tf,A,ft,In01,pulso,
4     W01,Wft,ts)
5     %           SumaInM_w SumaInD
6     %           SumatV SumaRD
7
8     t=ti:A:tf;
9     nm01=size(In01); % [1 1001]
10    nmft=size(ft);
11    k=1;
12    %           ts=50; % 5
13    vp=1;
14    SumaR=0;
15    SumaRD=0;
16    Sumaw=0;
17    for i=1:nm01(1)
18        for j=1:nm01(2)
19            if In01(i,j)==pulso
20                [Fnc FncD]=alpha(t(j),tf,A,ts,vp);
21                suma{k}=Fnc*W01(i);
22                sumaw{k}=Fnc;
23                sumaD{k}=FncD*W01(i);
24                k=k+1;
25            end
26        end
27    end
28    if k>1
29        for i=1:k-1
30            Msuma(i,:)= [zeros(1,(length(t)-length(suma{i}))) , suma{i}];
31            Msumaw(i,:)= [zeros(1,(length(t)-length(sumaw{i}))) , sumaw{i}];
32            MsumaD(i,:)= [zeros(1,(length(t)-length(sumaD{i}))) , sumaD{i}];
33        end

```

```

33     else
34         Msuma=zeros(1,length(t));
35         Msumaw=zeros(1,length(t));
36         MsumaD=zeros(1,length(t));
37     end
38     for i=1:nmft(1)
39         ft(i,:)=Wft(i)*ft(i,:);
40     end
41     SumaInM=Msuma;
42     SumaInMw=Msumaw;
43     SumaInMD=MsumaD;
44     Msuma=[Msuma;ft];
45
46     for i=1:length(t)
47         SUMAT=0;
48         SUMATD=0;
49         SUMATW=0;
50         for j=1:length(Msuma(:,1))
51             SUMAT=Msuma(j,i)+SUMAT;
52             SumaIn(i)=SUMAT;
53         end
54         for j=1:length(Msumaw(:,1))
55             SUMATW=Msumaw(j,i)+SUMATW;
56             SumaInw(i)=SUMATW;
57         end
58         for j=1:length(MsumaD(:,1))
59             SUMATD=MsumaD(j,i)+SUMATD;
60             SumaInD(i)=SUMATD;
61         end
62         SumaR=SumaR+SUMAT;
63         Sumaw=Sumaw+SUMATW;
64         SumaRD=SumaRD+SUMATD;
65     end
66     for j=1:length(SumaInM(:,1))
67         SUMATV=0;
68         SUMATVW=0;
69         for i=1:length(t)
70             SUMATV=SumaInM(j,i)+SUMATV;
71             SUMATVW=SumaInMw(j,i)+SUMATVW;
72         end
73         SumatV(j)=SUMATV;
74         SumatVW(j)=SUMATVW;

```



```

75         end
76     end

```

B.1.4. Modelo Izhikevich

```

function [V S] = Izhikevich(a,b,c,d,I,Vr,Tf,A,umbral,pulso)
2 % c=Vr;
3 % d=18;
4 % u=-2;
5 % v=(Vr*81.62)/60;
a1=0;
v=Vr;
8 % a=.2;
9 % b=.02;
10 for i=1:Tf/A
11     v=v+.1*(0.04*v^2+5*v+154-u+I(i));
12     u=u+.1*(a*(b*v-u));
13     if (v>=umbral);
14         v=c;
15         u=u+d;
16         S(i)=pulso;
17     else
18         S(i)=Vr;
19     end
20     V(i)=v;
21 end
22 end

```

B.1.5. Modelo de integración

```

1
function [S,V] = NEUROLIF(Ti,Tf,A,umbral,pulso,Vr,C,I,R)
3 umbral=umbral-Vr;
4 Tm=R*C-Vr;
5 As=-1/Tm;
6 Bs=1/C;
7 H=((exp(As*A)-1)/As)*Bs;
8 G=exp(As*A);
9 V(1)=0;
10 S(1)=Vr;

```

```

11     for i=2:(Tf/A)
12         V(i)=G*V(i-1)+H*I(i-1);
13         if V(i)>= umbral
14             S(i)=pulso;
15             V(i)=0;
16         else
17             S(i)=Vr;
18         end
19     end
20     V=V+Vr;
21 end

```

B.1.6. Modelo Disparo Perfecto

