



**UNIVERSIDAD AUTÓNOMA
DEL ESTADO DE HIDALGO**



INSTITUTO DE CIENCIAS BÁSICAS E INGENIERÍA

CENTRO DE INVESTIGACIÓN EN TECNOLOGÍAS DE INFORMACIÓN Y SISTEMAS

MAESTRÍA EN CIENCIAS EN AUTOMATIZACIÓN Y CONTROL
CONTROL INTELIGENTE DE SISTEMAS NO LINEALES EMPLEANDO
REDES NEURONALES ARTIFICIALES DE TERCERA GENERACIÓN

T E S I S

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

PRESENTA:

LUIS REY LECHUGA GUTIÉRREZ

ASESORES:

LUIS ENRIQUE RAMOS VELASCO

JOEL SUÁREZ CANSINO

PACHUCA DE SOTO, HIDALGO, JUNIO DE 2014



Luis Rey Lechuga Gutiérrez

**CONTROL INTELIGENTE
DE SISTEMAS NO
LINEALES EMPLEANDO
REDES NEURONALES
ARTIFICIALES DE
TERCERA GENERACIÓN**

Resumen

CONTROL INTELIGENTE DE SISTEMAS NO LINEALES EMPLEANDO REDES NEURONALES ARTIFICIALES DE TERCERA GENERACIÓN

Se presenta el diseño e implementación de un control inteligente para sistemas no lineales de naturaleza estable de tipo SISO, involucrando al área de las redes neuronales artificiales de tercera generación. La identificación y control del sistema propuesto, se logra por medio de neuronas artificiales de tercera generación, y posteriormente se prueba dicha arquitectura bajo diferentes tipos de entrenamiento y modelos neuronales para comprobar su eficacia, esta propuesta va más allá de buscar sólo la creación de un control adaptable aplicable a sistemas no lineales, sino que se busca que este control tenga un mejor desempeño contra los controles neuronales existentes hasta ahora. Se espera que esto ocurra pues las redes neuronales de tercera generación tienen una gran similitud con las redes neuronales biológicas, esto les da una gran velocidad de procesamiento y versatilidad a la adaptación, en comparación con sus antecesoras.

Se presentan los resultados obtenidos bajo los siguientes casos:

- Control de un sistema no lineal de primer orden.

Entrenamiento: Algoritmo SpikeProp, Algoritmo Regla Delta Generalizada

Modelos: Perfect Integrate and Fire (PIF), Leaky Integrate and Fire (LIF),
Izhikevich

Finalmente, se realizó una interfaz de usuario en la plataforma de Laboratorio de Instrumentación Virtual (LabVIEW) 2011, con la cual es posible implementar este tipo de redes neuronales artificiales a sistemas físicos, como lo es el motor de corriente directa y el péndulo invertido.

Abstract

INTELLIGENT CONTROL OF NONLINEAR SYSTEMS USING ARTIFICIAL NEURAL NETWORKS OF THIRD GENERATION

This thesis presents the design and implementation of a SISO type intelligent control for nonlinear systems of a stable nature, which involves the field of third generation of artificial neural networks. The identification and control of the proposed system are achieved through third generation artificial neurons, which define a feedforward artificial neural network architecture. The Perfect Integrator, the Leaky Integrator and the Izhikevich models respectively define three different variants of the feedforward architecture.

The supervised training of the network considers two encoding types of the firing neurons; namely, one known as Delay Coding (in fact, this encoding is a kind of Temporal Encoding) and the other known as a train of pulses (in fact, this encoding is a kind of Frequency Rate). The combination of the supervised training along with the corresponding encoding define the SpikeProp and the Generalized Delta Rule training algorithms.

Hence, the proposed artificial neural network architectures are tested under several conditions to estimate their performance in the control of a dynamical system of first order.

The thesis not only creates an adaptable control, which is applicable to nonlinear systems, but also achieves a better performance when compared against the current existing controls, based on second generation artificial neural.

Since the third generation artificial neural networks have a great similarity with the biological neural networks, they have an acceptable velocity of processing and versatil adaptability, which improves the well known performance of the second ge-

neration of artificial neural networks.

We present the results obtained for the following cases:

- Control of a nonlinear system to first order.
Training: SpikeProp algorithm, Generalized Delta Rule algorithm.
Models: PIF, LIF, Izhikevich.

Finally, a user interface was designed in LabVIEW 2011 platform, which is implemented for the control of physic systems using the third generation of artificial neuronal networks.

Índice general

Resumen	III
Abstract	V
Índice general	VII
Índice de figuras	XI
Índice de tablas	XV
Acrónimos y abreviaturas	XVII
1. Introducción	1
1.1. Introducción	1
1.1.1. Descripción de la problemática	2
1.1.2. Planteamiento del problema	3
1.2. Hipótesis	3
1.3. Justificación	4
1.4. Resultados obtenidos	5
1.5. Organización de la tesis	6
2. Estudio del estado del arte	7
2.1. Introducción	7
2.2. Trabajos relevantes	8
2.3. Marco teórico	9
2.4. Comentarios	9
3. Redes neuronales de tercera generación	11
3.1. Introducción	12
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	16
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	18
3.2.1.1.	El modelo de Hodgkin y Huxley	19
3.2.1.2.	El modelo Izhikevich	21
3.2.2.	Modelos umbral y disparo	22
3.2.2.1.	Integración y disparo perfecto	24
3.2.2.2.	Integración y disparo ideal	24
3.2.2.3.	Modelo del pulso de respuesta	26
3.2.2.4.	El kernel	27
3.2.2.5.	SRM simplificado	27
3.2.3.	Comparación de modelos neuronales	29
3.3.	Código neuronal	29
3.3.1.	Codificación por tasa	30
3.3.2.	Codificación temporal	30
3.3.3.	Codificación por población	32
3.4.	Métodos de aprendizaje	33
3.4.1.	Aprendizaje no supervisado	33
3.4.1.1.	Método de Hebb	33
3.4.1.2.	Método del Hebbiano local	36
3.4.2.	Aprendizaje supervisado	36
3.4.2.1.	Método SpikeProp	37
3.4.2.2.	Regla modificada de Widrow-Hoff o regla delta generalizada a spikes	39
3.5.	Comentarios	41
4.	Resultados	43
4.1.	Introducción	43
4.2.	PID-S	45
4.3.	Resultados en simulación numérica	47
4.3.0.3.	Sistema lineal de primer orden	47
4.3.1.	Sistema no lineal de segundo orden	51
4.4.	Resultados de laboratorio	61
4.4.1.	Sistema no lineal de primer orden	61
4.4.2.	Sistema no lineal de segundo orden	73
4.5.	Conclusiones	76
4.6.	Trabajos a futuro	76
	Glosario	77
	Bibliografía	83

A. Desarrollo del control por redes neuronales de tercera generación	89
A.1. Introducción	89
A.2. Control de un valor real	90
A.2.1. Inicialización de las variables	90
A.2.2. Entrenamiento de una neurona	91
A.3. Identificación de un sistema	93
A.4. Control de un sistema dinámico	100
B. Comparación entre el PID-S y el PID	103
B.1. Introducción	103
B.2. Resultados en simulación numérica	104
B.3. Resultados de laboratorio	105
C. Artículo propuesto a revisión	108

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. Modelo Hodking y Huxley	19
3.6. Modelo de potencial de membrana de Hodking y Huxley	21
3.7. Parámetros del modelo de Izhikevich	23
3.8. Modelo Izhikevich	23
3.9. Respuesta modelo LIF	25
3.10. Modelo PIF	25
3.11. Modelo LIF	26
3.12. Kernel SRM	28
3.13. Modelo SRM_0	28
3.14. Codificación por tasa	31
3.15. Codificación por población	32
3.16. Modelo de Hebb	34
3.17. Ventana de aprendizaje de Hebb	35
3.18. Función de aprendizaje local de Hebb	37
3.19. Relación entrada salida	40
4.1. PID	44
4.2. PID-S	45
4.3. Diagrama a bloques de codificación y decodificación	46
4.4. Control PID-S neurona PIF, regla modificada de Widrow-Hoff	48
4.5. Control PID-S neurona LIF, regla modificada de Widrow-Hoff	49
4.6. Control PID-S neurona Izhikevich, regla modificada de Widrow-Hoff	50
4.7. Ley de control PID-S, neuronas PIF y LIF, regla modificada de Widrow-Hoff	52
4.8. Ley de control PID-S, neurona Izhikevich, regla modificada de Widrow-Hoff	53

4.9. Control PID-S neurona PIF, entrenamiento SpikeProp	54
4.10. Control PID-S neurona LIF, entrenamiento SpikeProp	55
4.11. Control PID-S neurona Izhikevich, entrenamiento SpikeProp	56
4.12. Ley de control PID-S, neuronas PIF y LIF, entrenamiento SpikeProp	57
4.13. Ley de control PID-S, neuronas Izhikevich, entrenamiento SpikeProp	58
4.14. Pendulo invertido	58
4.15. Control del péndulo	59
4.16. Error del péndulo	60
4.17. Plataforma experimental	61
4.18. Control PID-S neurona PIF, entrenamiento relación de trenes de pulso entrada-salida	63
4.19. Control PID-S neurona LIF, entrenamiento relación de trenes de pulso entrada-salida	64
4.20. Control PID-S neurona Izhikevich, entrenamiento relación de trenes de pulso entrada-salida	65
4.21. Ley de control PID-S, neuronas PIF y LIF, entrenamiento relación de trenes de pulso entrada-salida	66
4.22. Ley de control PID-S, neuronas Izhikevich, entrenamiento relación de trenes de pulso entrada-salida	67
4.23. Control PID-S neurona PIF, entrenamiento SpikeProp	68
4.24. Control PID-S neurona LIF, entrenamiento SpikeProp	69
4.25. Control PID-S neurona Izhikevich, entrenamiento SpikeProp	70
4.26. Ley de control PID-S, neuronas PIF y LIF, entrenamiento SpikeProp	71
4.27. Ley de control PID-S, neuronas Izhikevich entrenamiento SpikeProp .	72
4.28. Interfaz para control del péndulo	73
4.29. Control del péndulo resultados laboratorio	74
4.30. Error del péndulo resultados laboratorio	75
A.1. Codificación por retardos	91
A.2. Entrenamiento de una neurona	91
A.3. Entrenamiento gráfico de una neurona	92
A.4. Evolución del error primer entrenmiento	93
A.5. Evolución de pesos, primer entrenamiento	94
A.6. Potencial de membrana final, primer entrenamiento	95
A.7. Esquema de identificación	95
A.8. Entrenamiento de identificación	97
A.9. Error durante el entrenamiento de identificación	98
A.10. Estimación neuronal	99
A.11. Potencial de membrana en la identificación	99
A.12. Esquema de control	100

A.13. Control de un sistema dinámico	101
A.14. Error del control de un sistema dinámico	102
A.15. Ley de control de un sistema dinámico	102
B.1. Comparación de controles, simulación	104
B.2. Comparación de ley de control, simulación	105
B.3. Comparación de controles, físicos	106
B.4. Comparación de ley de control, simulación, físicos	107

Índice de tablas

3.1. Parámetros de Hodking y Huxley	20
3.2. Comparación de parámetros	29
3.3. Algoritmo SpikeProp	39

Acrónimos y abreviaturas

Siglas	Descripción
cm	Unidad de medida de distancia expresada en centímetros
CS	Cuckoo Search Algorithm
DAQ	Tarjeta de Adquisición de Datos
EPSP	Potencial de Excitación Postsináptico
GRF	Campos receptivos con múltiples funciones Gaussianas
IPSP	Potencial Inhibitorio Postsináptico
LGR	Lugar Geométrico de las Raíces
LM	Levenberg-Marquardt
LTD	Long Term Depression
LTP	Long Term Potentiation
ms	Unidad de medida de tiempo expresada en milisegundos
mV	Unidad de medida de potencial expresada en milivolts
PID	Controlador Proporcional, Integral y Derivativo
PSO	Particle Swarm Optimization
ReSuMe	Remote Supervised Method
RNA	Redes Neuronales Artificiales
SAP	Potencial Post Pulso
SISO	Sistema de Una Entrada y Una Salida
SNN	Spiking Neural Networks
SRM	Modelo del Pulso de Respuesta
STDP	Spike Timing Dependent Plasticity
STR	Sistema en Tiempo Real
SWAT	Synaptic Weight Association Training

Capítulo 1

Introducción

1.1. Introducción

En los últimos años, se ha incrementado el estudio y aplicación de las Redes Neuronales Artificiales (RNA) en las áreas de cómputo inteligente y control, esto debido principalmente a su capacidad de aprendizaje y su gran versatilidad en la identificación de sistemas y ahora que ha surgido la nueva generación de redes neuronales artificiales, se tiene una herramienta más poderosa para el control de una gran cantidad de sistemas dinámicos, por lo que se vaticina un auge en el interés de esta área en particular.

En la industria se usan una gran diversidad de máquinas, con distintas características tanto mecánicas como eléctricas y diversas configuraciones, lo cual, implica excesivo trabajo por parte de los ingenieros si se desean implementar leyes de control basadas en el modelo de este tipo de sistemas, lo que significa que realicen una serie de pruebas y cálculos para la identificación paramétrica, y puesta en marcha de procesos industriales, cabe mencionar que si la dinámica de la planta cambia ya sea por el desgaste intrínseco del sistema generado por fricción, o por esfuerzos axiales no contemplados provenientes de sistemas periféricos, o simplemente la vida útil de algún componente se agota, la representación matemática que se propuso *a priori* para modelar la dinámica del sistema ya no será la correcta. Razones por las cuales surge la idea de proponer un algoritmo de identificación y 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 o la referencia, la señal de control siempre será la adecuada. Para comprobar la eficacia del algoritmo propuesto con redes neuronales de tercera generación en sistemas reales, se realizan las pertinentes pruebas con dos modelos físicos, el Motor de Corriente Directa (MCD) y Péndulo Invertido.

1.1.1. 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, o modelos difusos, o modelos neuro-difusos, entre otros.

Aunque no necesariamente se requiere del modelo matemático de la planta para aplicarle una ley de control adecuada, si es necesario el formular un sistema de ecuaciones diferenciales para el diseño de un modelo matemático. La opción de controlar un sistema a través de su modelo, tiene el problema de que se debe proponer adecuadamente el sistema con ecuaciones diferenciales que lo describen correctamente. Lo cual no es tan sencillo, 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.

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, $E^2(\mathbf{x}, \mathbf{y}, \mathbf{w})$. La determinación de los parámetros \mathbf{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 $E^2(\mathbf{x}, \mathbf{y}, \mathbf{w})$. 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 [27].

Se pueden encontrar también algunas dificultades cuando se opta por modelos lógico difusos. Estos modelos han sido catalogados por algunos autores como poco flexibles, además de que el costo del tiempo de implementación de este tipo de alternativas aumenta considerablemente en situaciones de sintonización [2]. Este mismo autor resalta otros diferentes inconvenientes de un modelo lógico difuso.

1.1.2. Planteamiento del problema

Las técnicas tradicionales para el control de un sistema, requieren en su mayoría definir una correcta representación matemática de éste, dicha representación está constituida por variables y parámetros que, durante la fase de modelado, resultan difíciles de especificar u obtener, además de que generalmente un gran porcentaje de estos parámetros pueden cambiar por factores inherentes a la planta, como calor, humedad, fricción y/o vida útil de sus componentes. Por lo que su modelo se encuentra en constante cambio, esto puede dificultar el control que se le aplica al sistema pues en determinados casos puede ser necesario reformular las ecuaciones que en algún momento representaron correctamente su dinámica.

En el presente trabajo de tesis se plantea el siguiente problema: El cómo controlar sistemas no lineales donde se asume, que no se conoce su modelo matemático.

1.2. Hipótesis

Las características no lineales que poseen las redes neuronales artificiales de tercera generación aunado con la mayor similitud que existe entre su dinámica y la dinámica de las redes neuronales biológicas reales, hacen que ésta generación de redes neuronales artificiales sea el candidato ideal para analizar e instrumentar un control inteligente aplicado a sistemas no lineales, las características intrínsecas de plasticidad y memoria que poseen las redes neuronales artificiales de tercera generación proveen una mejoría en comparación con las cualidades de las redes neuronales artificiales de generaciones anteriores. Dicha mejoría se manifiesta en el número de neuronas implementadas para la resolución de un mismo problema, el número de iteraciones requeridas para converger a la referencia deseada y por ende se reduce el número de iteraciones para determinar apropiadamente los pesos presinápticos, para el caso de aprendizaje supervisado.

1.3. Justificación

En la actualidad los modelos que mejor representan al comportamiento de las redes neuronales biológicas están dados por las redes neuronales de pulsos [39], 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 estos 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 [61]. 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 [33].

El área de la optimización de algoritmos de redes neuronales se tiene el PSO(Particle Swarm Optimization) para entrenamiento de redes neuronales de tipo Spike de forma óptima, los resultados muestran que este método produce una secuencia de pulsos muy similar a la secuencia de pulsos de salida deseada, reduciendo el error de convergencia, con un menor número de ciclos de ejecución en comparación a entrenamientos clásicos, esto asegura que el tiempo de convergencia disminuya [4].

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 [57].

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 [45].

1.4. Resultados obtenidos

Se presentan los objetivos específicos que se obtuvieron a lo largo del desarrollo de la tesis los cuales en conjunto sirven como piezas claves para finalmente alcanzar el objetivo global.

Objetivo general

Implementar las redes neuronales de tercera generación para el control de sistemas no lineales tipo SISO de naturaleza estable.

Objetivos específicos

Los objetivos específicos de la tesis son los siguientes:

- **Identificar el modelo dinámico de una planta no lineal por medio de redes neuronales de tercera generación.** Implementando diferentes algoritmos de entrenamiento como lo es SpikeProp (El algoritmo de entrenamiento SpikeProp se sustenta en una actualización de los pesos sinápticos basada en la relación que existe entre todas las neuronas pre sinápticas y post sinápticas que se encuentran en la correspondiente pre y post ramificación de cada neurona) [51], QuickProp (Entrenamiento basado en SpikeProp, pero con una convergencia más rápida pues se anticipa la dinámica de la evolución de los pesos) [14], Re.Su.Me. (El método Re.Su.Me. es un nuevo método de aprendizaje supervisado, que implementa las reglas Hebbianas con un novedoso concepto de supervisión remota. El error generado en el tiempo de aprendizaje inicia con valores relativamente pequeños y tiende a cero con el transcurso del tiempo. Además de poseer la cualidad de tener un control multivariable, gracias a que su supervisión remota puede aplicarse a un número de salidas y no sólo en una, como otros métodos de entrenamiento [45]). SWAT (Synaptic Weight Association Training) para redes neuronales de tercera generación (en el caso particular del entrenamiento SWAT se conoce como un tipo de regla que produce una distribución unimodal de pesos, donde la dimensión de la ventana de plasticidad asociada es modulada causando la estabilidad después de un período de entrenamiento. Por lo que es un método que garantiza una rápida convergencia así como muy exacta)[33].
- **Hallar la mejor ecuación para la codificación y decodificación de señales entrada y salida,** respectivamente. Las neuronas de tercera generación o neuronas Spike reciben su nombre por su particular forma de transmitir información por medio de pulsos, y no sólo la frecuencia de los pulsos se puede

interpretar como información (Codificación por tasa [52]) sino también el tiempo en que una neurona tarda en saturarse o el tiempo que a ésta le lleva en disipar su propia carga eléctrica (Codificación temporal [54]), así pues la obtención del método correcto de codificación y decodificación en entrada y salida respectivamente de la red neuronal son otros dos objetivos particulares para la tesis.

- **Controlar un sistema dinámico en simulación**, uno de los objetivos específicos más relevantes es el de unir el conocimiento de las redes neuronales artificiales de tercera generación con el área de la ingeniería de control. Este tipo de sinergia se ha llevado a cabo utilizando redes neuronales de segunda generación en conjunto con un control PID [26] 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 es llevado a cabo con los algoritmos de entrenamiento previamente mencionados [47].
- **Realizar trabajo experimental** del control de velocidad de un MCD y posición de un péndulo invertido con el algoritmo propuesto.
- **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 realiza 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.