```

1 function [V01,V] = NEUROPIF(Ti,Tf,A,umbral,pulso,Vr,C,I)
2 integral=0;
3 t=Ti:A:Tf;
4     for i=1:(Tf/A)
5         integral=integral+(((I(i+1)-I(i))*A/2)+I(i)*A);
6         V(i)=Vr+(1/C)*integral;
7         if V(i)>= umbral
8             integral=0;
9             V01(i)=pulso;
10            V(i)=pulso;
11        else
12            V01(i)=Vr;
13        end
14    end
15 end

```

Apéndice C

Modelo carro-péndulo invertido

C.1. Sistemas no lineales subactuados

Los sistemas subactuados están compuestos por una amplia gama de sistemas mecánicos. Por ejemplo, robots móviles no holonómicos, robots bípedos, vehículos submarinos, manipuladores con estructura flexible, misiles, satélites de comunicaciones, etc. [64]. El término subactuado se refiere a que no todas las uniones o grados de libertad (GL) del sistema tienen un actuador o son directamente controladas. Por lo tanto, los investigadores han puesto bastante atención a los problemas de control asociados con sistemas subactuados y han propuesto diversas estrategias de control para resolverlo tales como, control por back-stepping [62], control basado en energía o pasividad [18] ó [42], control adaptable [32], control difuso[73] o inteligente [7], control híbrido [20], control por modos deslizantes[75], entre otras.

En la práctica, un pequeño conjunto de sistemas subactuados ha cobrado gran popularidad por el desafío que representan, para los que existe una amplia gama de controladores en la literatura. Evidentemente, hay una infinidad de ejemplos inexplorados, tantos como queramos idear, para lo cual las técnicas existentes se muestran insuficientes. Algunas investigaciones efectuadas para solucionar problemas relacionados con el control de sistemas subactuados que han tenido un impacto considerable en las áreas educativas y de investigación.

Como primer ejemplo, podemos mencionar que M. Spong [64] presenta una excelente introducción al sistema Acrobot 8 utilizando la linelización parcial por retroalimentación. Otro trabajo interesante para este sistema es el propuesto por Yan Zheng et al. [78], el cual desarrolla una ley de control difusa de estructura variable que utiliza la teoría de Lyapunov para garantizar estabilidad asintótica global. También [8] presenta un esquema de control inteligente para resolver el problema de trasladar de la posición colgante hasta la posición invertida del Acrobot, así como el de mantener el sistema en la posición inestable cuando inicia muy cerca de esta posición.

Para un sistema similar al anterior denominado Pendubot, [65] presenta el diseño y control del sistema usando conceptos tales como linealización parcial por retroalimentación dinámica cero y se comenta su uso con fines educativos. En [58] y [71] se propone un esquema de regulación de salida no lineal para sistemas subactuados. Fantoni et. al.[19] proponen un algoritmo de estabilización considerando las propiedades de pasividad del pendubot y utilizan métodos basados en energía para diseñar la ley de control. Además, el análisis de convergencia se realiza por medio de la Teoría de Lyapunov.