1.5. Organización de la tesis

En el Capítulo 2, se analizan los trabajos relacionados con el desarrollo de esta investigación. En el Capítulo 3, se da un panorama general de lo que son las redes neuronales artificiales de tercera generación, y se da una revisión a los trabajos que sobresalen para el diseño y desarrollo de arquitecturas de redes neuronales de tercera generación. En el Capítulo 4, se muestran los resultados, aplicando la arquitectura propuesta entrelazando las redes neuronales artificiales de tercera generación y el área de control, con diferentes arquitecturas de entrenamiento y modelos neuronales. En el mismo capítulo se realizan las conclusiones y trabajos futuros, consecuencias de esta tesis de maestría. En el Anexo A se muestran de manera general los pasos y resultados que se desarrollaron para llegar finalmente al control PID-S. Finalmente en el Anexo B se realiza una comparación entre el PID-S y el control más implementado en la industria, el PIDclásico.

Capítulo 2

Estudio del estado del arte

El objetivo del presente capítulo es mostrar las ventajas y desventajas reportadas en los trabajos de investigación, que se han venido realizando sobre las redes neuronales artificiales, enfocadas hacia el área de control.

La organización de este capítulo es la siguiente: en la Sección 2.1 se da una breve introducción sobre las áreas de intersección en las que se desarrolla este trabajo. En la Sección 2.2 se presentan las ventajas y desventajas que muestran cada uno de los trabajos de investigación que están directamente relacionados con este trabajo de tesis. En la Sección 2.3 se describen las herramientas teóricas que se emplearán para resolver el problema, finalmente, en la Sección 2.4 se resumen los comentarios de los trabajos de investigación sobre los cuales se formuló el punto de partida para la realización de la presenta tesis.

2.1. Introducción

A continuación se mencionan algunos trabajos relacionados con la propuesta a desarrollar, así como aplicaciones que se han elaborado utilizando redes neuronales de tercera y segunda generación enfocadas al control y resolución a problemas de ingeniería.

La unidad fundamental de procesamiento del sistema nervioso central es la neurona, en la cual se ha fijado particular interés por emular su funcionamiento [62], de esta manera surge la inquietud por la creación del modelado matemático de las redes neuronales reales, el cual es una relación entrada-salida cuya dinámica compite con la de sus equivalentes biológicas, pues la información no está contenida en el valor del pulso sino en la frecuencia de este [28].

2.2. Trabajos relevantes

La implementación de las redes neuronales artificiales en el área de control de plantas dinámicas, ha ido en auge en los últimos años, esto debido a su gran capacidad de adaptación y gran flexibilidad para la identificación de sistemas 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 [12], pero si la señal de referencia y/o el cambio del modelo no son derivables temporalmente, el control ocasiona que la variable de proceso tenga un sobre impulso 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 mejorar el control PID de ganancias ajustables, las cuales ofrecen una mejor respuesta en comparación con teorías de auto ajustes clásicos como cancelación y reasignación de polos [22], 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[15].

El control conocido como GPIDNN soporta de una mejor manera los retardos inherentes del sistema, antes de que éste se haga inestable [65], 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, como se verá más adelante.

La implementación de redes neuronales de tercera generación para el control de una planta de primer orden ya se ha llevado a cabo [47], arrojando muy buenos resultados cuando se utiliza una ecuación diferencial de segundo orden como señal de referencia, es pertinente mencionar que la identificación y control de sistemas dinámicos sólo 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.

Existen diferentes métodos para entrenar redes neuronales de tipo Spike y cada uno 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 los algoritmos de entrenamiento SpikeProp y entrenamiento de la Regla Delta Generalizada 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 son aplicados a la identificación y control de un sistema físico, el desempeño de las redes neuronales artificiales de tercera generación es el adecuado pues se conservan las características y cualidades de estos entrenamientos.

2.3. Marco teórico

Las herramientas teóricas que se emplearán para resolver el problema, sugieren primero un estudio a los sistemas dinámicos, tanto lineales [5] como no lineales [55], para comprender su comportamiento y las implicaciones de un control en sistemas dinámicos por redes neuronales artificiales de tercera generación, lo cual nos lleva al estudio sobre los controles clásicos involucrados [42], por otra parte se estudian las dinámicas de las redes neuronales artificiales de segunda [17] y tercera generación [62], y más específicamente con un enfoque a la identificación de sistemas en tiempo real [47], [21]. Posterior a la selección del algoritmo de control se realiza la comparación en simulación con un sistema no lineal por lo que es necesario poseer conocimiento acerca del proceso de modelado e identificación de sistemas dinámicos [36].

Para la etapa de implementación física del control sólo se necesita conocer sobre dos áreas específicas, electrónica, de la cual surgirá la etapa de potencia e interface necesaria [9] y programación para cargar la lógica del PID-Spike a la tarjeta de adquisición [20].

Como se puede apreciar la teoría que respalda a este trabajo abarca desde la parte teórica y va sustentando al mismo hasta poder implementarlo físicamente.

2.4. Comentarios

Las redes neuronales de tercera generación representan el siguiente paso en la evolución del control inteligente, su gran similitud con las redes neuronales biológicas reales forjan grandes ventajas en la implementación.

Al emular lo que a la naturaleza le llevó miles de años de evolución no sólo nos acerca más a entender el funcionamiento del comportamiento del cerebro humano real,

sino que matemáticamente aprenden, recuerdan y evolucionan con una velocidad y exactitud superior a sus antecesoras de segunda generación.

Aunque por otro lado las principales desventajas de las redes neuronales de tercera generación son:

- **La escasez de la información sobre el tema.** Las redes neuronales de tercera generación, son un área nueva de investigación activa, surgieron alrededor del 2000 y escasamente se comprende su funcionamiento, 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 prácticamente inexistentes.
- **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 todos los valores de ésta, sólo un subconjunto de la misma.

Capítulo 3

Redes neuronales de tercera generación

Las redes neuronales Spike o de pulsos han sido llamadas como 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 máquinas eléctricas, 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, esto para que la señal de control converja de una manera más rápida a la referencia dada.

Este capítulo se organiza de la siguiente manera, en la Sección 3.1 se aborda un panorama general sobre las redes neuronales y se divide en varias subsecciones, en donde cada una explica tanto el comportamiento y características de una neurona real, así como las características y comportamiento de las neuronas artificiales. En la Sección 3.2 se discute sobre los modelos neuronales más utilizados hasta ahora, ya sea por su facilidad en su implementación computacional o por su similitud con las neuronas biológicas. La Sección 3.3 aborda el tema de la codificación y decodificación necesaria para ligar los modelos neuronales con algún otro tipo de sistemas. En la Sección 3.4 se encuentran los métodos de entrenamiento que utilizan las neuronas artificiales de tercera generación, la sección naturalmente se divide en aprendizajes supervisados y no supervisados, las cuales se subdividen en dos ejemplos de cada uno. Finalmente en la Sección 3.5 se realizan algunos comentarios respecto a este capítulo.

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 [39], 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 sólo tenía dos estados (“alto” o “bajo”), basadas en la suma de las señales de entrada multiplicadas por sus pesos, este tipo de redes neuronales artificiales pueden instrumentar funciones booleanas, su principal desventaja es que no tienen la capacidad de aprender, pues sus pesos sinápticos no evolucionan.

La segunda generación de redes neuronales artificiales utiliza funciones de activación sigmoideas con cambio de umbral, esta segunda generación es mucho más poderosa que las neuronas de primera generación, y desde la era de la computación digital la capacidad de procesamiento ha permitido que el número de neuronas para resolver un mismo problema se reduzca, en comparación con la computación analógica [60]. 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 gradiente del 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 la segunda generación posee la cualidad de aprender y la primera no [39].

Comparado con una computadora a la que le toma alrededor de tres segundos realizar el reconocimiento facial de alguna persona, las neuronas situadas en la corteza cerebral tienen un funcionamiento más rápido, pues para realizar dicho reconocimiento 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 [50]. 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

[18], 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 potencial de acción, 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 [62]. 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 acercan aún más al comportamiento biológico real. Pues sus señales codifican información temporal, lo cual significa que manejan información espacio-temporal. Wolfgang [38], 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, el soma y el axón, además de un proceso propio de éstas llamado sinapsis. Generalmente, las dendritas son quienes reciben la información de las neuronas periféricas conectadas a éstas, el 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 presináptica y consecuentemente la neurona que recibe el pulso (la neurona “*I*”) es llamada una neurona postsináptica.

Hodgkin y Huxley [29] encontraron de manera experimental, que el axón de un calamar gigante maneja información codificada por trenes de pulsos [40], Figura 3.1.

3.1.2. Potencial de la membrana

Todas las neuronas están rodeadas por iones positivos y negativos. En su interior hay un exceso de cargas negativas y en el exterior hay un exceso de carga positiva. Ambas cargas generan el potencial de membrana.

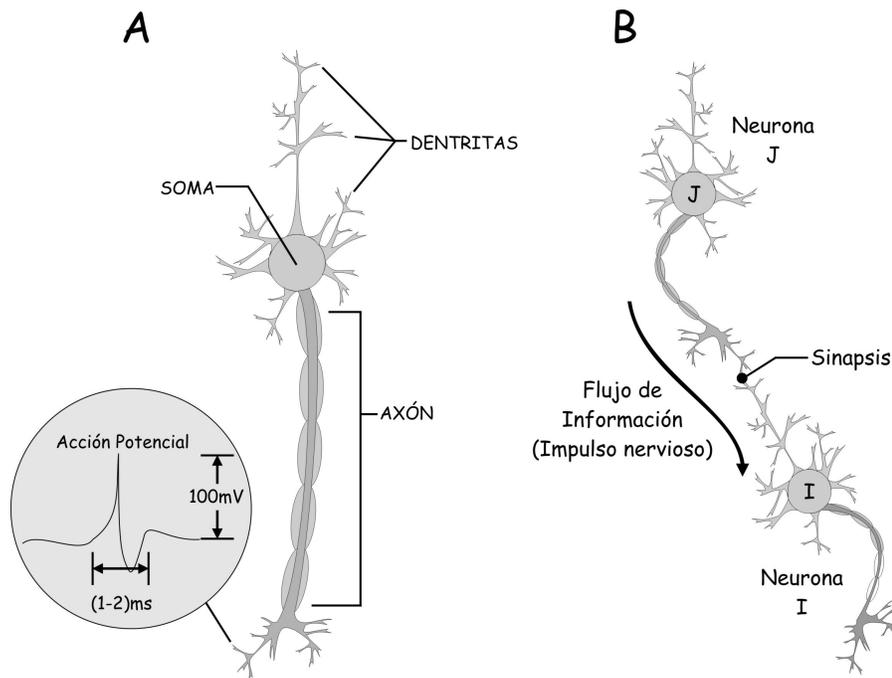


Figura 3.1: **A.** Se muestra un ejemplo de la acción potencial neuronal, la cual prácticamente es un pulso de voltaje con una duración de 1-2 ms, y una amplitud de 100 mV. **B.** Transmisión de la señal (información) de la neurona presináptica J a la neurona postsináptica I, donde la sinapsis entre estas neuronas es denotada por el círculo negro [62].

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 y 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 $-60mV$ a los $-70mV$.

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) [18], Figura 1.2. Si el potencial de membrana recibe un impulso esto causará depolarización o hiperpolarización y 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 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 [18]. El pulso generado en una neurona es idéntico al de cualquier otra, así como la señal que se mueve de la neurona presináptica a la neurona postsináptica [62]. El conjunto de pulsos emitidos por la neurona dentro de una ventana de tiempo arbitraria, es nombrado tren de pulsos y se puede representar por la siguiente ecuación.

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

El subíndice i define el número de neurona y el superíndice es el pulso emitido por la neurona en el instante de tiempo t , siendo n el pulso más reciente emitido por la n ésima neurona.

Exactamente después de la transmisión del pulso de la neurona presináptica a la neurona postsináptica, 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 [62]. 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 reposo esto siempre después de que un pulso es emitido, se conoce como SAP (Spike After Potencial), Figura 3.2

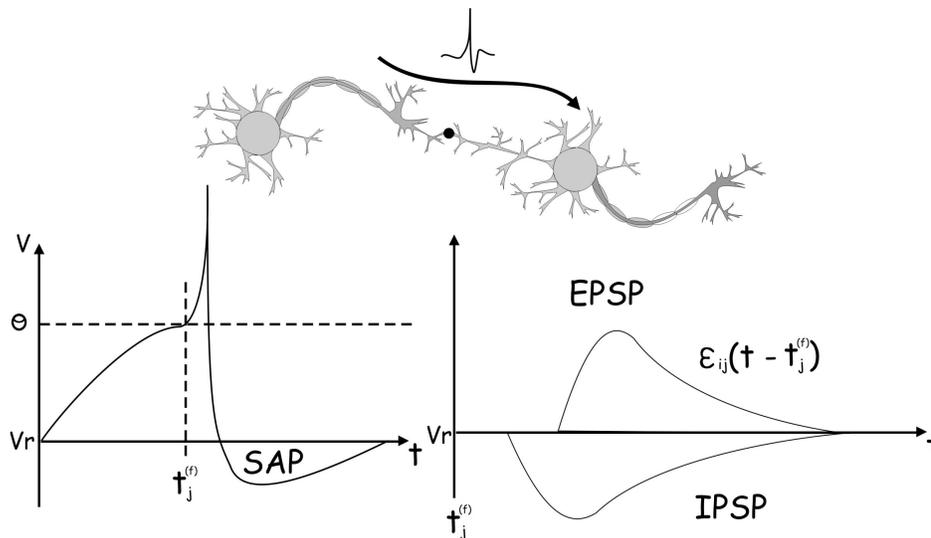


Figura 3.2: El potencial de membrana va incrementando y en el tiempo $t_j^{(f)}$ el potencial de la membrana rebasa el umbral y esto produce un pulso [40], posteriormente la neurona se relaja y su potencial de membrana se coloca debajo del potencial de reposo y va regresando lentamente al potencial de reposo. El fenómeno de la sinapsis se puede dividir en un potencial de excitación postsináptico (EPSP) y un potencial inhibitorio postsináptico (IPSP)

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 [18], en esta tesis sólo los aspectos más relevantes se tomarán en cuenta.

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

es llenado de un líquido, el cual genera un potencial postsináptico (PSP) hacia las dendritas de la neurona postsinápticas [18]. Este proceso no ocurre instantáneamente; hay un pequeño retardo generado en cada sinapsis.

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

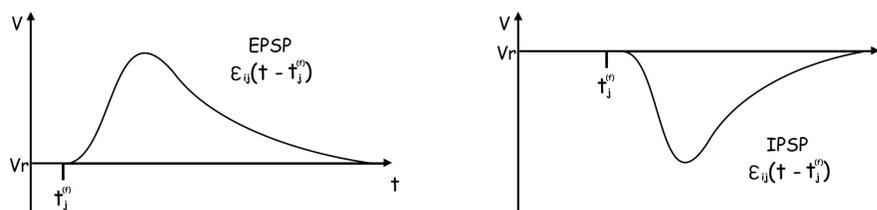


Figura 3.3: Se muestran los comportamientos del potencial postsináptico de excitación (EPSP) y el potencial postsináptico de inhibición (IPSP), ambos de una neurona biológica [39].

3.1.5. Los pesos sinápticos

Cada sinapsis va acompañada por un factor que incrementa o decrementa la señal de salida de la neurona presináptica, 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 modifiquen 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, de cada par de capas neuronales presinápticas i a sus respectivas neuronas postsinápticas j para una arquitectura feedforward. Recientes resultados experimentales, han mostrado que entre todas las interconexiones de dos neuronas biológicas puede haber múltiples sinapsis con múltiples retardos [54],[19], Figura 3.4. Esta pensamiento ha sido usado en numerosos trabajos aprovechando dichos parámetros para mejorar los algoritmos de entrenamiento.

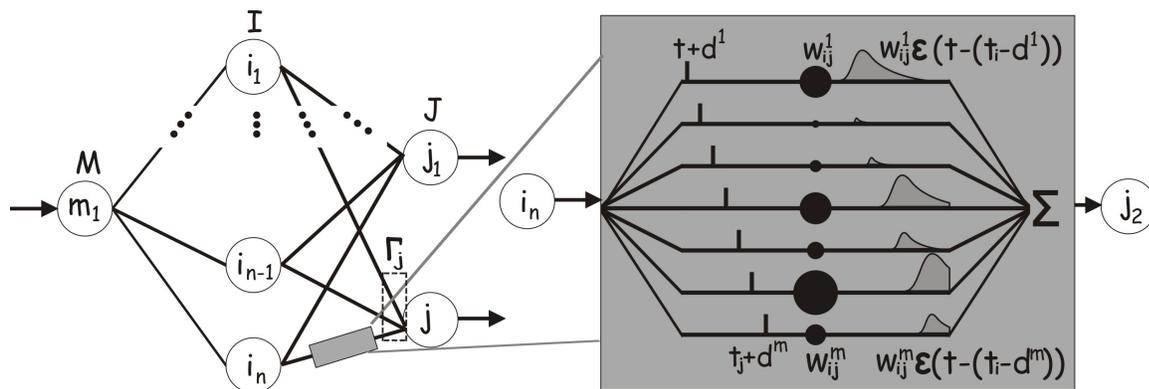


Figura 3.4: Modelo de una red neuronal simple. La capa de neuronas postsinápticas J recibe el conjunto de entradas Γ_j (Señales presinápticas de la j -ésima neurona). El acercamiento entre la interconexión de la neurona i -ésima a la neurona j -ésima muestra que cada conexión entre dos neuronas consiste 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 ($W_J I^k$) [48]. 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 [62] 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 rebaza 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 [40], basados en su experimentos ellos modelaron el funcionamiento del axón de un calamar gigante [29]. Su trabajo básicamente, describe qué ocurre con los canales de iones de la célula.

3.2.1.1. El modelo de Hodgkin y Huxley

El diagrama esquemático del modelo de Hodgkin y Huxley, puede ser visto en la Figura 3.5. La membrana semipermeable de la célula separa el interior del exterior de ésta, por medio de un líquido extra celular el cual actúa como capacitor [62].

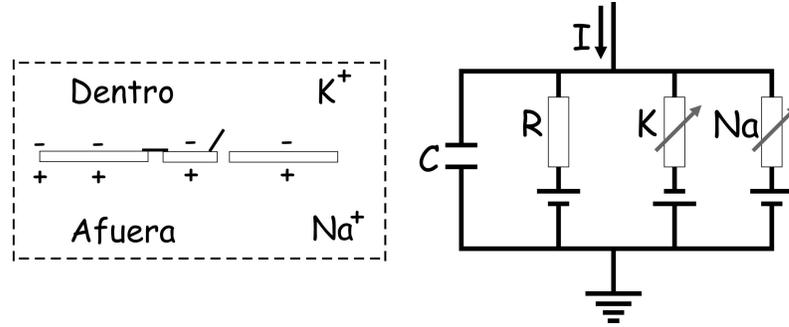


Figura 3.5: Diagrama esquemático del modelo de Hodgkin y Huxley [62]. La señal de entrada es la corriente $I(t)$ y las baterías representan el potencial de reposo forjado por la diferencia en la concentración de iones. En este modelo existen tres tipos de corriente iónica: sodio (Na), potasio (K) y corriente ideal.

Este modelo puede ser descrito por medio de una ecuación RC, la ecuación 3.3 es la expresión matemática asociada a la Figura 3.5, donde I_K es la suma de todos los canales de iones: el canal de sodio (Na), el canal de potasio (K) y un canal ideal específico con una resistencia R . La capacitancia de la membrana es $C = 1\mu F/cm^2$.

$$I(x) = I_c + \sum_k I_k(t) \quad (3.3)$$

Para poder obtener el potencial de membrana en cualquier tiempo t , la ecuación 1.3 se representa como:

$$C \frac{du}{dt} = - \sum_k I_k(t) + I(t) \quad (3.4)$$

En la ecuación 3.5 se puede observar la sumatoria de las k -ésimas corrientes I_k del modelo de Hodgkin y Huxley. Es importante mencionar que si todos los canales de iones (g_{Na} o g_K) están abiertos, estos transmitirían corriente con una máxima conductancia y de lo contrario si estos canales están cerrados no existiría paso de corriente. Las variables m , n y h , son también conocidas como variables de apertura, las cuales describen la probabilidad de que los canales se abran o cierren. Esto quiere decir que las variables m y h controlan los iones de sodio Na^+ y la variable n controla

el canal de potasio K^+

$$\sum_k I_k = g_{Na}m^3h(u - E_{Na}) + g_kn^4(u - E_k) + g_L(u - E_L) \quad (3.5)$$

donde E_{Na} , E_K y E_L , son parámetros que representan el potencial de reposo de sus respectivos canales iónicos. las variables m , n y h , son descritas por las siguientes ecuaciones diferenciales:

$$\dot{m} = \alpha_m(u)(1 - m) - \beta_m(u)m \quad (3.6)$$

$$\dot{n} = \alpha_n(u)(1 - n) - \beta_n(u)n \quad (3.7)$$

$$\dot{h} = \alpha_h(u)(1 - h) - \beta_h(u)h \quad (3.8)$$

Las funciones α y β están dadas por la tabla 3.1, la cual es una relación empírica de \mathbf{u} , dicha relación ha sido creada por Hodking y Huxley obtenida en un experimento sobre el axón de un calamar gigante.

Tabla 3.1: Parámetros de Hodking y Huxley para la ecuación 3.5.

x	E_x	g_x
Na	150mV	120 ms/cm ²
K	-12mV	36 ms/cm ²
L	10.6mV	0.3 ms/cm ²
x	α_x (u/mV)	β_x (u/mV)
n	$(0.1-0.01)/[\exp(1-0.1\mathbf{u})-1]$	$0.125 \exp(-\mathbf{u}/80)$
m	$(2.5-0.1)/[\exp(2.5-0.1\mathbf{u})-1]$	$4 \exp(-\mathbf{u}/18)$
h	$0.07 \exp(-\mathbf{u}/20)$	$1/[\exp(3-0.1\mathbf{u})+1]$

Si algún estímulo externo causa que el voltaje de la membrana se incremente, la conductancia del canal de sodio (Na) incrementa junto con m . Si este estímulo positivo permanece lo suficiente, una acción potencial (pulso) se emitirá.

La corriente del canal de potasio (K), se encarga de disminuir el potencial de membrana pues sólo deja fluir a la corriente en una sola dirección; dicho flujo de corriente del canal de potasio es controlada por n . El efecto global generado ambos canales de iones (Na , K) forja una acción potencial (pulso), el cual siempre es seguido por un sobre impulso negativo.

Si aplicamos una corriente de entrada $I(t)$ al modelo de Hodgkin y Huxley y ésta es lo suficientemente grande o constante para que la membrana emita un pulso, el potencial posterior caerá en un sobre impulso negativo el cual impide que se genere un segundo pulso instantáneamente, este efecto propio de las neuronas biológicas reales se conoce como periodo de refracción absoluta y es una de las propiedades del modelo de Hodgkin y Huxley. en la Figura 3.6 se muestra que si el valor de $I(t)$ es menor al valor del umbral de la membrana del modelo matemático ($I_\theta = 6\mu A/cm^2$), el potencial de membrana regresa a su valor de reposo.

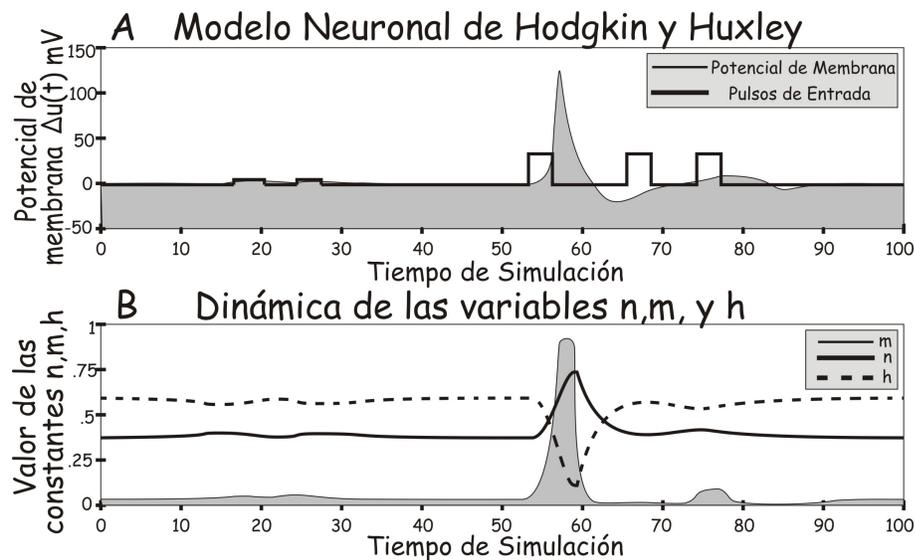


Figura 3.6: **A.** Los 20ms la neurona recibe dos pulsos de entrada de $2\mu A/cm^2$ de amplitud pero no son lo suficientemente fuertes para emitir un pulso. Cuando el tiempo t es igual a los 65ms la neurona recibe un pulso de $7\mu A/cm^2$ de amplitud, lo cual causa una acción potencial (pulso), a los 72ms y 78ms se repite el mismo pulso, ambas veces con la misma magnitud, pero la neurona no dispara, esto debido a la propiedad de refracción absoluta. **B.** Dinámica de las variables m , n , y h de acuerdo al potencial de membrana.

3.2.1.2. El modelo Izhikevich

Izhikevich [31] 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 de primer orden, la

ecuación 3.9 y 3.10

$$\dot{V} = 0.04V^2 + 5V + 140 - U + I \quad (3.9)$$

$$\dot{U} = a(bV - u) \quad (3.10)$$

La variable V representa el potencial de membrana de la neurona y u representa la variable de recuperación de membrana, 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 muestran los parámetros de la ecuación 3.10 [31].

- 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 3.7 los parámetros del modelo neuronal propuesto por Izhikevich pueden ser observados, y en la Figura 3.8 se muestra el potencial de membrana, la variable de recuperación y la señal de entrada.

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 [1], 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 de basan en la sumatoria de todas las contribuciones de las

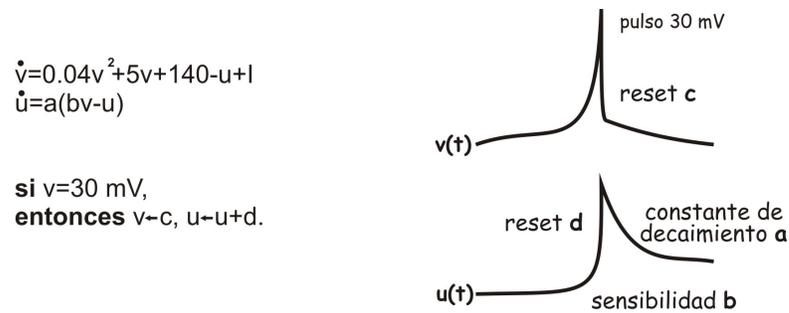


Figura 3.7: Se muestra las ecuaciones diferenciales que describen el comportamiento del modelo de Izhikevich, así como su representación gráfica.

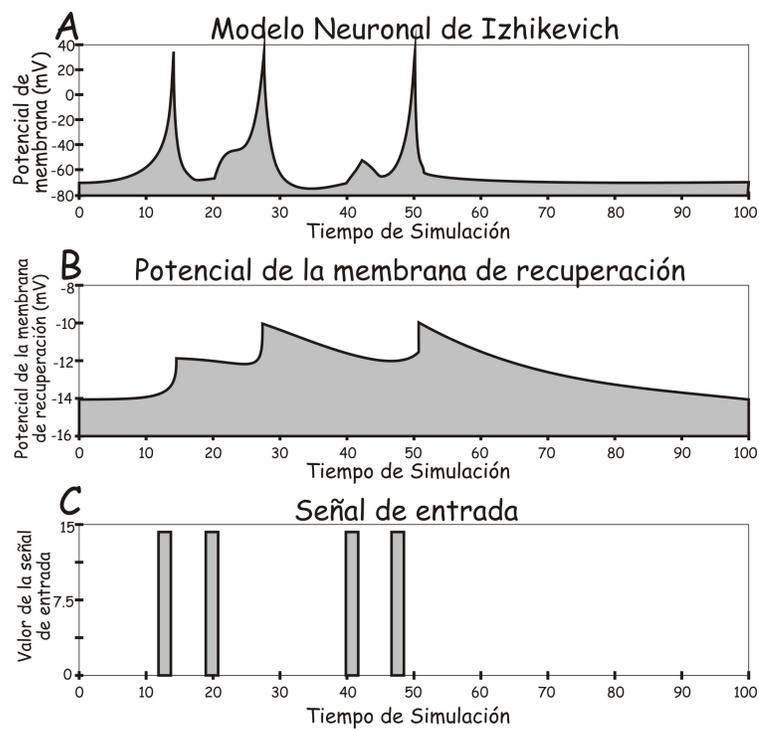


Figura 3.8: En A se presenta la respuesta del potencial de membrana del modelo de redes neuronales de tercera generación propuesto por Izhikevich, en B se grafica la variable de recuperación de la membrana y en C se muestran los pulsos y valor de entrada a la neurona.

neuronas presinápticas al potencial de membrana, y si el potencial de membrana supera el umbral fijado previamente, entonces la neurona disparará.

Este tipo de modelos Integración y disparo se puede dividir en dos submodelos: 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 sólo 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 3.11, la cual 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

$$t^{(f)} : u(t^{(f)}) = \theta \quad \text{y} \quad \left. \frac{du(t)}{dt} \right|_{t=t^{(f)}} > 0 \quad (3.11)$$

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 sólo por sus tiempos de disparo $t^{(f)}$, Figura 3.9.

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 3.10.

La ecuación diferencial que describe este sistema está dada por:

$$V(t) = V_r + \frac{1}{C} \int_{t_0}^t I(t) dt \quad (3.12)$$

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 e $I(t)$ 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 sólo por el switch, sino también por la resistencia, Figura 3.11.

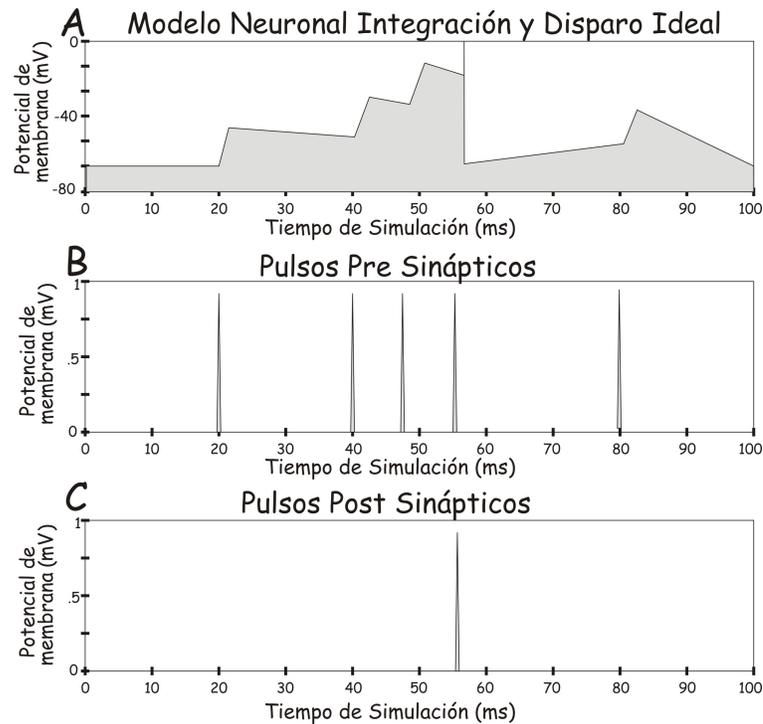


Figura 3.9: Se describe el modelo neuronal de integración y disparo ideal. En la Figura **A** se muestra el potencial de membrana cuyo umbral se fijó en 1. La Figura **B** se muestra los pulsos que recibe la neurona postsináptica de una neurona presináptica. La Figura **C** describe la señal de salida (acción potencial) de la neurona postsináptica bajo la conducta del potencial de membrana Figura **A**.

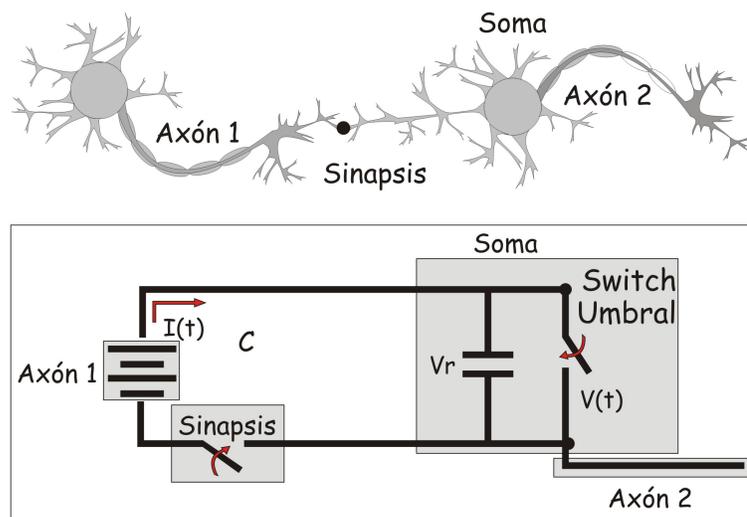


Figura 3.10

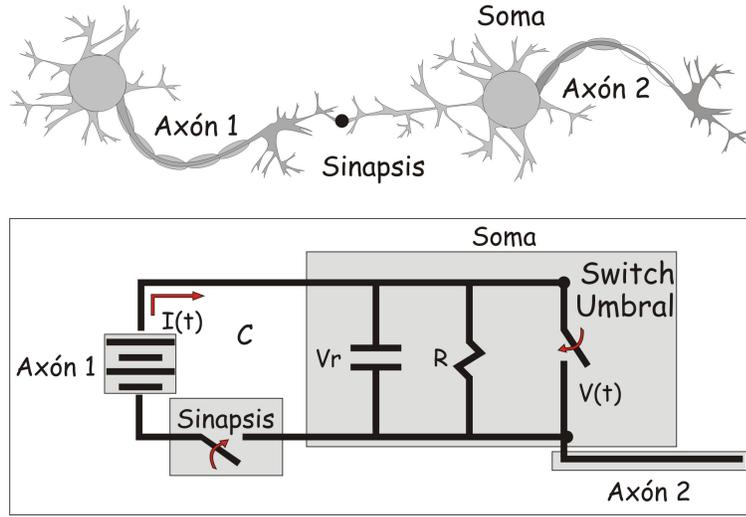


Figura 3.11

La ecuación diferencial que describe este sistema está dada por:

$$V(t) = V_r e^{-\frac{1}{RC}(t-t_0)} + \frac{1}{C} \int_{t_0}^t I(\tau) e^{-\frac{1}{RC}(t-\tau)} d\tau \quad (3.13)$$

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 e, $I(t)$ es el valor de la corriente de la fuente y R es el valor de la resistencia en $m\Omega$.

3.2.2.3. Modelo del pulso de respuesta

El modelo del pulso de respuesta (SRM) fue creado, al igual que los modelos PIF y LIF, para reducir el modelo tetra dimensional de Hodgkin-Huxley a una sola ecuación. Ha sido probado que el modelo SRM puede emular hasta 90 % el comportamiento de un tren de pulsos del modelo de Hodgkin-Huxley [59].

La principal diferencia entre el modelo del pulso de respuesta (SRM) y los modelos de integración y disparo (I&F) es que ambos son descritos por una ecuación diferencial, mientras que en el caso de los modelos SRM el potencial de membrana es descrito por la respuesta de un kernel [62].

$$u_j(t) = \eta(t - \hat{t}_j) + \sum_i w_{ji} \sum_j \varepsilon_{ji}(t - \hat{t}_j, t - t_i^{(f)}) + \int_0^\infty k(t - \hat{t}_j, s) I^{ext}(t - s) ds \quad (3.14)$$

donde $t_j^{(f)}$ constituye los pulsos de emitidos por la neurona presináptica i , mientras que w_{ji} representa a la eficacia sináptica (pesos sinápticos) entre la neurona presináptica i a la neurona postsináptica j , y I^{ext} es la corriente externa (no necesariamente pulsos, sino que comprende también señales analógicas), la variable s es obtenida por $s = t - t_i^{(f)}$ o en el caso donde múltiples retardos son implementados: $s = t - t_i^{(f)} - \Delta^k$ [51],[8],[49]. Finalmente \hat{t}_j es el pulso más reciente emitido por la neurona postsináptica j .

La neurona post sináptica j dispara un pulso cuando el potencial de membrana supera el valor del umbral θ :

$$t = t^{(f)} \Leftrightarrow u_j(t) = \theta \quad \text{y} \quad \frac{du_j(t)}{dt} > 0 \quad (3.15)$$

Usualmente para el modelo SRM se propone un umbral dinámico θ en lugar de fijar uno:

$$\theta = \theta(t - \hat{t}_j) \quad (3.16)$$

3.2.2.4. El kernel

La respuesta del Kernel $\eta(t - \hat{t}_j)$ es el responsable del comportamiento del potencial de respuesta (pulso) emitido por la neurona, esto significa que es directamente responsable de la dinámica del sobre impulso negativo que ocurre después de la emisión del pulso. y es caracterizado por el disparo ocurrido en el tiempo $t_j^{(f)}$ cuando es rebasado el umbral θ .

La conducta del kernel $k(t - \hat{t}_j, s)$ es la respuesta del potencial de membrana a una entrada de corriente basada en el pulso de salida \hat{t}_j , Figura 3.12.

3.2.2.5. SRM simplificado

Una de las más famosas modificaciones del modelo de pulso de respuesta es el SRM_0 [62] [40]. El cual deriva de una respuesta de kernel simplificada, figure 3.13. En SRM_0 el potencial de membrana de la ecuación 3.14 se convierte en:

$$u_j(t) = \eta(t - \hat{t}_j) + \sum_i w_{ji} \sum_{t_i^{(f)}} \varepsilon_0(t - \hat{t}_j^{(f)}) \quad (3.17)$$

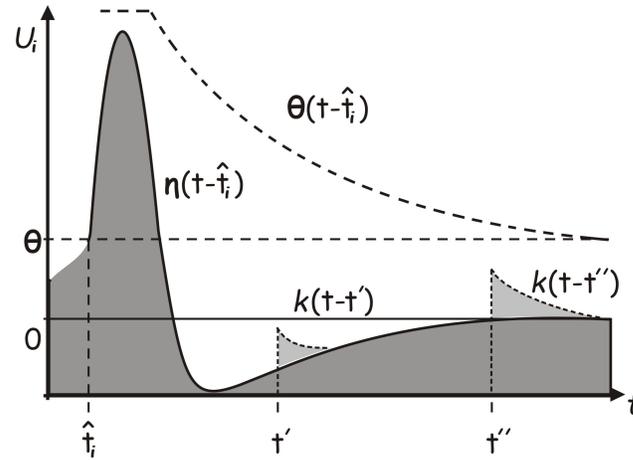


Figura 3.12: La respuesta del kernel $\eta(t-\hat{t}_j, s)$ ocurre cuando el potencial de membrana de la neurona post sináptica cruza el umbral, esta respuesta del kernel puede ser tanto excitatoria (EPSP) o inhibitoria (IPSP)

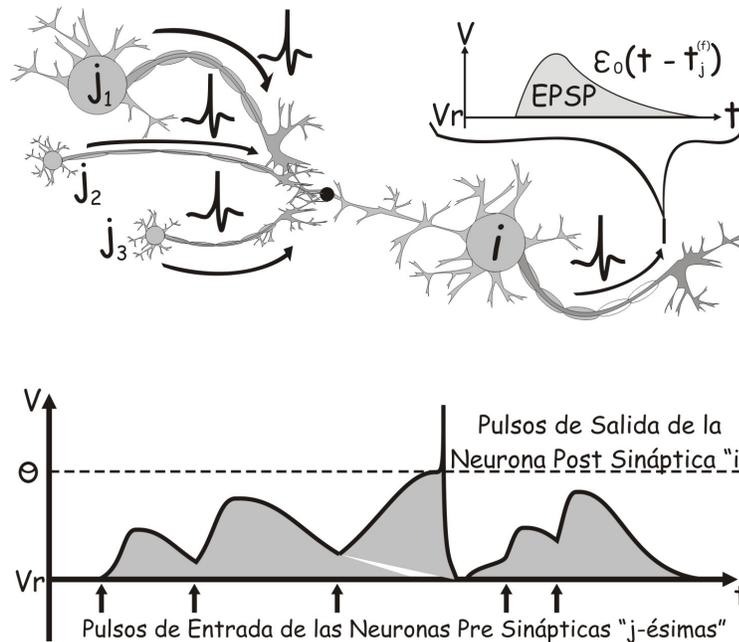


Figura 3.13: Modelo de pulso de respuesta SRM_0 con un umbral constante. La sumatoria de las j -ésimas neuronas causa un pulso post sináptico de excitación (EPSP) $\varepsilon_0(S)$. Cuando el umbral es rebasado, este pulso emitido por la neurona post sináptica i causa un kernel negativo $\eta(S)$ [62].