Un sistema subactuado muy interesante es el Péndulo de Furuta (péndulo rotacional) desarrollado por K. Furuta en el Instituto Tecnológico de Tokio. Algunos trabajos relevantes sobre el control de este sistema se comentan enseguida. En 1992, Furuta et. al. [21] propusieron un controlador que utiliza el método de control por retroalimentación del pseudo-estado. En 1996, Iwashiro et. al. [44] consideraron el uso de métodos basados en energía para controlar el sistema. En 1999, Olfati-Saber [54] propuso una estabilización semi-global para el péndulo invertido rotacional usando controladores de punto fijo.

Un sistema que ha atraído la atención de los investigadores es el sistema viga-bola[55] para el cual existen diversos trabajos en donde se proponen estrategias como el método por retroalimentación aproximada desarrollado por Hauser et al.[28], saturaciones anidadas pequeñas [27]193] y estabilización por retroalimentación de salida [68].

El sistema de aterrizaje y despegue de un avión (PVTOL) [29] es otro ejemplo de sistema subactuado y diversas metodologías para controlar este sistema se pueden encontrar en la literatura. En 1996, Teel [67] ilustra los resultados del teorema de pequeña ganancia aplicados al sistema PVTOL.

C.1.1. Sistema no lineal subactuado tipo carro péndulo invertido

Los sistemas tipo péndulo invertido (SPI) están formados por un péndulo en el cual uno de sus extremos está unido mediante una articulación a una base móvil lo que le permite girar libremente y, debido a que la aceleración angular del péndulo no puede ser controlada, estos sistemas son considerados ejemplos de sistemas subactuados.

Existen básicamente dos problemas relacionados con el control de péndulos invertidos que se han estudiado. El primero, es el de llevar el péndulo desde cualquier posición y en particular de la posición colgante natural hasta la posición invertida. A este problema se le conoce como swing up. El segundo problema consiste en estabilizar al péndulo alrededor del punto de equilibrio inestable.

C.1.1.1. Carro-péndulo invertido

Este sistema es uno de los más populares para diseño de leyes de control no lineal en los laboratorios y se ve reflejado en la gran cantidad de artículos publicados en la literatura [71], [58], [11], [23], [17], [24], y su estudio está motivado por las diversas aplicaciones a el control de posición de los sistemas de despegue de cohetes y el control de vibraciones mecánicas en construcciones.

Este dispositivo consiste en un péndulo con libertad para rotar montado encima de un carro que se mueve a lo largo de una vía debido al impulso de una fuerza [53], como se muestra en la Figura C.1.

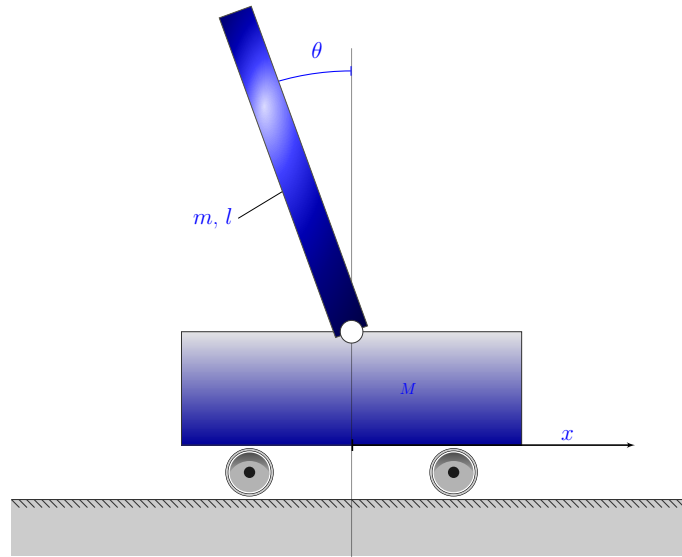


Figura C.1: Sistema mecánico subactuado tipo carro péndulo.

C.1.1.2. Ecuaciones dinámicas del sistema carro-péndulo invertido

Las ecuaciones dinámicas que constituyen el modelo matemático de un sistema pueden ser determinadas a partir de la ecuación de Euler-Lagrange, la cual está basada en la variación de la energía potencial y la energía cinética del sistema.

La ecuación de Euler-Lagrange está definida por la siguiente expresión [39]

$$\frac{d}{dt} \left[\frac{\partial}{\partial \dot{q}} L \right] - \frac{\partial}{\partial q} L = \tau \quad (\text{C.1})$$

donde q es el vector de coordenadas generalizadas, L denota el Lagrangiano definido como $L = \sum_{i=0}^n (K_i - V_i)$, con K como la energía cinética, V como la energía potencial y n el número de grados de libertad y τ es cualquier momento aplicado al sistema tal como el torque de un motor, una fricción o una fuerza exterior. Para el caso de estudio que se desarrolla en esta tesis, τ representa un torque en el eje del motor.