Tabla 3.2: Comparación de las propiedades computacionales de las redes neuronales de tercera generación. El número de ciclos es una aproximación numérica de operaciones lineales (Suma, Resta Multiplicación, etc.) necesarias para simular al modelo durante una ventana de tiempo de 1ms. Cada casilla vacía indica que el modelo relacionado a esa fila puede caer en cualquiera de los dos casos (si contiene la propiedad o no) pero el autor de dicha representación matemática expresa que esto depende de la elección correcta de las variables intrínsecas del modelo. [32].

Modelos	Biophysically meaningful	Tonic Spiking	Phasic Spiking	Tonic Bursting	Phasic Bursting	Mixed Mode	Spike frequency adaptation	Class 1 excitable	Class 2 excitable	Spike latency	Subthreshold oscillations	Resonator	Integrator	Rebound Spike	Rebound burst	Threshold variability	Bistability	DAP	Accommodation	Inhibition-Indiced spiking	Inhibition-Indiced bursting	Chaos	No. Ciclos
Integración y Disparo	-	+	-	-	-	-	-	+	-	-	-	-	+	-	-	-	-	-	-	-	-	-	5
Integración y Disparo Adaptable	-	+	-	-	-	-	+	+	-	-	-	-	+	-	-	-	-	+	-	-	-	-	10
Integración y Disparo o Explosión	-	+	+	-	+	-	+	+	-	-	-	-	+	+	+	-	+	+	-	-	-	-	13
Resonancia y Disparo	-	+	+	-	-	-	-	+	+	-	+	+	+	+	-	-	+	+	+	-	-	+	10
Integración Cuadrática y Disparo	-	+	-	-	-	-	-	+	-	-	-	-	+	-	-	+	+	-	-	-	-	-	7
Izhikevich (2003)	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	13
FitzHugh - Nagumo	-	+	+	-	-	-	-	+	-	+	+	+	-	+	-	+	+	-	+	+	-	-	72
Hindmarsh-Rose	-	+	+	+	-	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	-	+	120
Morris-Lecar	+	+	+	-	-	-	-	+	+	+	+	+	+	+	+	+	+	+	+	+	-	-	600
Wilson	-	+	+	+	-	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	-	-	180
Hodgkin-Huxley	+	+	+	+	-	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	1200

3.2.3. Comparación de modelos neuronales

El costo computacional es expresado en ciclos (Operaciones Lineales) en una ventana de tiempo. Como se puede observar en la tabla 3.2 el modelo de Hodgkin-Huxley necesita 1200 ciclos por cada milisegundo, lo cual indica que no es viable en aplicaciones de arquitecturas neuronales a gran escala. Por otra parte el modelo de integración y disparo sólo necesita 5 ciclos por mili segundo pero carece de similitud biológica. los modelos SRM y SRM simplificado (SRM_0) no aparecen en esta comparación.

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 sólo 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 [62]. Este fue el principal modelo de codificación neuronal por muchos años.

Thorpe et al. [50] probaron 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. [58], 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 [37],[34] o analógicamente [6]. 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 [62]:

$$v = \frac{\eta_{sp}(T)}{T} \quad (3.18)$$

donde T es la ventana de tiempo, $\eta_{sp}(T)$ es el número de pulsos emitidos en la ventana de tiempo. Existen tres procedimientos para realizar el promedio [62]: Tasa por conteo de pulsos (promediar sobre el tiempo), Tasa por densidad de pulsos (promediar sobre varios trenes de pulso), tasa por actividad de poblaciones (promediar sobre diversas neuronas).

3.3.2. Codificación temporal

En la codificación temporal la información es codificada por pulsos temporales [7]. Hopfield [30] propuso un método de codificar señales analógicas dentro de pulsos temporales con respecto a un diseño oscilatorio. este método ha sido probado experimentalmente en una anguila eléctrica. Además Mass [39] propuso un método para codificar señales analógicas en forma de pulsos temporales. Un diferente método ha

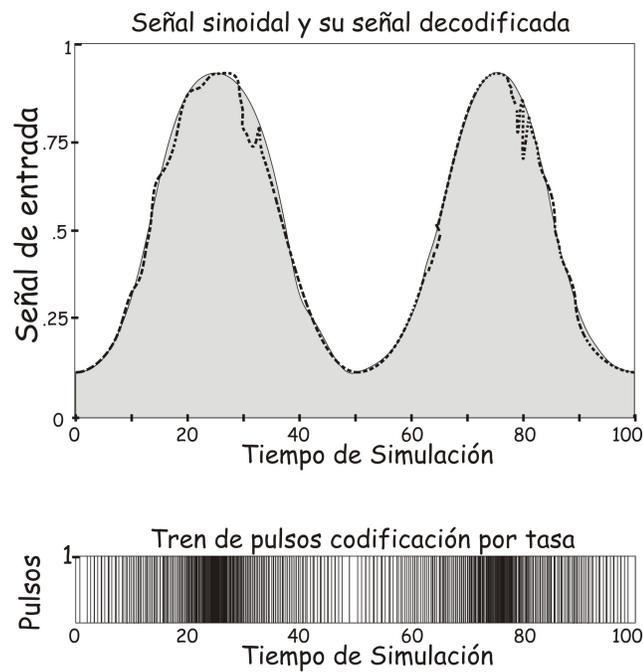


Figura 3.14: Una señal sinoidal se codifica por medio del método de codificación por tasa, en la parte de abajo de la Figura se muestra la codificación en un tren de pulso donde se percibe que en las crestas de la señal hay una mayor densidad de pulsos, posteriormente se decodifica dicho tren de pulsos para obtener una señal (línea punteada) muy parecida a la señal sinoidal original [52].

sido sugerido por Wen y Sendhoff [63], donde las 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. [48] propusieron 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, Figura 3.15.

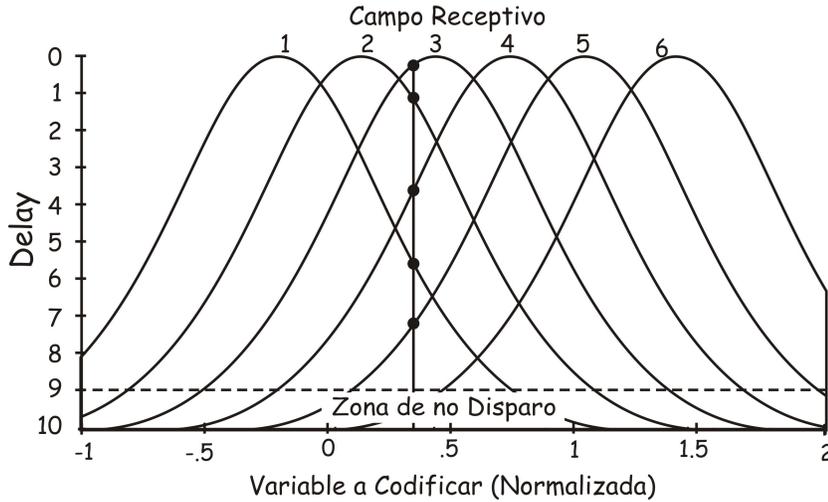


Figura 3.15: Codificación con campos receptivos Gaussianos. El eje horizontal representa la entrada de la variable real a codificar, el eje vertical representa los tiempos para armar el tren de pulsos de la variable codificada.

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 [6]:

$$C_i = I_{min} + \left(\frac{2i - 3}{2}\right) \left(\frac{I_{max} - I_{min}}{m - 2}\right) \quad (3.19)$$

$$\sigma_i = \frac{1}{\gamma} \frac{I_{max} - I_{min}}{m - 2} \quad (3.20)$$

donde σ es una constante usualmente 1.5. Se propone un umbral del cual cualquier valor por encima de éste la neurona no dispara. En el ejemplo de la Figura 3.15, el valor analógico a codificar en disparos temporales en 6 neuronas es de 0.3, lo cual la codificación reslta 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 dispar pues se encuentra arriba del umbral.

3.4. Métodos de aprendizaje

Los pesos w_{ij}^k entre la neurona presináptica i y la neurona postsináptica j 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 no supervisado

En 1949 Hebb formuló la famosa ley de Hebb: “Cuando el axón de la célula A está cerca de excitar a la célula B o repetidamente hace que ésta dispare. Un proceso o cambio metabólico toma lugar en una o ambas células tal que la eficiencia de A al igual que la eficiencia de la célula B incrementa”.

La ley de Hebb es modificada para que los pesos sean adaptados con base a la actividad de la neurona presináptica y postsináptica.

3.4.1.1. Método de Hebb

En la Figura 3.16 se observan los resultados experimentales de Bi y Poo [62] en el hipocampo neuronal.

Más que nada, Bi y Poo [62] encontraron que la eficiencia sináptica Δw_{ij} va directamente relacionada con los tiempos de disparos entre las neuronas presináptica y postsináptica. Esta relación es llamada Plasticidad del Pulso Dependiente del Tiempo (STDP) [16]

La forma de calcular la actualización de los pesos sinápticos ha sido propuesta por

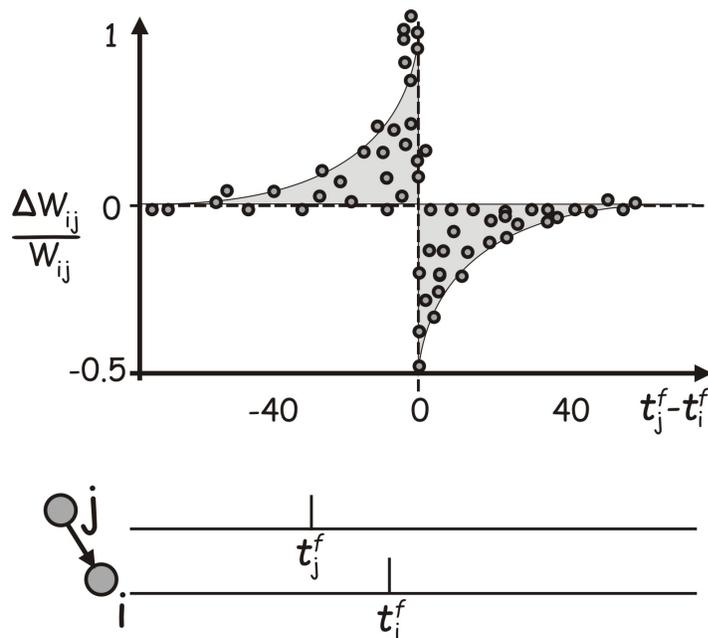


Figura 3.16: Los pesos neuronales entre la neurona pre sináptica j y la neurona post sináptica i solo se modifican si el tiempo de los disparos entre ambas neuronas se encuentran lo suficientemente cerca uno del otro. Estos datos fueron tomados del trabajo de Bi y Poo (1998) [62]. Donde la neurona j es la neurona pre sináptica y la neurona i se conoce como la neurona post sináptica, así como los tiempos t_j^f y t_i^f , representar los tiempos en los que las neuronas pre y post sinápticas respectivamente disparan.

Gestner et al. [62] con la implementación de una ventana de aprendizaje:

$$\Delta w = \begin{cases} A_+ \exp(s/\tau_1) & \text{para } s < 0 \\ A_- \exp(s/\tau_2) & \text{para } s > 0 \end{cases} \quad (3.21)$$

donde $s = t_j^f - t_i^f$ es la diferencia de tiempo entre los disparos de la neurona pre sináptica y la neurona post sináptica. El τ_1 y τ_2 son constantes, las cuales dictan la tasa de decaimiento o saturación de la función de aprendizaje. A_+ y A_- son implementadas para tener inferencia directamente en la tasa de aprendizaje (Velocidad con la que aprenden las neuronas), esta ventana de aprendizaje se puede observar en la Figura 3.17

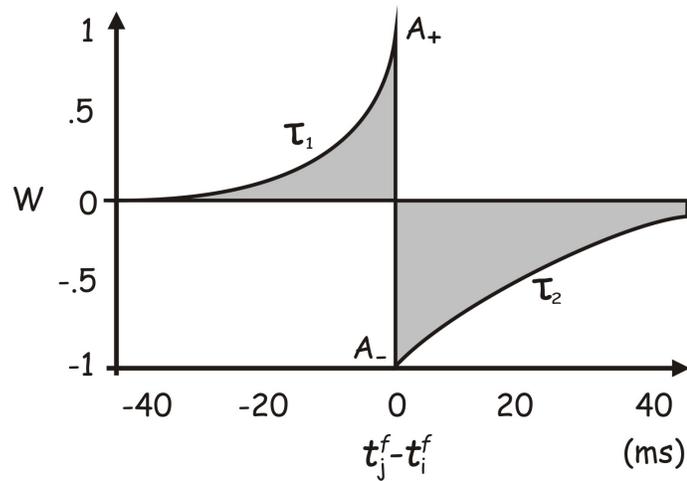


Figura 3.17: La ventana de aprendizaje exponencial propuesta por Hebb basada en la teoría de Bi y Poo, donde la modificación de los pesos sinápticos va directamente relacionados con la diferencia que existe entre los pulso pre y post sinápticos, y donde los valores de A_+ , A_- , τ_1 y τ_2 están dados por 1, -1, 10 y 20 respectivamente.

Numerosos métodos se han propuesto con el propósito de satisfacer la necesidad de obtener los valores máximos y mínimos de los pesos en aprendizaje no supervisado, pues esto es un problema muy común pues valores muy altos inestabilizarían las neuronas y por el contrario, tasas de aprendizaje pequeñas, harían que la salida de las neuronas tarde demasiado en converger.

Uno de estos métodos de entrenamiento es el Entrenamiento por Asociación de Pesos Sinápticos (SWAT) [33]. En el método de SWAT la STDP es combinada con el método de Bienenstock-Cooper-Munro (BCM), donde un umbral descendiente es usado para estabilizar los pesos. Este modelo usa una arquitectura de alimentación hacia adelante similar a los hipocampos donde una sinapsis de excitación (EPSP) y

sinapsis de inhibición (IPSP) se encuentran entre neuronas presinápticas y postsinápticas.

3.4.1.2. Método del Hebbiano local

El método del Hebbiano Local con aprendizaje dependiente de retardos temporales, se basa en la regla Winner-Take-All [24], donde múltiples sub-sinapsis con múltiples retardos (Retardos de los Pulsos Neuronales) son implementados, tal como en la Figura 3.4.

Este método de entrenamiento no supervisado es muy popular en tareas de clasificación [48],[6],[35]. La regla de aprendizaje que se implementa es una función Gaussiana $L(\Delta t)$, que toma como entrada (al igual que la regla de Hebb) la diferencia de los pulsos de la neurona presináptica y postsináptica, para actualizar los pesos, las ecuaciones 3.22-3.24, muestran como se realiza el aprendizaje, donde cada salida de la neurona presináptica es el centro de una función de base radial (RBF), [6],[48].

$$\Delta w_{ij}^k = \eta L(\Delta t_{ij}) \quad (3.22)$$

$$L(\Delta t) = (1 + b) \exp \left[\frac{(\Delta t - c)^2}{2(k - 1)} \right] - b \quad (3.23)$$

$$k = 1 - \frac{v^2}{2 \ln \left[\frac{b}{1+b} \right]} \quad (3.24)$$

donde Δw_{ij} representa la amplitud del cambio de los pesos, $L(\Delta t_{ij})$ es la función de aprendizaje, η es la tasa de aprendizaje, v es el ancho de la ventana de aprendizaje, Δt es la diferencia entre el disparo de la neurona presináptica j y la neurona postsináptica i , la constante b es la actualización negativa de los pesos y c es donde se localiza el centro de la función de aprendizaje [6], Figura 3.18

3.4.2. Aprendizaje supervisado

Como se ha descrito en las secciones previas, el comportamiento de las redes neuronales de tercera generación se asemeja 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

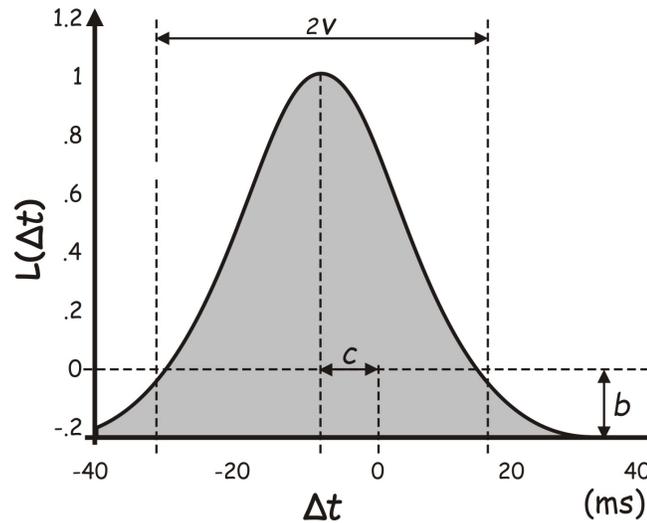


Figura 3.18: Se muestra la función de aprendizaje del método de Hebb. Donde el eje de las ordenadas es la regla de aprendizaje ($L\Delta t$) y el eje de las abscisas representa la diferencia temporal entre disparos Δt , los valores constantes \mathbf{b} , \mathbf{c} y \mathbf{v} son 0.2, -2.3 y -5 respectivamente. Y al igual que en la sección anterior el peso máximo y mínimo se encuentran acotados, 0 y W_{MAX} [48],[6]

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.

3.4.2.1. Método SpikeProp

Bohte et al. [51] propusieron 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, Figura 3.4.

La arquitectura que necesita este entrenamiento es de tipo feedforward, 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 con una capa oculta. \mathbf{H} es la capa de entrada, \mathbf{I} es la capa oculta y \mathbf{J} es la capa de salida.

Bohte et al. [51] implementaron el modelo del pulso de respuesta (SRM) para describir el potencial de membrana de cada pulso emitido por las neuronas, el kernel

asociado cada SRM es sin el fragmento de refracción de la ecuación, esto quiere decir que la ecuación 3.17 queda de la siguiente forma:

$$u_j(t) = \sum_{i \in \Gamma_j} \sum_{k=1}^m w_{ij}^k \varepsilon(t - t_i - d^k) \quad (3.25)$$

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:

$$E = \frac{1}{2} \sum_{j \in J} (t_j^a - t_j^d)^2 \quad (3.26)$$

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:

$$\Delta w_{ij}^k = -\eta y_i^k(t_j^a) \delta_j \quad (3.27)$$

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

$$\delta_j = \frac{(t_j^a - t_j^d)}{\sum_{i \in \Gamma_j} \sum_l w_{ij}^l \left(\frac{\partial y_i^l(t_j^a)}{\partial t_i^a} \right)} \quad (3.28)$$

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

$$\Delta w_{hi}^k = -\eta y_h^k(t_i^a) \delta_i = -\eta \frac{y_h^k(t_i^a) \sum_j \left[\delta_j \sum_k w_{ij}^k \left(\frac{\partial y_i^k(t_j^a)}{\partial t_i^a} \right) \right]}{\sum_{n \in \Gamma_i} \sum_l w_{ni}^l \left(\frac{\partial y_n^l(t_i^a)}{\partial t_i^a} \right)} \quad (3.29)$$

donde δ_i es:

$$\delta_i = \frac{\sum_{j \in \Gamma_i} \left[\delta_j \sum_k w_{ij}^k \left(\frac{\partial y_i^k(t_j^a)}{\partial t_i^a} \right) \right]}{\sum_{h \in \Gamma_i} \sum_l w_{hi}^l \left(\frac{\partial y_h^l(t_i^a)}{\partial t_i^a} \right)} \quad (3.30)$$

La prueba matemática de este algoritmo se puede encontrar en la referencia [14], Finalmente el método SpikeProp puede ser desarrollado en la siguiente tabla 3.3.

Tabla 3.3

Algoritmo SpikeProp

 Se calcula δ_j para todas las salidas, de acuerdo a 3.28

 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 3.30

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

 adapta w_{ij}^k utilizando $\Delta w_{ij}^k = -\eta y_i^k(t_j) \delta_j$ 3.27,

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

 calcular w_{hi}^k utilizando $\Delta w_{ij}^k = -\eta y_i^k(t_j) \delta_j$ 3.29.

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

3.4.2.2. Regla modificada de Widrow-Hoff o regla delta generalizada a spikes

Similar a otros algoritmos de entrenamiento supervisado, los pesos sinápticos de la red neuronal se ajustan iterativamente basados en una relación de entrada-salida deseada [3]. Se comienza a partir de la regla Widrow-Hoff común para la modificación del peso de una sinapsis i :

$$\Delta \mathbf{w}_i^{WH} = \lambda \mathbf{x}_i (\mathbf{y}_d - \mathbf{y}_{out}) \quad (3.31)$$

donde $\lambda \in \mathbb{R}$ es un tipo de aprendizaje positivo con valores reales, \mathbf{x}_i es la entrada a través de la neurona presináptica por medio de de la sinapsis i y finalmente \mathbf{y}_d y \mathbf{y}_o se refieren a la salida deseada y la salida real emitida por la neurona respectivamente. Mediante la integración de $\Delta \mathbf{w}_i^{WH}$ se obtiene una $\Delta \mathbf{w}_i$ escalar que se utiliza para actualizar el peso de la sinapsis i :

$$\Delta w_i = \lambda \int x_i(t) (y_d(t) - y_o(t)) dt \quad (3.32)$$

Los pesos se actualizan en un proceso iterativo, usando: $\mathbf{w}_i(\mathbf{e} + 1) = \mathbf{w}_i(\mathbf{e}) + \Delta \mathbf{w}_i$, donde \mathbf{e} es la época actual del proceso de aprendizaje.

Como se muestra en la Figura 3.19A, existe un tren de pulsos de entrada, provenientes de la neurona presináptica, en la Figura 3.19C se muestra la transformación de cada pulso al kernel, similar a la transformación de distancia métrica usada para

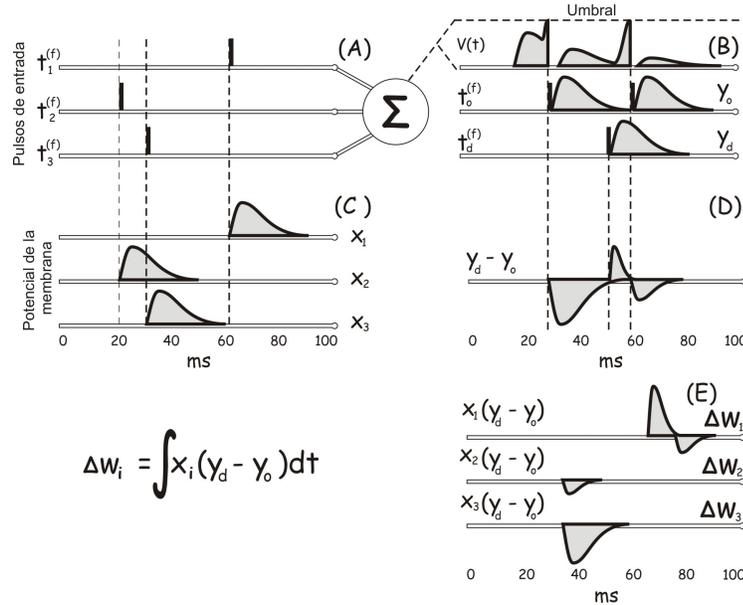


Figura 3.19: Ilustración del algoritmo de entrenamiento “Relación de trenes de pulsos entrada salida”

comparar trenes de pulsos [41]:

$$\alpha(t) = \exp\left(\frac{1}{\tau_s} t\right) \exp\left(-\frac{t}{\tau_s}\right) \Theta \quad (3.33)$$

donde $\Theta(t)$ se refiere a la función de Heaviside y τ_s es el tiempo donde ocurrirá el valor máximo de la función sináptica, este conjunto de señales transformadas entra a la neurona postsináptica donde el potencial de membrana se incrementa como se aprecia en la Figura 3.19B, dicho estímulo provoca la excitación de la neurona postsináptica dando como resultado la generación de dos pulsos de salida en los instantes $t_o^{(0)}$ y $t_o^{(1)}$, respectivamente, ninguno de ellos es igual al pulso temporal deseado $t_d^{(0)}$, esto se aprecia mejor en la Figura 3.19D que representa el error entre la referencia deseada y la señal de salida, dado (3.34).

$$E = \int |y_d(t) - y_o(t)| dt \quad (3.34)$$

Finalmente para completar la actualización de los pesos se aplica la ecuación (3.32) como se muestra en la Figura 3.19E.

3.5. Comentarios

En este capítulo se mostraron las bases de las redes neuronales de tercera generación, esto nos servirá para comprender el cómo se diseña y realiza el control basado en este tipo de sistemas. Así como para entender qué tipo de modelos neuronales, codificación y entrenamiento son los que mejor se acoplan para desempeñar un control adecuado.

Capítulo 4

Resultados

En el presente capítulo se muestra la fusión entre las redes neuronales de tercera generación y el área de control, los resultados obtenidos en simulación numérica sobre un modelo lineal de primer y segundo orden, de igual forma se muestran los resultados obtenidos de manera experimental con un motor de CD, todas estas pruebas son realizadas con una combinación de diferentes modelos neuronales con dos diferentes tipos de entrenamientos.

4.1. Introducción

Los controladores más utilizados actualmente en la industria son los controladores PID (Aström, 1997) o alguna de sus modificaciones, dada la facilidad de su implementación y los recursos tecnológicos que requiere. Sin embargo, el algoritmo PID lineal presenta limitaciones de robustez y convergencia de utilizar cuando el proceso a controlar presenta dinámicas complejas, zonas muertas y características altamente no lineales. El funcionamiento del controlador PID en general se basa en actuar en forma proporcional, integral y derivativa sobre la señal de error $e(t)$, definida como la diferencia entre la señal de referencia $Yd(t)$ y la señal de salida del proceso $y(t)$, con la finalidad de acondicionar la señal de control $u(t)$ a partir de la suma de estas tres acciones de control teniendo efectos de convergencia a la salida de la planta como se aprecia en la Figura 4.1, donde k_p , k_i y k_d son las ganancias del controlador a ser sintonizadas. Existen distintas técnicas analíticas y experimentales con el fin de sintonizar esas ganancias (Aström and Hägglund, 2006). Una alternativa es auto-sintonizar en línea dichas ganancias como en: (Ramos-Velasco et al., 2013), donde utilizan redes neuronales artificiales *wavenet* para identificar la planta y calcular esas ganancias, dichos artículos consideran sólo sistemas dinámicos SISO. La propuesta de control descrita en este trabajo es una técnica alternativa a las ya existentes, en el

cual se hace uso de redes neuronales de tercera generación para llevar a cabo la tarea de control, sin hacer uso de su modelo matemático ni de los parámetros del mismo, se presentan resultados en simulación numérica y de laboratorio para el seguimiento de trayectorias.

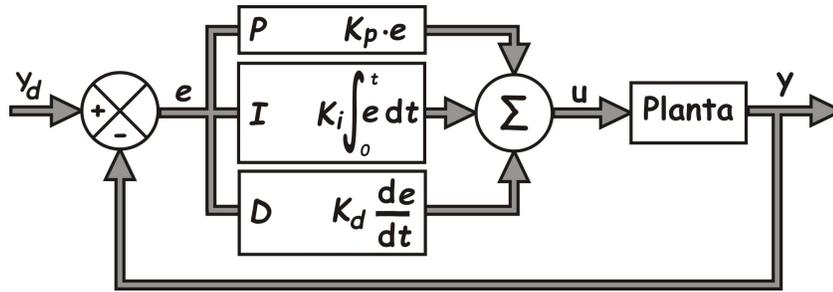


Figura 4.1: Esquema de control PID clásico.

Los sistemas físicos son inherentemente no lineales, así que el control adecuado para estos debe ser un control no lineal [56], pueden ser difícil tratar con este tipo de sistemas pues un control no adecuado puede convertirlo en un sistema no estable en el sentido de Lyapunov y por el contrario, si se diseña un buen control, sistemas controlables no estables en el sentido de Lyapunov pueden ser llevados a la referencia deseada sin mayor problema [13].

Las redes neuronales artificiales de tercera generación son los modelos que mejor emulan el comportamiento de las redes neuronales biológicas, este tipo de modelos tienen un sustento matemático basado en ecuaciones diferenciales en su mayoría no lineales [62], por lo que surgió la idea de fusionar las redes neuronales de tercera generación con los sistemas dinámicos que se desean controlar, y de esta forma obtener intrínsecamente un nuevo modelo el cual será regido por los entrenamientos supervisados.

La aplicación de las redes neuronales artificiales de pulso asume:

- Primeramente la definición de un modelo matemático que emule el comportamiento de las neuronas biológicas, un modelo en el que se encuentre presente la idea de potencial de neurona.
- La definición de una codificación de información del mundo externo en pulsos.
- La definición de una arquitectura de red de neuronas.

- La definición de un algoritmo de entrenamiento apropiado a las necesidades del diseñador.

De acuerdo con esto, este capítulo se encuentra organizado de la siguiente manera: en la Sección 4.2 se muestra el esquema general propuesto del controlador PID-S, en la Sección 4.3 son presentados los resultados obtenidos en simulación numérica trabajando con un sistema lineal de primer orden, con dos métodos de entrenamiento., el entrenamiento SpikeProp y la Regla Modificada de Widrow-Hoff, en la Sección 4.4 se realizan los mismos experimentos que en la sección 4.3, solo que ahora el esquema PID-S se implementa en un motor de corriente directa. Finalmente en la sección 4.5 se presentan las conclusiones de esta tesis, así como el trabajo a futuro sobre el tema de redes neuronales de tercera generación y sus aplicaciones en el área de control.

4.2. PID-S

Los modelos matemáticos que representan el comportamiento de las neuronas biológicas son en su mayoría formulaciones no lineales [53], y los algoritmos de aprendizaje supervisado existentes [40, 44] funcionan sin mayores problemas con ellos. De acuerdo con esto, surgió la idea de controlar sistemas dinámicos con entrenamientos supervisados, diseñados exclusivamente para redes neuronales de tercera generación. Ciertamente los entrenamientos no fueron diseñados para el control de sistemas dinámicos, para esto se contemplan las propiedades del control lineal PID clásico [43], dando como resultado el esquema de control PID-S mostrado en la Figura 4.2, cuyo esquema de control es una de las contribuciones originales de esta tesis.

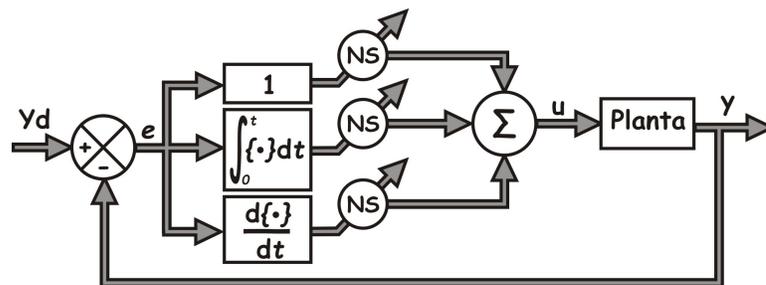


Figura 4.2: Esquema de control PID-S

De la Figura 4.2 se puede observar que la entrada a cada una de las tres neuronas es la señal de error, la integral del error y la derivada del error, respectivamente. Un análisis más detallado del proceso de codificación con múltiples retardos y decodificación se muestra en la Figura 4.3. La codificación que se utiliza para relacionar las

redes neuronales de tercera generación con el sistema a controlar, está definida por la interpolación lineal [47, 64],

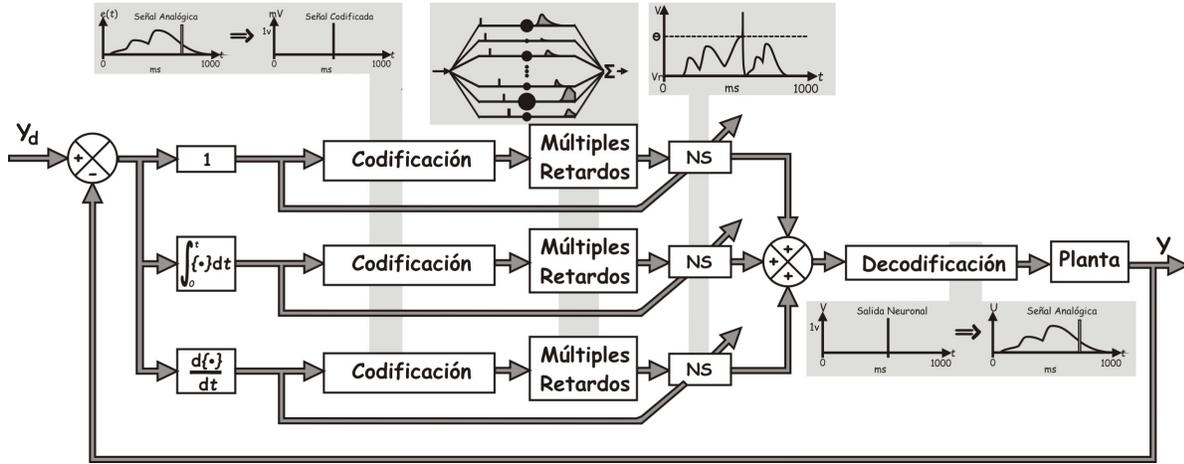


Figura 4.3: Diagrama a bloques del proceso de codificación y decodificación empleadas en el algoritmo de control.

$$t_i(x_e) = t_{max} - \text{round} \left(t_{min} + \frac{(x_e - x_{emin})(t_{max} - t_{min})}{(x_{emax} - x_{emin})} \right) \quad (4.1)$$

donde *round* es la función de redondeo realizada en los intervalos de las ventanas de tiempo, dichas ventanas son definidas por el diseñador tomando en cuenta qué tan suave se quiere la señal de control y con qué hardware se cuenta para la realización de los cálculos, además $t_i(x_e)$ es el instante de tiempo donde se emite el pulso dentro de la ventana neuronal, t_{min} y t_{max} son el mínimo y máximo valor de la ventana de tiempo neuronal respectivamente, y los valores x_e , x_{emin} y x_{emax} son el valor actual, mínimo y máximo del error, integral del error y derivada del error para cada uno de los bloques de codificación correspondiente.

La ecuación que corresponde a la decodificación de la salida neuronal a la señal de entrada a la planta es [64]:

$$u_i = u_{min} + \frac{(t_{max} - t_{min} - t_i)(u_{max} - u_{min})}{(t_{max} - t_{min})} \quad (4.2)$$

donde u_i , u_{min} y u_{max} son el valor actual, mínimo y máximo de la señal de control que se le aplica a la planta.

Algo importante de resaltar es que antes de comenzar la secuencia de control, la salida de la neurona debe inicializarse en el cero codificado, es decir la señal de control

inicial emitida por cada una de las tres neuronas debe ser cero.

4.3. Resultados en simulación numérica

Para probar la eficiencia del control PID-S, se experimentó bajo simulación con diferentes sistemas, modelos neuronales y aprendizajes supervisados. Con una referencia que depende del tiempo, descrita por la ecuación:

$$Yd(t) = \begin{cases} 800 \frac{rad}{seg} & \text{si } 0 < t \leq 300 \\ 100 \frac{rad}{seg} & \text{si } 300 < t \leq 600 \\ 400 \frac{rad}{seg} & \text{si } 600 < t \leq 1000 \end{cases} \quad (4.3)$$

4.3.0.3. Sistema lineal de primer orden

Primeramente, se propone un motor de CD para fines de experimentación, donde la representación entrada-salida está dada por la función de transferencia:

$$\frac{Y(S)}{G(S)} = \frac{83.3}{9.98S + 0.997} \quad (4.4)$$

dicha función es obtenida experimentalmente aplicando una entrada de 4 volts al motor de CD, como se puede observar el modelo propuesto corresponde a un sistema lineal de primer orden estable, y aunque físicamente un motor de corriente directa es un sistema no lineal, para efectos de simulación se puede aproximar a un sistema de primer orden despreciando los efectos inductivos-inerciales.

1. Regla modificada de Widrow-Hoff

En las Figuras 4.4, 4.5 y 4.6 se muestra la respuesta del sistema utilizando el control PID-S bajo el entrenamiento nombrado Regla Modificada de Widrow-Hoff y el modelo PIF, LIF e Izhikevich respectivamente, los valores de entrenamiento de cada trío de neuronas en ambas pruebas son 1, 0.05 y 0.001 para las tres neuronas PIF, 1, 0.05 y 0.001 para las tres neuronas LIF, y 0.01, 0.001 y 0.00001 para las tres neuronas Izhikevich entrenadas con e , $\frac{de}{dt}$ y $\int edt$ respectivamente cada uno de los modelos.

En la Figura 4.7 se muestra la ley de control emitida por dos de los modelos neuronales de tipo umbral y disparo con los mayores niveles de abstracción, y aunque pertenecen al mismo grupo de modelos neuronales el comportamiento

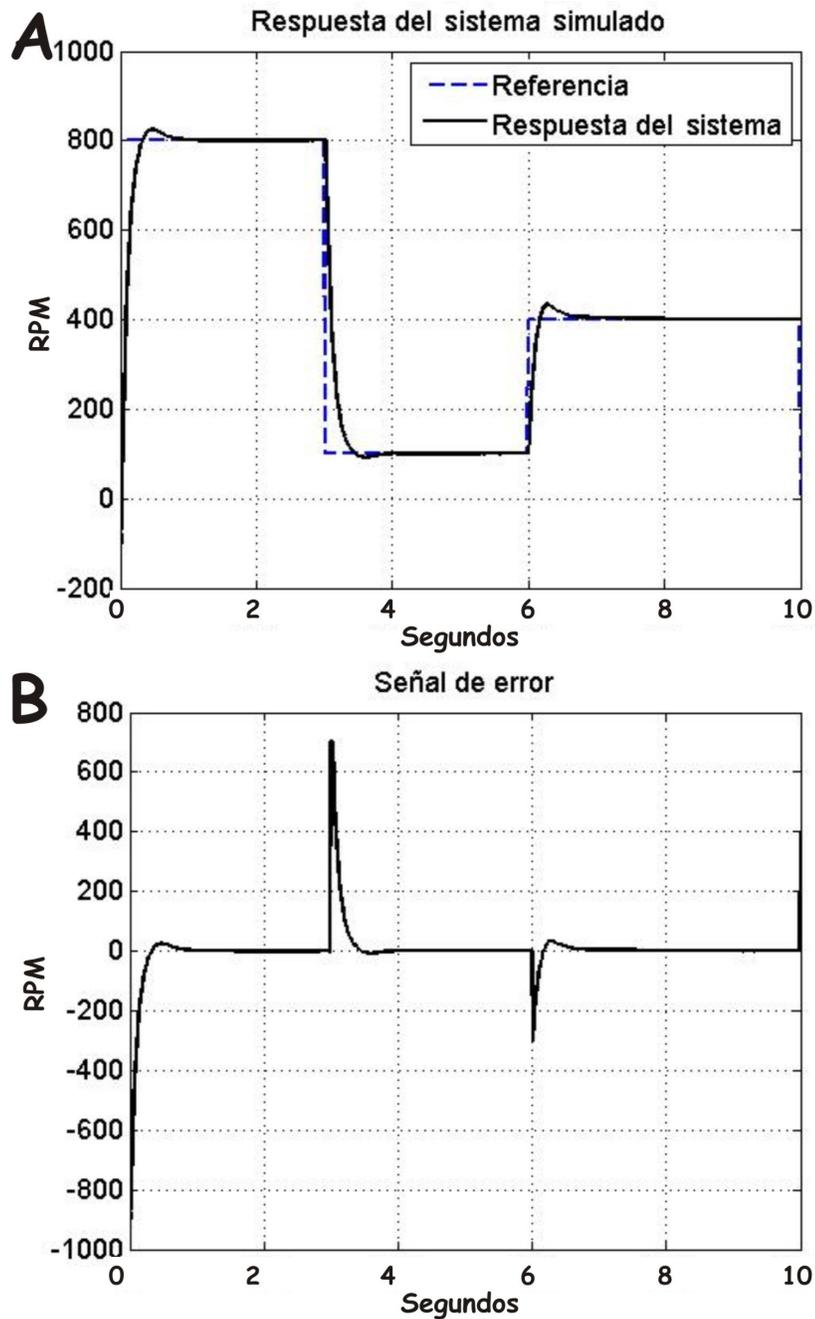


Figura 4.4: En la Figura **A** se describe la respuesta del sistema al control PID-S, la Figura **B** muestra el error de la velocidad del motor con neuronas tipo PIF, entrenadas con el método llamado Regla Modificada de Widrow-Hoff.