Para el sistema carro - péndulo, la energía cinética está dada por $K = K_c + K_p$, donde K_c es la energía cinética del carro cuyo diagrama de cuerpo libre se muestra en la Figura C.2 y se determina como:

$$K_c = \frac{1}{2}M\dot{x}^2 \quad (C.2)$$

y K_p es la energía cinética del péndulo cuyo diagrama de cuerpo libre se presenta en la Figura C.3 y se determina por

$$K_p = \frac{1}{2}m(\dot{x}^2 + 2l\dot{\theta}\dot{x}\cos\theta + l^2\dot{\theta}^2) + \frac{1}{2}J\dot{\theta}^2 \quad (C.3)$$

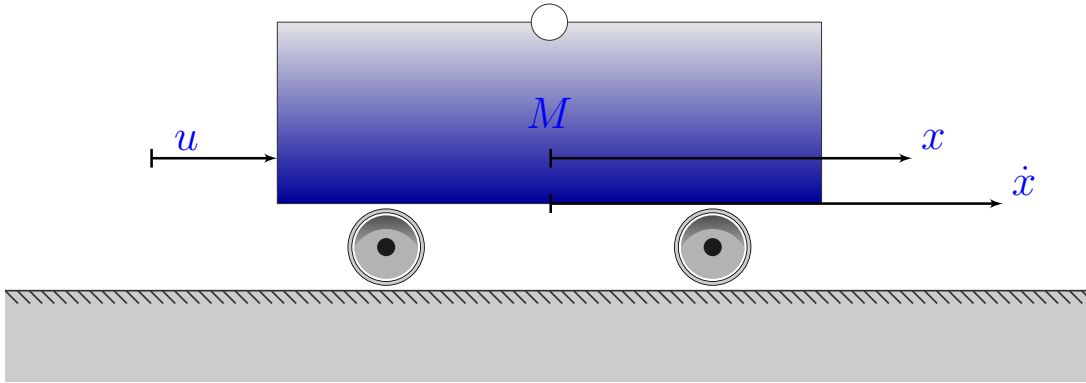


Figura C.2: Diagrama de cuerpo libre correspondiente al carro.

La energía potencial o de posición de un cuerpo de masa m colocado a una altura h está determinada por:

$$V = mgh \quad (C.4)$$

Para el sistema carro - péndulo la energía potencial está dada por la suma de la energía potencial del carro más la energía potencial del péndulo, es decir $V = V_c + V_p$.

De acuerdo con (C.4) la energía potencial para el carro, es cero ya que a lo largo de su trayectoria horizontal su altura no cambia. En caso contrario, el péndulo (Figura C.3)

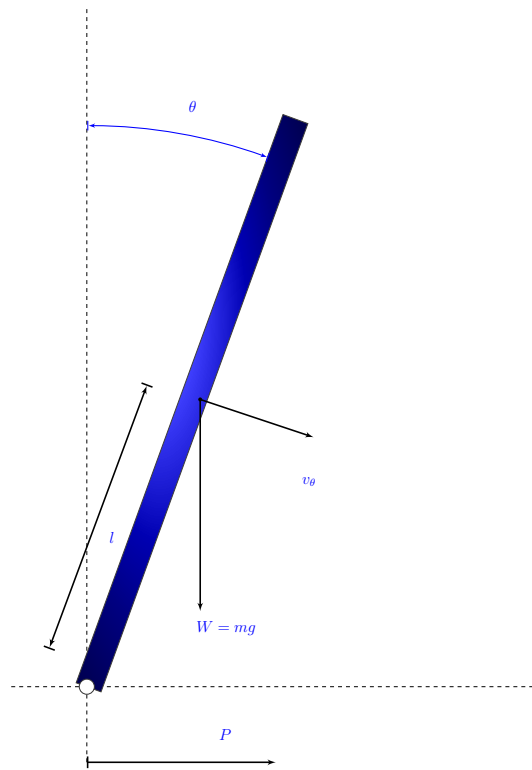


Figura C.3: Diagrama de cuerpo libre correspondiente al péndulo.

tiene una energía que depende de la altura de su centro de masa ($l\cos\theta$), entonces la energía potencial del péndulo está dada por:

$$V_p = mgl \cos \theta \quad (\text{C.5})$$

El Lagrangiano del sistema L está dado por la suma de las energías cinéticas y potenciales de cada una de las masas independientes

$$L = K_c + K_p - V_c = \frac{1}{2}M\dot{x}^2 + \frac{1}{2}m(\dot{x}^2 + 2l\dot{\theta}\dot{x}\cos\theta + l^2\dot{\theta}^2) + \frac{1}{2}J\dot{\theta}^2 - mgl \cos \theta \quad (\text{C.6})$$

El vector de coordenadas generalizadas q (vector de posición de las masas involucradas en el sistema) está compuesto por la posición del carro y el ángulo del péndulo con respecto a la vertical

$$q(t) = \begin{bmatrix} x(t) \\ \theta(t) \end{bmatrix} \quad (\text{C.7})$$

La ecuación de Euler-Lagrange C.1 para este sistema en particular está dada por

$$\begin{bmatrix} \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{x}} \right) - \frac{\partial L}{\partial x} = \tau_1 \\ \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}} \right) - \frac{\partial L}{\partial \theta} = \tau_2 \end{bmatrix} \quad (\text{C.8})$$

donde, desarrollando (C.8), resulta

$$(M + m)\ddot{x} + ml\ddot{\theta} \cos \theta - ml\dot{\theta}^2 \sin \theta = u \quad (\text{C.9})$$

$$(ml^2 + J)\ddot{\theta} + ml\ddot{x} \cos \theta - mgl \sin \theta = 0$$

donde $\tau_1 = k_{motor}u$ y $\tau_2 = 0$, ya que sólo se tiene un actuador, cuya constante $k_{motor} = 1$ es dada por el fabricante [57]. Las ecuaciones dinámicas (C.9) describen el comportamiento del sistema carro péndulo invertido. Estas ecuaciones pueden escribirse en forma más compacta como

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau \quad (\text{C.10})$$

donde

$$q = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}; \dot{q} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix}; \ddot{q} = \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix}, \text{ y}$$

$$D(q) = \begin{bmatrix} M + l & ml \cos(q_2) \\ ml \cos(q_2) & ml^2 + J \end{bmatrix}$$

$$C(q, \dot{q}) = \begin{bmatrix} 0 & -ml\dot{q}_2 \sin(q_2) \\ 0 & 0 \end{bmatrix}$$