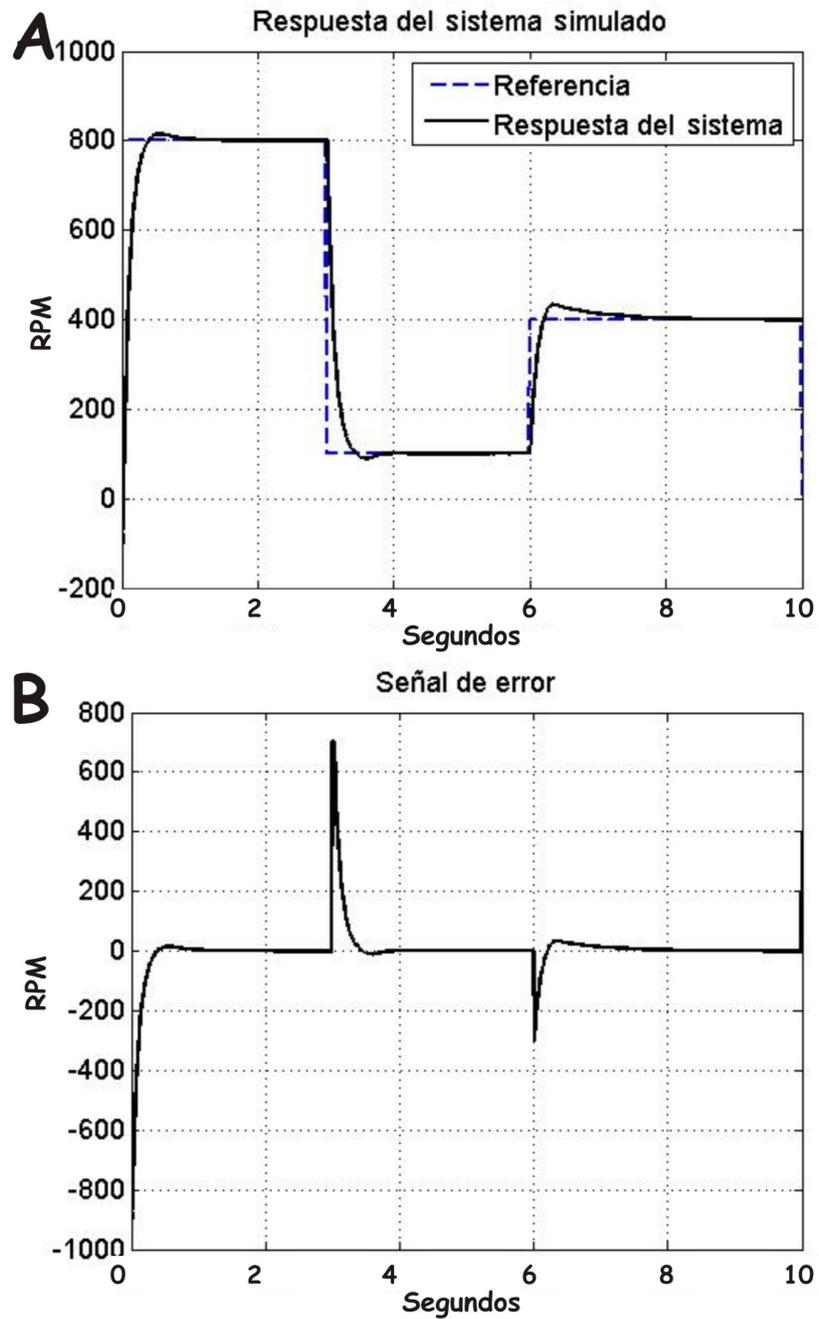


Figura 4.5: En la Figura **A** se describe la respuesta del sistema al control PID-S, la Figura **B** muestra el error de la velocidad del motor con neuronas tipo LIF, entrenadas con el método de la Regla Modificada de Widrow-Hoff.

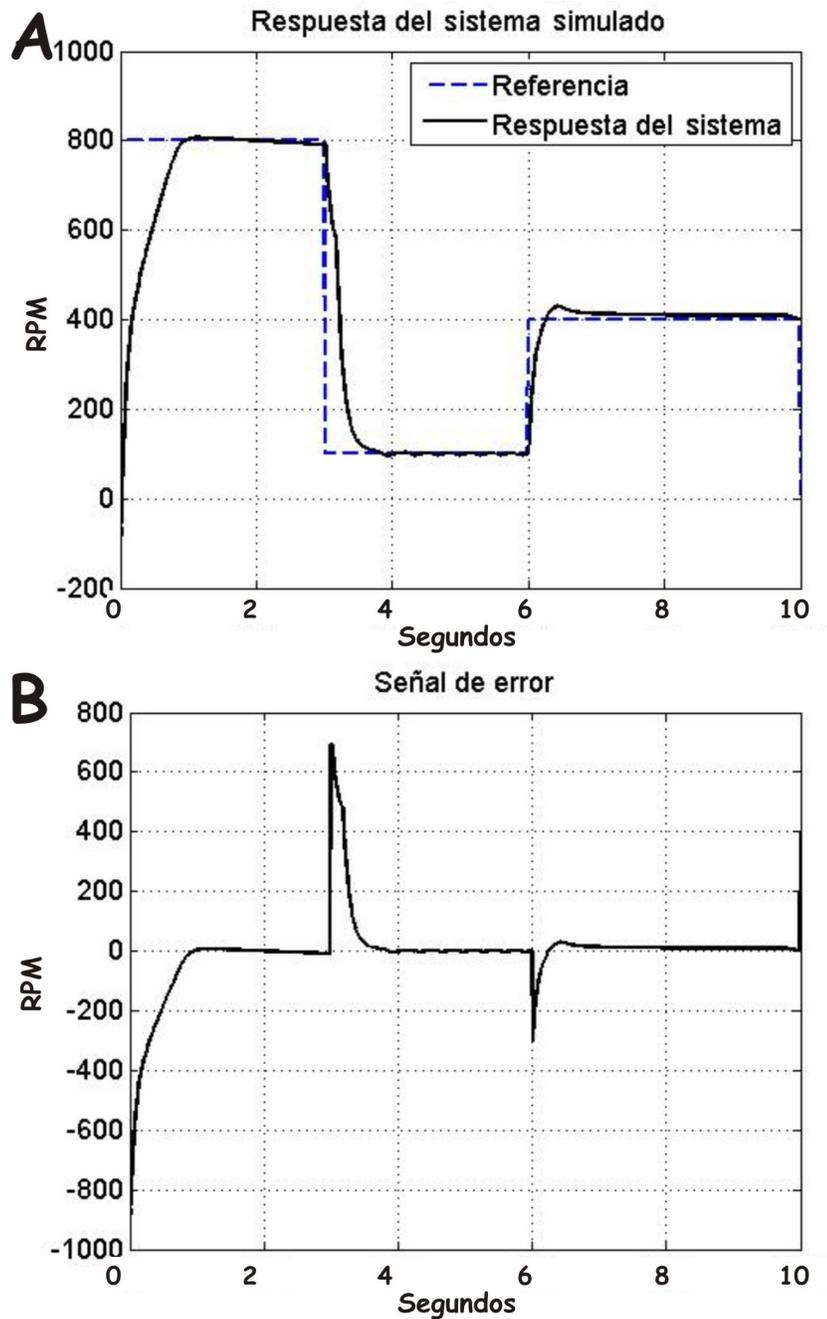


Figura 4.6: En la Figura **A** se describe la respuesta del sistema al control PID-S, la Figura **B** muestra el error de la velocidad del motor con neuronas tipo Izhikevich, entrenadas con el método de la Regla Modificada de Widrow-Hoff.

es diferente entre sí.

En la Figura 4.8 se muestra la ley de control emitida por el esquema PID-S con neuronas Izhikevich, este tipo de modelo neuronal tiene un gran parecido a las neuronas biológicas reales, estrictamente al de las neuronas de la corteza cerebral de las ratas.

2. Entrenamiento SpikeProp

En las Figuras 4.9, 4.10 y 4.11 se muestra la respuesta del sistema utilizando el control PID-S bajo el entrenamiento SpikeProp y el modelo PIF, LIF e Izhikevich respectivamente, los valores de entrenamiento de cada trío de neuronas en ambas pruebas son -0.2, -0.01 y -0.001 para las tres neuronas PIF, -0.1, -0.01 y -0.001 para las tres neuronas LIF, y -0.05, -0.0008 y -0.00001 para las tres neuronas Izhikevich entrenadas con e , $\frac{de}{dt}$ y $\int edt$ respectivamente cada uno de los modelos.

En la Figura 4.12 se muestra la ley de control emitida por dos de los modelos neuronales de tipo umbral y disparo con los mayores niveles de abstracción, y aunque pertenecen al mismo grupo de modelos neuronales el comportamiento es diferente entre sí.

En la Figura 4.13 se muestra la ley de control emitida por el esquema PID-S con neuronas Izhikevich, este tipo de modelo neuronal tiene un gran parecido a las neuronas biológicas reales, estrictamente al de las neuronas de la corteza cerebral de las ratas.

4.3.1. Sistema no lineal de segundo orden

Una de las plantas más implementadas en el área de control es el péndulo invertido, Figura 4.14, del cual la ecuación diferencial que describe su comportamiento está descrita por

$$F = J\ddot{\theta} + \beta\dot{\theta} + lmgsin(\theta) \quad (4.5)$$

donde F es el par aplicado al eje de rotación del péndulo, $J = 1$ es la inercia de éste, $\beta = 0.1$ representa el coeficiente de fricción, $l = 0.3mt$ es la longitud del eje de rotación del péndulo a su centro de masa, $m = .3kg$ constituye la masa y $g = 9.81\frac{m}{s^2}$ describe a la fuerza de gravedad.

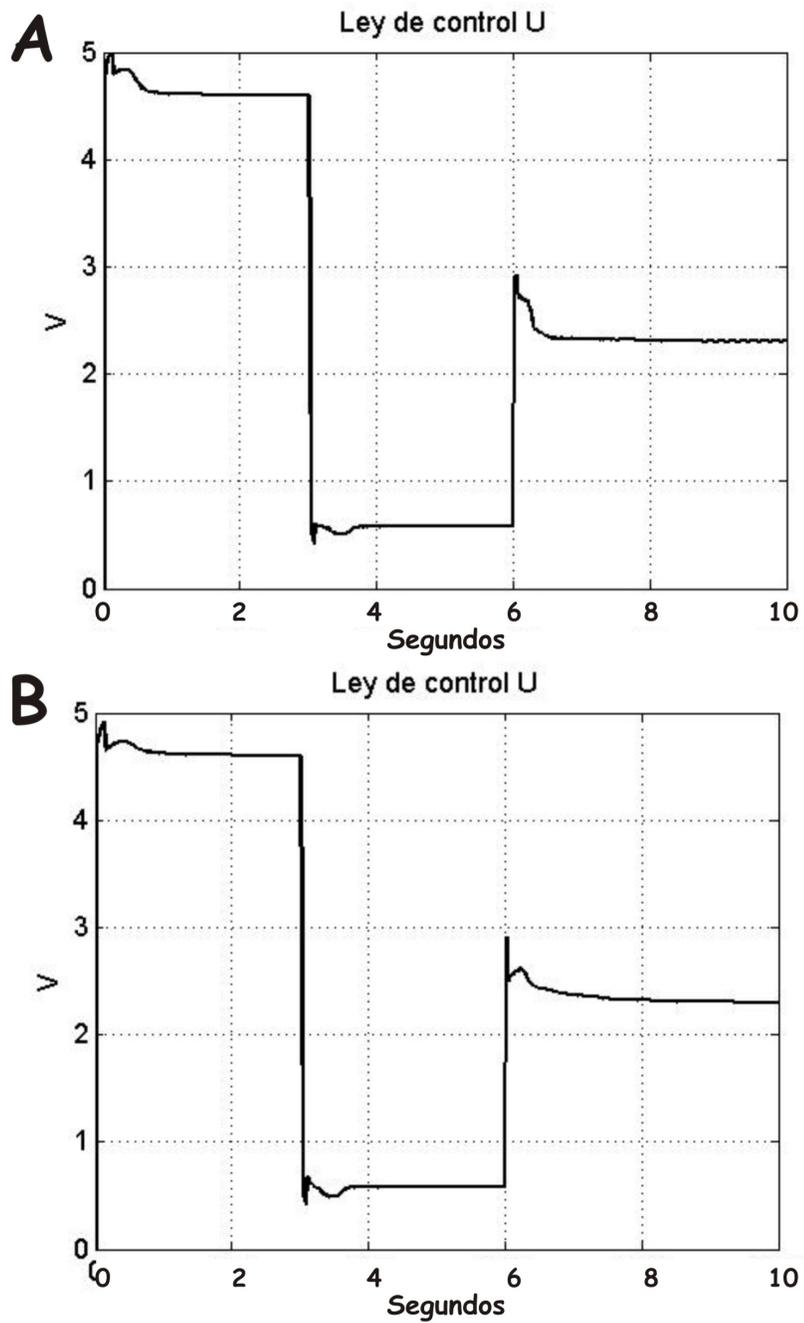


Figura 4.7: En la Figura **A** se describe la ley de control PID-S con neuronas PIF, la Figura **B** muestra la ley de control PID-S con neuronas LIF, entrenadas con el método Regla Modificada de Widrow-Hoff.

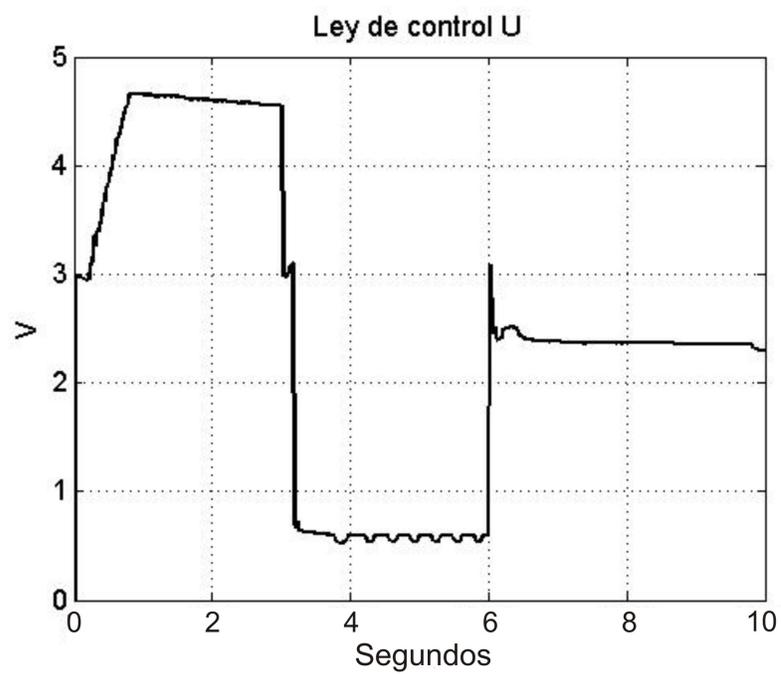


Figura 4.8: En la figura se describe la ley de control PID-S con neuronas Izhikevich, entrenadas con el método Regla Modificada de Widrow-Hoff.

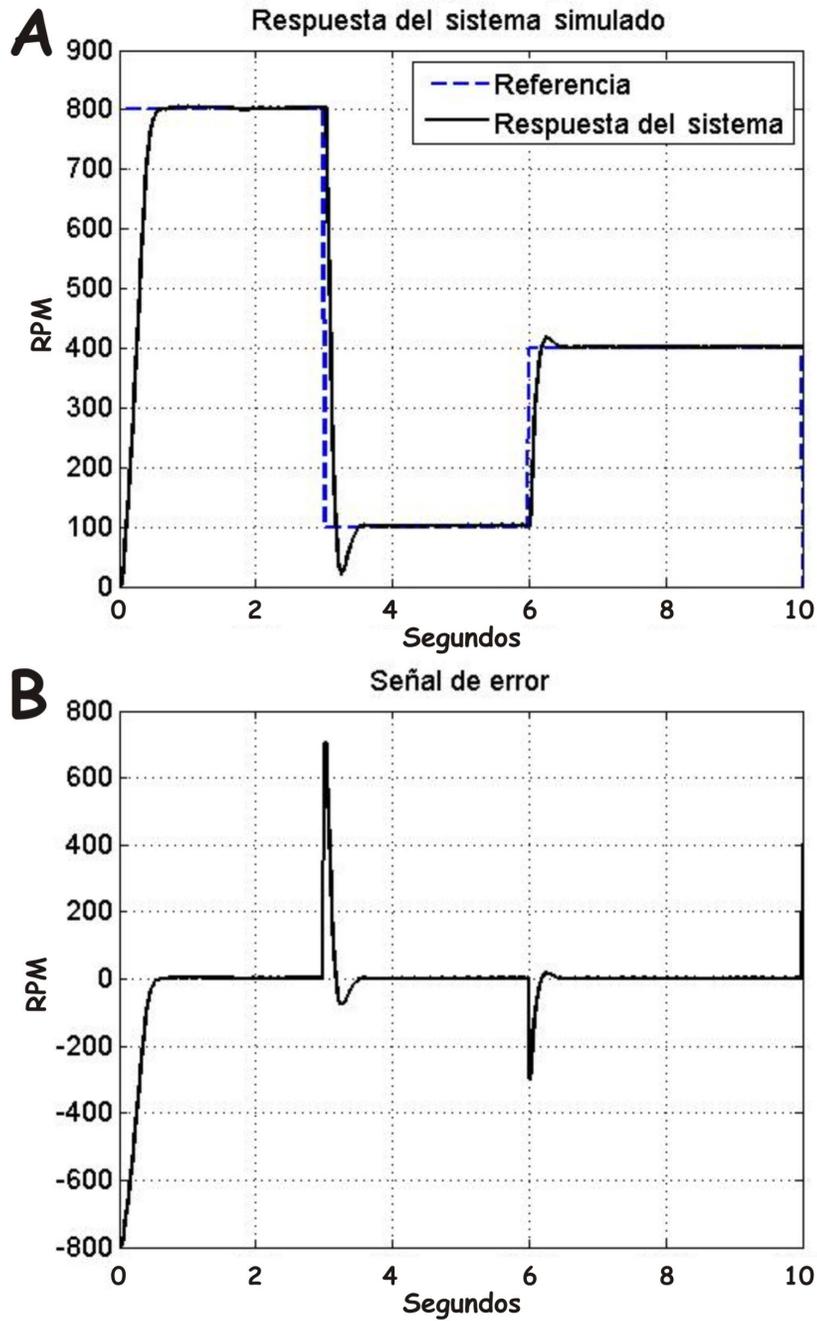


Figura 4.9: En la Figura **A** se describe la respuesta del sistema al control PID-S, la Figura **B** muestra el error de la velocidad del motor con neuronas tipo PIF, entrenadas con el método SpikeProp.

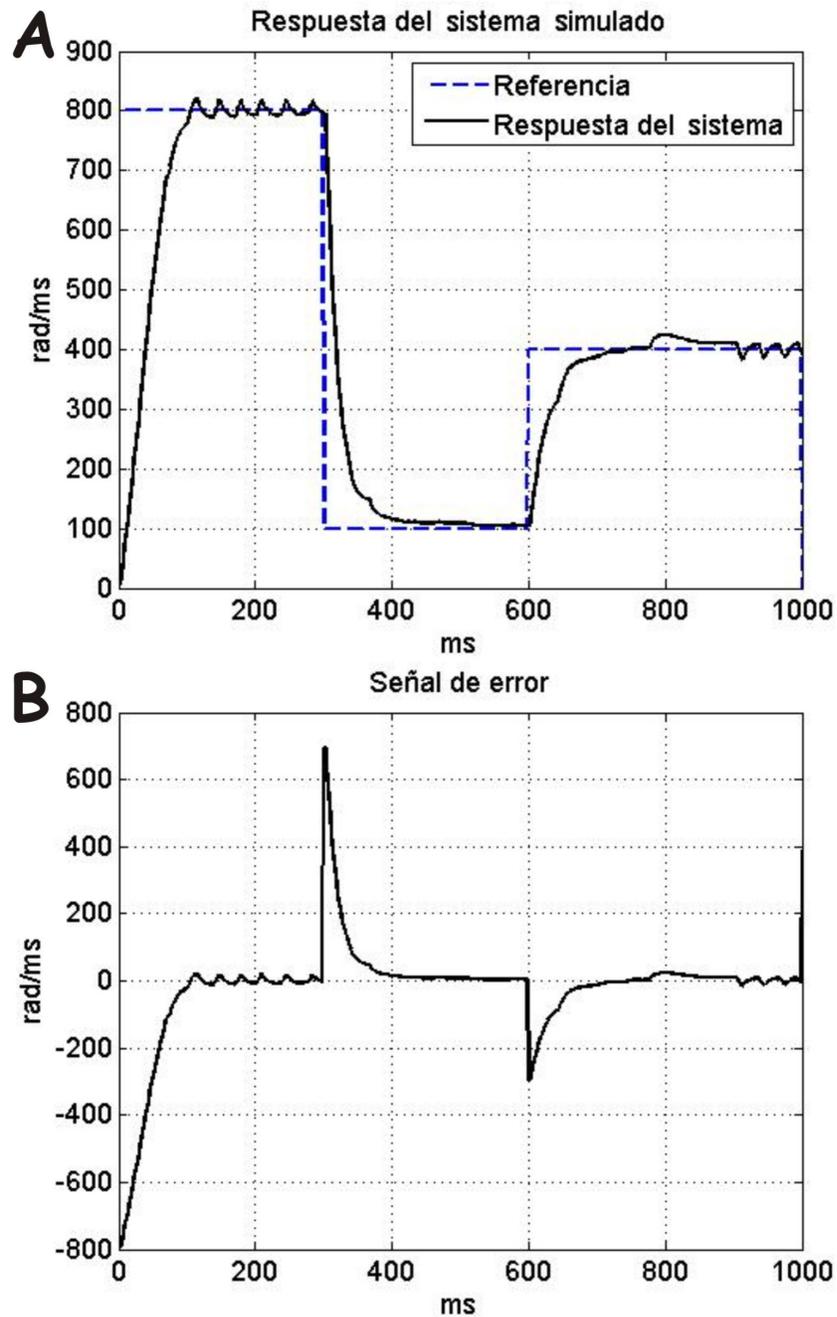


Figura 4.10: En la Figura **A** se describe la respuesta del sistema al control PID-S, la Figura **B** muestra el error de la velocidad del motor con neuronas tipo LIF, entrenadas con el método SpikeProp.

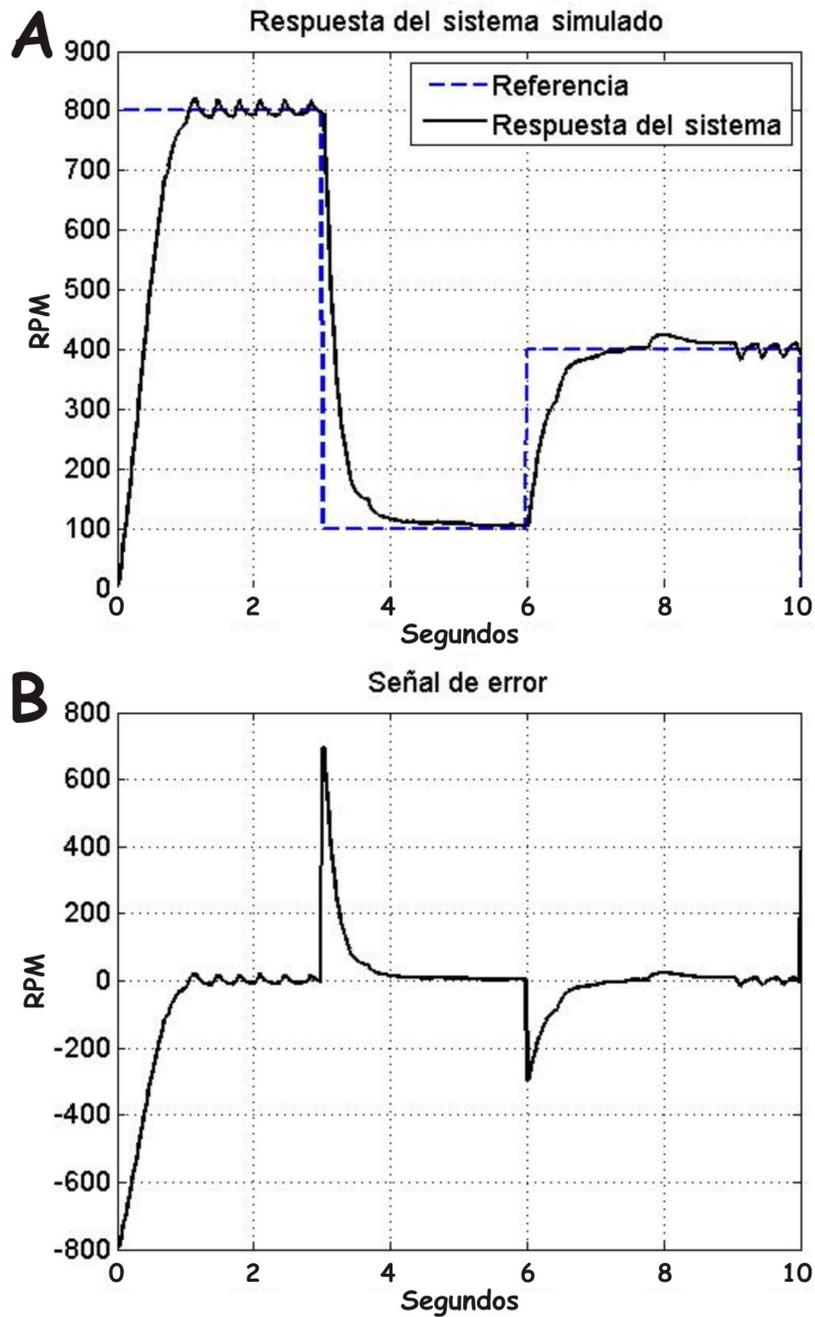


Figura 4.11: En la Figura **A** se describe la respuesta del sistema al control PID-S, la Figura **B** muestra el error de la velocidad del motor con neuronas tipo Izhikevich, entrenadas con el método SpikeProp.

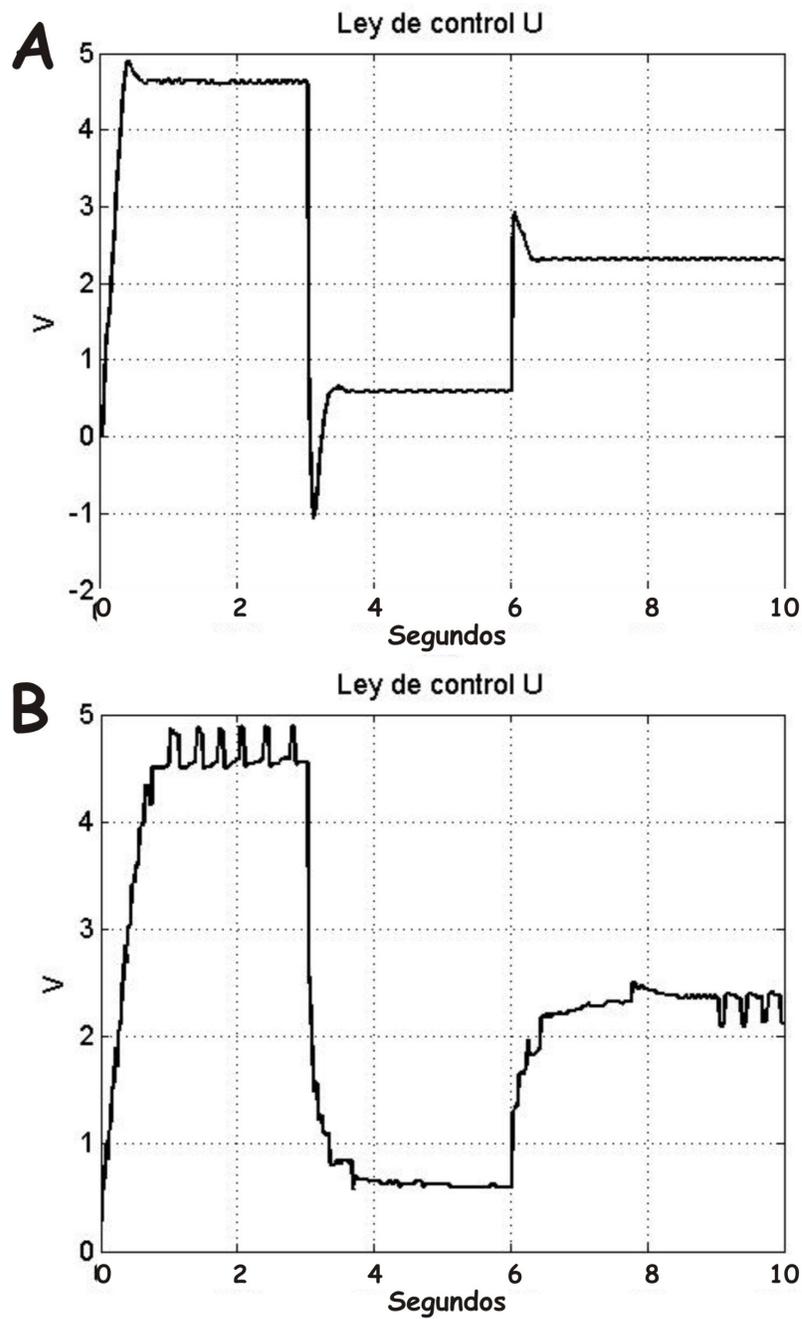


Figura 4.12: En la Figura **A** se describe la ley de control PID-S con neuronas PIF, la Figura **B** muestra la ley de control PID-S con neuronas LIF, entrenadas con el método SpikeProp.

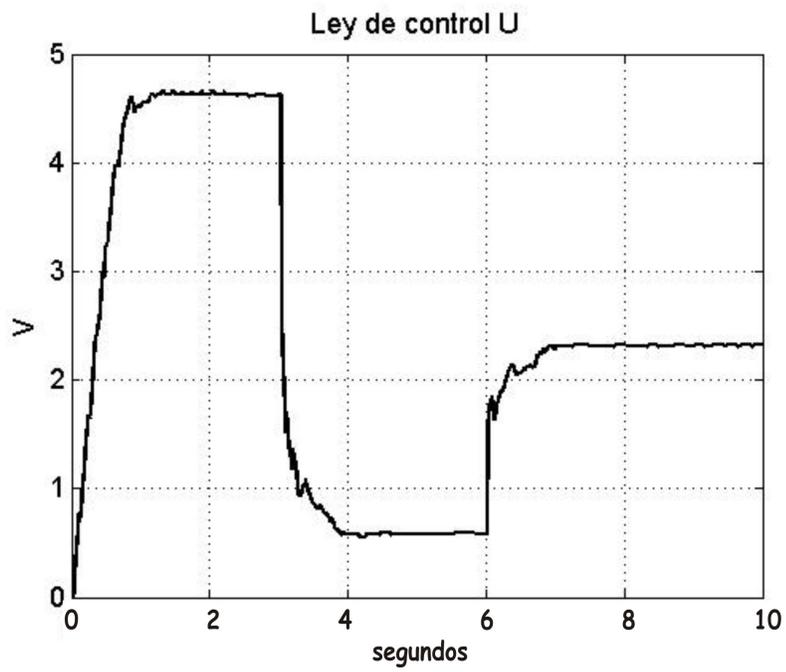


Figura 4.13: En la figura se describe la ley de control PID-S con neuronas Izhikevich, entrenadas con el método SpikeProp.

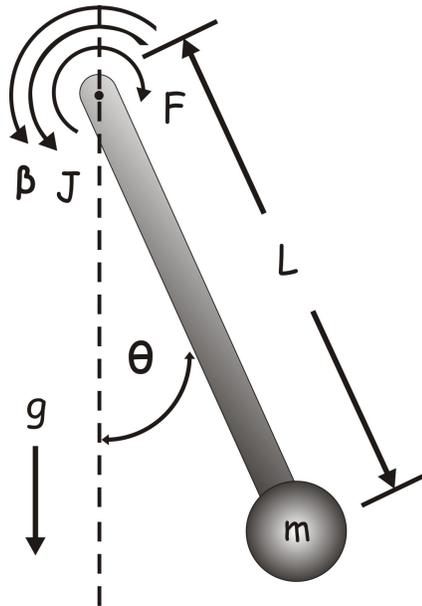


Figura 4.14: Pendulo Invertido

Éste es un sistema no lineal inestable, por lo que la ley de control debe ser no lineal, pero esto no representa ningún problema para las redes neuronales de tercera generación. Como se puede apreciar en la Figura 4.15 para este caso en particular, el péndulo se inicializa a 135° de su punto de equilibrio estable, y se desea que la ley de control lo lleve a 180° , posterior a eso, a los 5 segundos se le aplica una perturbación al sistema, y en ambos casos el péndulo llega a la referencia deseada, cabe mencionar que se trabaja en un zona no lineal inestable, por lo que los controles lineales como el PID clásico no realizan un buen control.

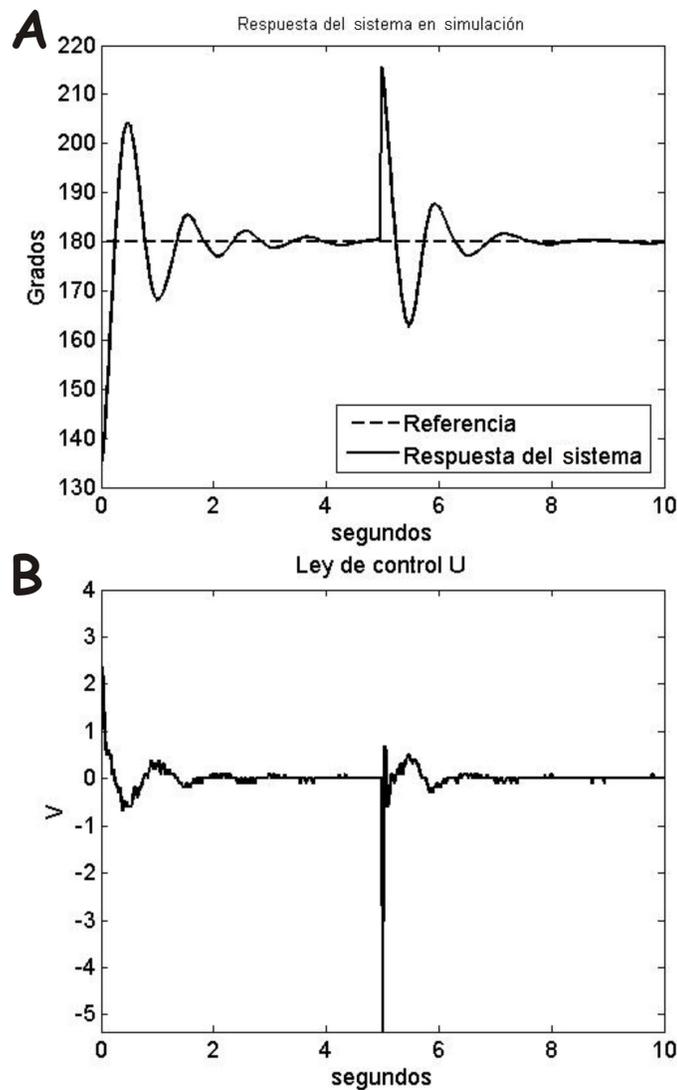


Figura 4.15: En la Figura **A** se muestra la respuesta del péndulo invertido al control PID-S y en la Figura **B** se aprecia la ley de control emitida por la neurona tipo PIF

Para este caso, se implementó el modelo de una neurona tipo LIF, la tasa de aprendizaje se fijó en 0.001, el valor de capacitancia, resistencia, y el valor de reposo y umbral del potencial de membrana para la neurona LIF, fueron fijados en $20\mu F$, 330Ω , $-60mV$ y $-30mV$ respectivamente. En las Figura 4.16 se muestra la grafica del error.

En la Figura ??, se muestra la señal de control del péndulo, emitida por la neurona LIF, como se puede apreciar cuando $t = 5$ segundos se induce una perturbación al sistema y a pesar de eso y de que el sistema es inestable, se consigue que el error llegue a cero tal como se muestra en la Figura 4.16.

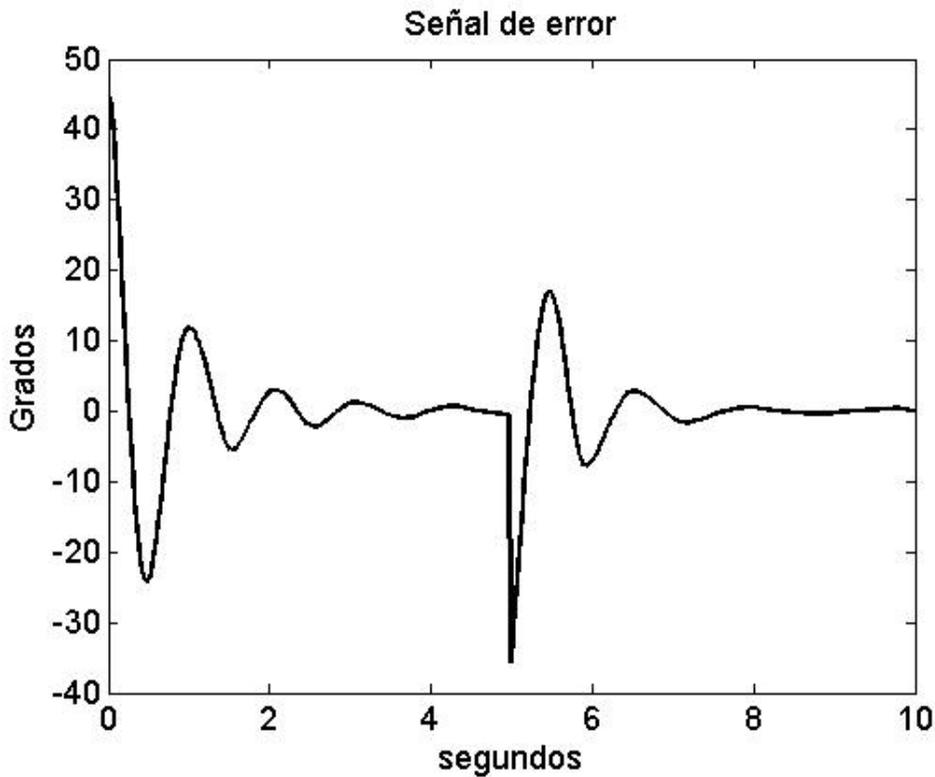


Figura 4.16: Señal de error del control del Pendulo Invertido, neurona PIF

4.4. Resultados de laboratorio

Para comprobar la eficacia del algoritmo basado en redes neuronales artificiales de tercera generación, dentro del área de control se ha implementado el esquema de control PID basado en redes neuronales artificiales de tipo Spike en dos sistemas físicos, un motor de corriente directa y un péndulo invertido.

4.4.1. Sistema no lineal de primer orden

Se implementó el algoritmo propuesto en la Sección 4.2 para el control de un motor de corriente directa, la plataforma experimental que fue utilizada se muestra en la Figura 4.17, la cual consta de un motor de corriente directa, una tarjeta de adquisición de datos de National Instruments (DAQ), una fuente de poder, una etapa de potencia y una computadora labtop marca Sony Vaio.

El periodo de muestreo empleado es igual a un milisegundo y la referencia es una señal que depende del tiempo con las mismas condiciones que en la etapa de simulación, Sección 4.3, ecuación 4.3. Las pruebas fueron realizadas con dos algoritmos de entrenamiento: Relación de Trenes de Pulsos de Entrada-Salida y SpikeProp, con su respectiva combinación de tres modelos neuronales cada uno.