$$G(q) = \begin{bmatrix} 0 \\ -mgl \sin(q_2) \end{bmatrix}$$

$$\tau = \begin{bmatrix} \tau_1 \\ 0 \end{bmatrix}$$

donde $D(q)$ es la matriz simétrica y definida positiva de inercias, $C(q, \dot{q})$ es la matriz de Coriolis - centrípeta (los términos que tienen un producto del tipo \dot{q}_i^2 son llamados fuerza centrípeta, cuando éste tiene un producto del tipo $\dot{q}_i \dot{q}_j, i \neq j$ son llamados términos de Coriolis), y $G(q)$ es el vector que contiene el término debido al efecto de la gravedad.

C.1.1.3. Representación en variables de estado

Utilizando la variables de estado

$$x_1 = x; \quad x_2 = \theta; \quad x_3 = \dot{x}; \quad x_4 = \dot{\theta}$$

el sistema C.9 se puede representar mediante un sistema de ecuaciones de la forma

$$\dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})u \quad (\text{C.11})$$

donde $\mathbf{x}, \dot{\mathbf{x}}, \ddot{\mathbf{x}} \in \mathbb{R}^4, u \in \mathbb{R}$ y

$$f(x) = \begin{bmatrix} x_3 \\ x_4 \\ \frac{m^2 gl^2 \sin x_2 \cos x_2 - (ml^2 + J)mlx_4^2 \sin x_2}{-(M+m)(ml^2 + J) + m^2 l^2 \cos^2 x_2} \\ \frac{-mgl(M+m) \sin x_2 + m^2 l^2 x_4^2 \sin x_2 \cos x_2}{-(M+m)(ml^2 + J) + m^2 l^2 \cos^2 x_2} \end{bmatrix} \quad (\text{C.12})$$

$$g(x) = \begin{bmatrix} 0 \\ 0 \\ \frac{-(ml^2+J)}{-(M+m)(ml^2+J)+m^2l^2 \cos^2 x_2} \\ \frac{ml \cos x_2}{-(M+m)(ml^2+J)+m^2l^2 \cos^2 x_2} \end{bmatrix} \quad (C.13)$$

C.2. Comentarios

La determinación del modelo matemático del sistema, ver *Ápndice C*, lleva consigo una gran cantidad de pasos los cuales utilizan herramientas como la formula de Euler-Lagrange y la linealización por series de Taylor. En este trabajo únicamente se muestran los resultados obtenidos para el sistema en particular. En [43] se muestra un análisis más detallado para la obtención del modelo matemático del sistema carro péndulo invertido.

Es importante mencionar que el modelo matemático obtenido en este capítulo sólo es empleado para propósitos de simulación numérica ya que el diseño del controlador propuesto es libre del modelo. Sistema no lineal subactuado tipo carro péndulo invertido

Los sistemas tipo péndulo invertido (SPI) están formados por un péndulo en el cual uno de sus extremos esta unido mediante una articulación a una base móvil lo que le permite girar libremente y, debido a que la aceleración angular del péndulo no puede ser controlada, estos sistemas son considerados ejemplos de sistemas subactuados.

Existen básicamente dos problemas relacionados con el control de péndulos invertidos que se han estudiado. El primero, es el de llevar el péndulo desde cualquier posición y en particular de la posición colgante natural hasta la posición invertida. A este problema se le conoce como swing up. El segundo problema consiste en estabilizar al

péndulo alrededor del punto de equilibrio inestable.

Variable	Nombre	Unidades	
τ_1	Torque del motor	$N - m$	
u	Entrada de control	<i>volts</i>	
p	Posición del carro	m	
θ	Ángulo entre la vertical y la posición del péndulo	<i>Grados</i>	
Parámetros	Valor	Unidades	
l	Longitud al centro de masa del péndulo	0.32	m
m	Masa del péndulo	0.23	Kg
M	Masa del carro	0.52	Kg
g	Constante de gravedad	9.81	m/seg^2
J	Inercia de la barra	0.007	$Kg * m^2$
k_{motor}	Constante de acoplamiento electromecánico	1.0	$N - m/volts$

Tabla C.1: Variables y parámetros del sistema carro - péndulo invertido.