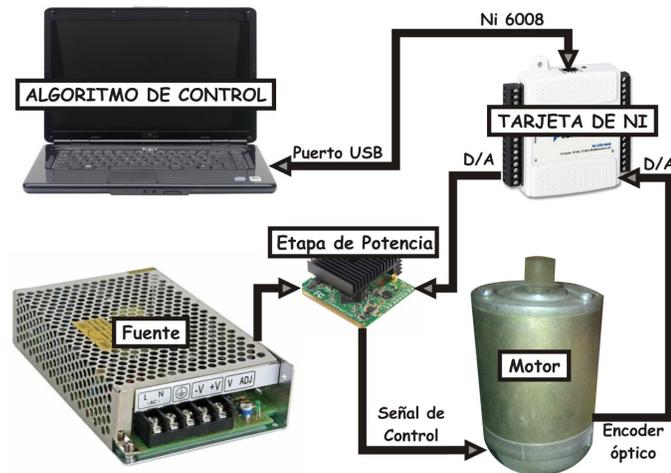


Figura 4.17: Plataforma Experimental.

1. Entrenamiento Relación de Trenes de Pulso Entrada-Salida

En las Figuras 4.18, 4.19 y 4.20 se muestra la respuesta del sistema utilizando el control PID-S bajo el entrenamiento Relación de Trenes de Pulso Entrada-Salida y el modelo PIF, LIF e Izhikevich respectivamente, los valores de entrenamiento de cada trío de neuronas en ambas pruebas son 1, 0.05 y 0.001 para las tres neuronas PIF, 1, 0.05 y 0.001 para las tres neuronas LIF, y 0.01, 0.001 y 0.00001 para las tres neuronas Izhikevich entrenadas con e , $\frac{de}{dt}$ y $\int edt$, respectivamente cada uno de los modelos.

En la Figura 4.21 se muestra la ley de control emitida por dos de los modelos neuronales de tipo umbral y disparo con los mayores niveles de abstracción, y aunque pertenecen al mismo grupo de modelos neuronales el comportamiento es diferente entre sí.

En la Figura 4.22 se muestra la ley de control emitida por el esquema PID-S con neuronas Izhikevich, este tipo de modelo neuronal tiene un gran parecido a las neuronas biológicas reales, estrictamente al de las neuronas de la corteza cerebral de las ratas.

2. Entrenamiento SpikeProp

En las Figuras 4.23, 4.24 y 4.25 se muestra la respuesta del sistema utilizando el control PID-S bajo el entrenamiento SpikeProp y el modelo PIF, LIF e Izhikevich respectivamente, los valores de entrenamiento de cada trío de neuronas en ambas pruebas son -0.2, -0.01 y -0.001 para las tres neuronas PIF, -0.1, -0.01 y -0.001 para las tres neuronas LIF, y -0.05, -0.0008 y -0.00001 para las tres neuronas Izhikevich entrenadas con e , $\frac{de}{dt}$ y $\int edt$ respectivamente cada uno de los modelos.

En la Figura 4.26 se muestra la ley de control emitida por dos de los modelos neuronales de tipo umbral y disparo con los mayores niveles de abstracción, y aunque pertenecen al mismo grupo de modelos neuronales el comportamiento es diferente entre sí.

En la Figura 4.27 se muestra la ley de control emitida por el esquema PID-S con neuronas Izhikevich, este tipo de modelo neuronal tiene un gran parecido a las neuronas biológicas reales, estrictamente al de las neuronas de la corteza cerebral de las ratas.

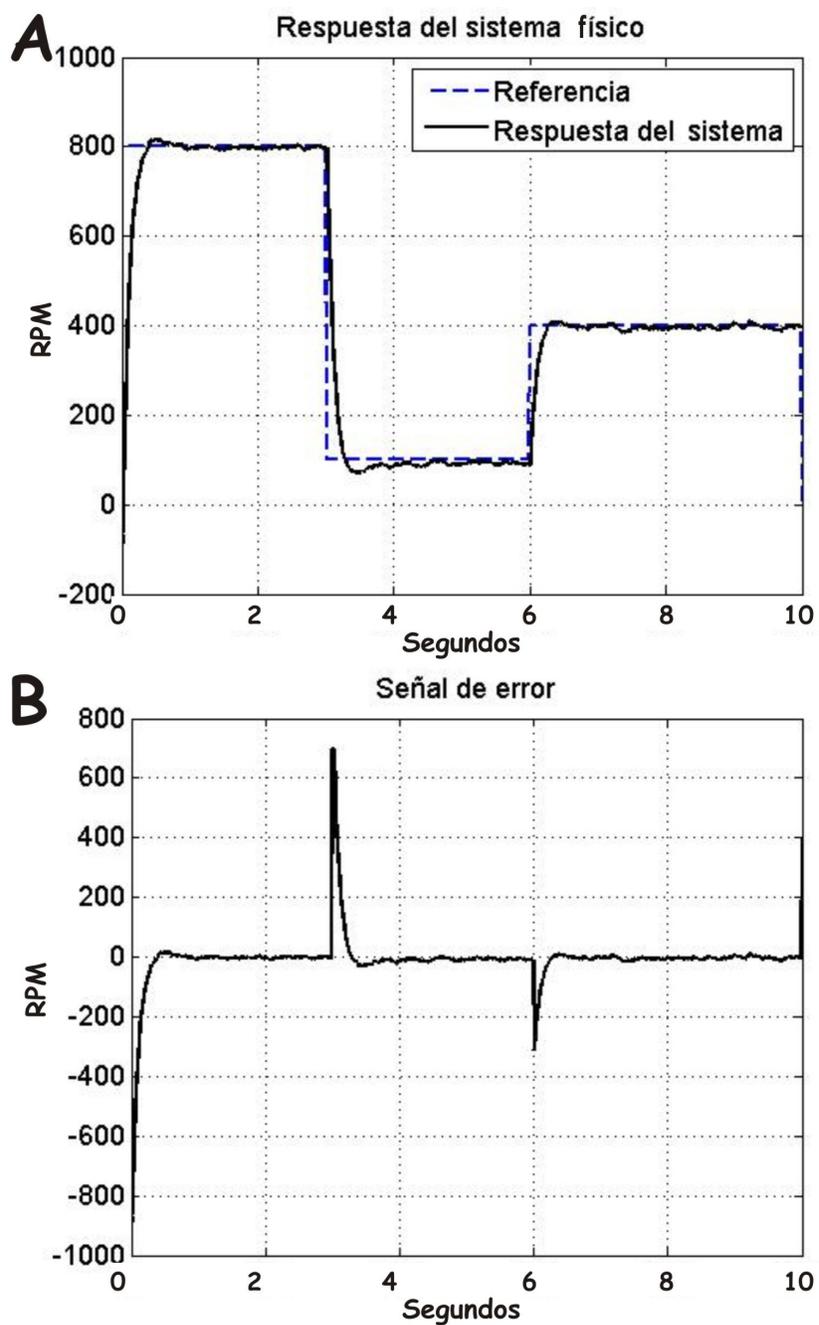


Figura 4.18: En la Figura **A** se describe la respuesta del sistema al control PID-S, la Figura **B** muestra el error de la velocidad del motor con neuronas tipo PIF, entrenadas con el método Regla Modificada de Widrow-Hoff.

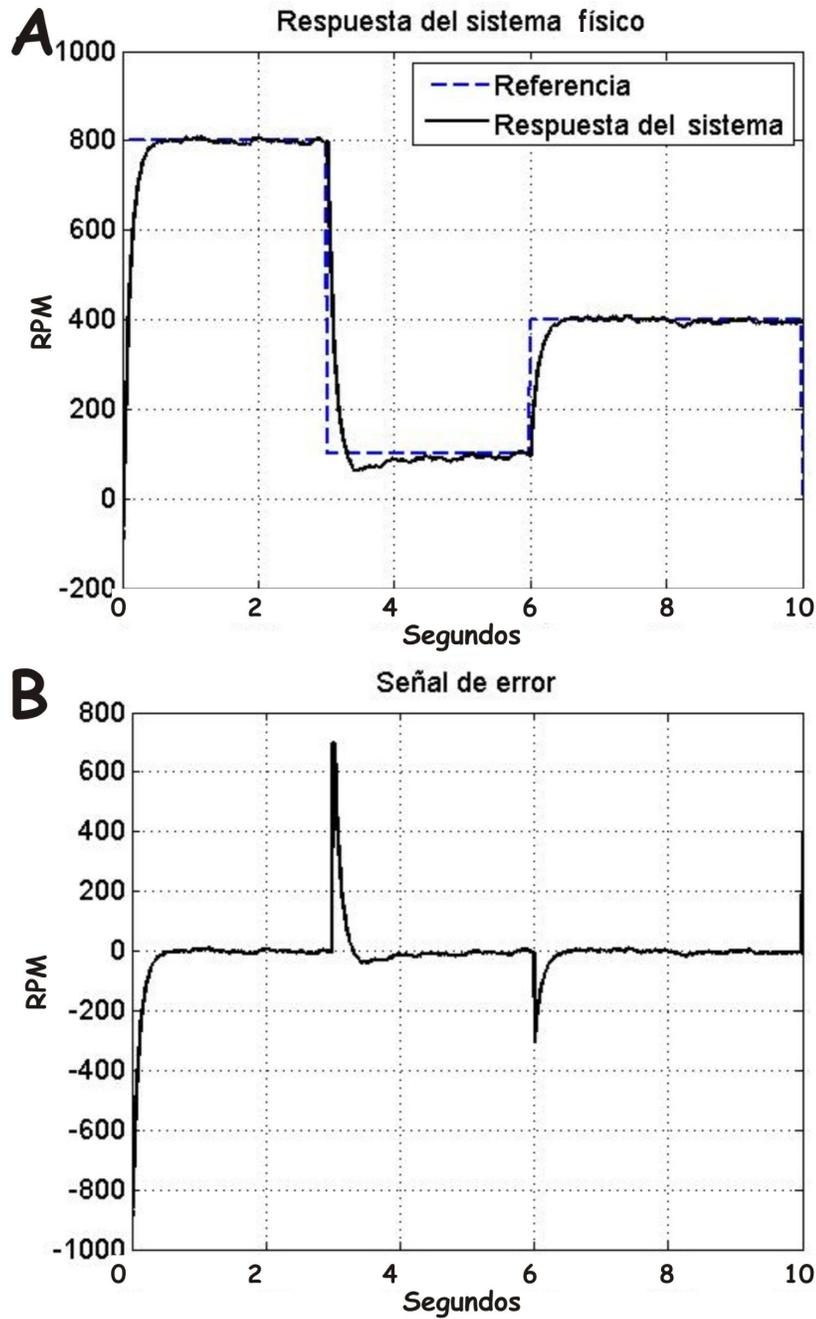


Figura 4.19: En la Figura **A** se describe la respuesta del sistema al control PID-S, la Figura **B** muestra el error de la velocidad del motor con neuronas tipo LIF, entrenadas con el método Regla Modificada de Widrow-Hoff.

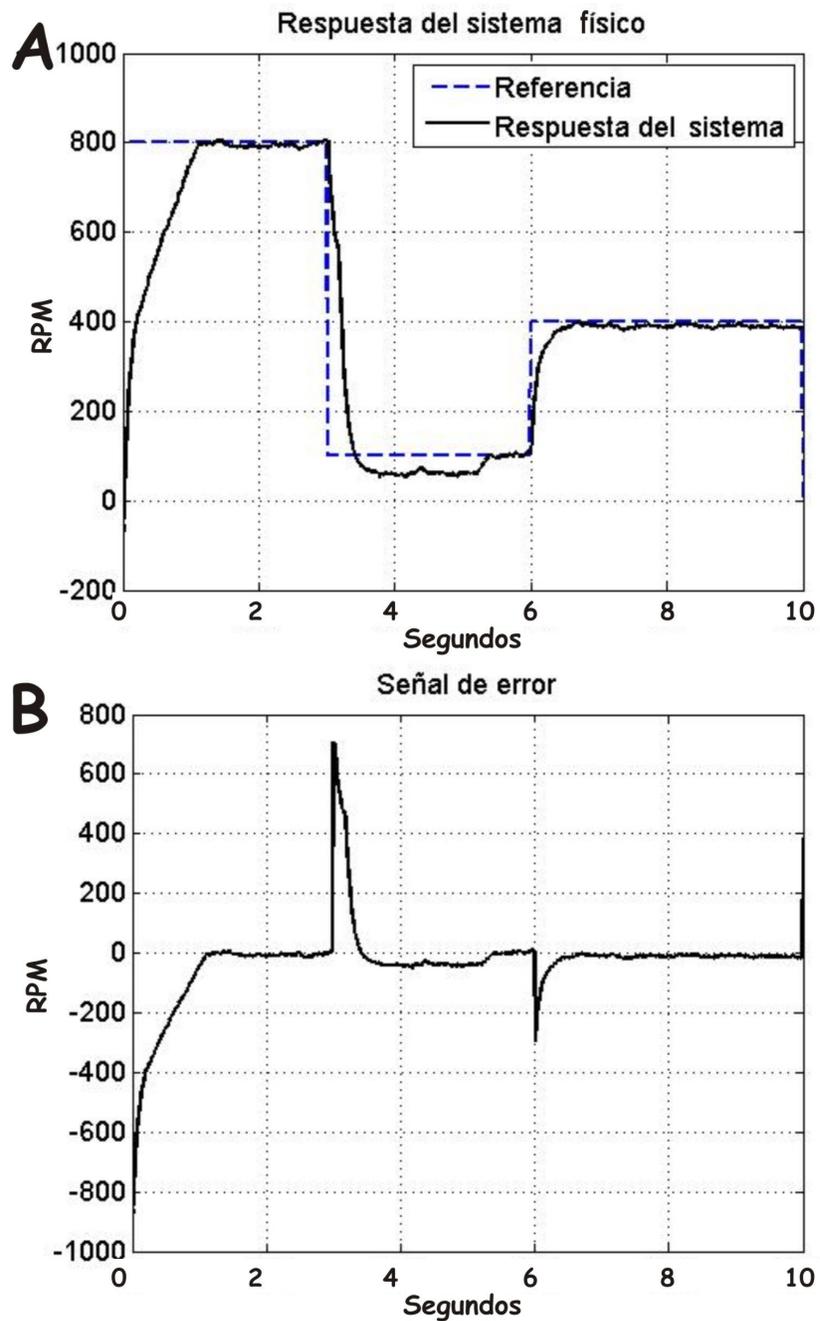


Figura 4.20: En la Figura **A** se describe la respuesta del sistema al control PID-S, la Figura **B** muestra el error de la velocidad del motor con neuronas tipo Izhikevich, entrenadas con el método Regla Modificada de Widrow-Hoff.

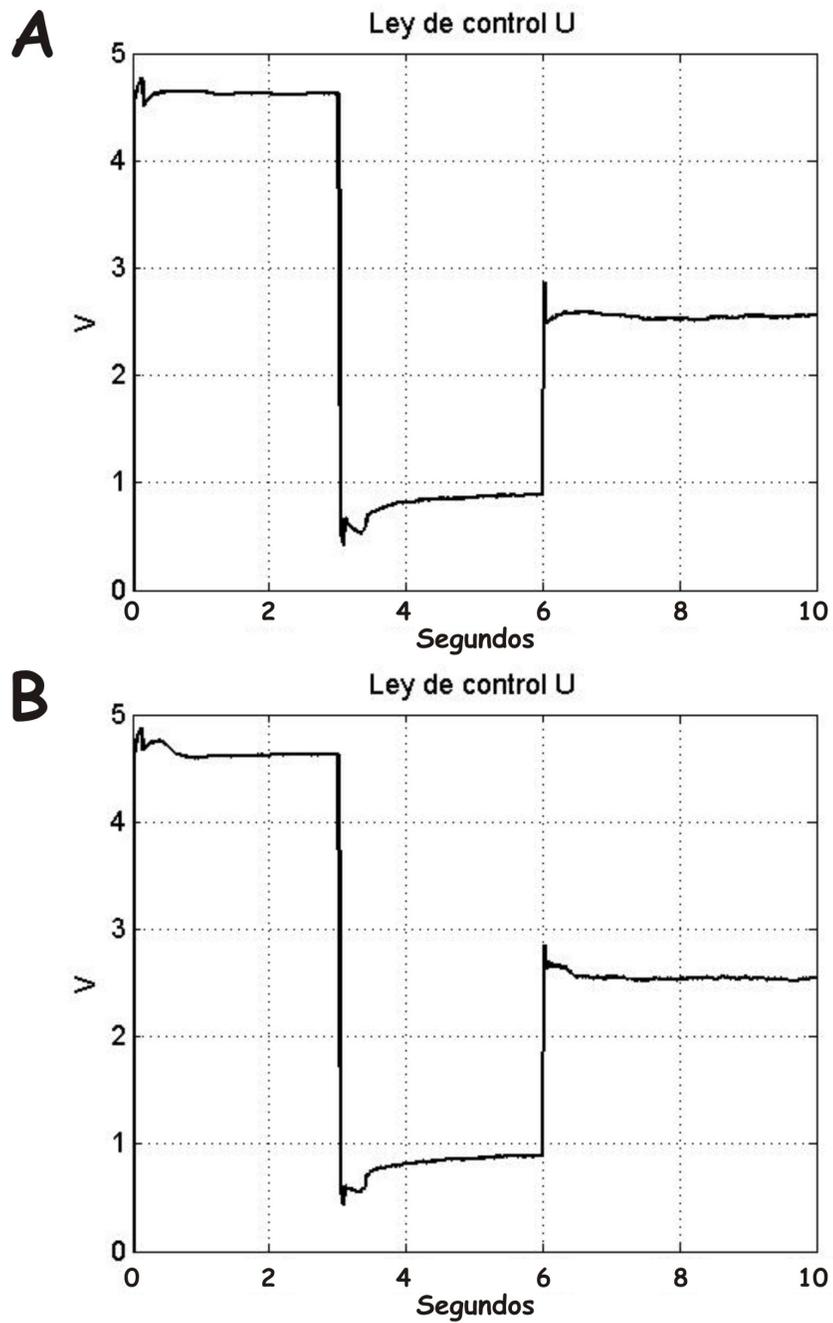


Figura 4.21: En la Figura **A** se describe la ley de control PID-S con neuronas PIF, la Figura **B** muestra la ley de control PID-S con neuronas LIF, entrenadas con el método Regla Modificada de Widrow-Hoff.

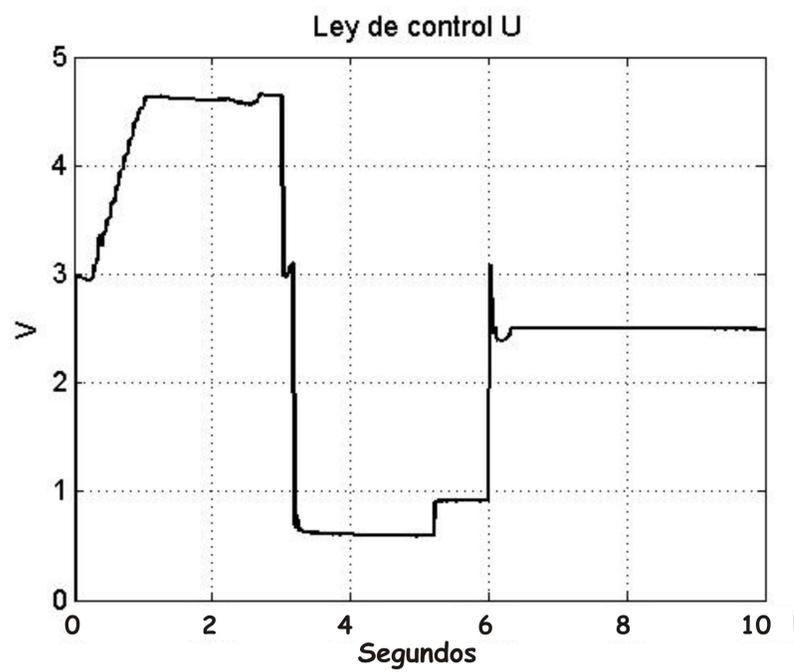


Figura 4.22: En la figura se describe la ley de control PID-S con neuronas Izhikevich, entrenadas con el método Regla Modificada de Widrow-Hoff.

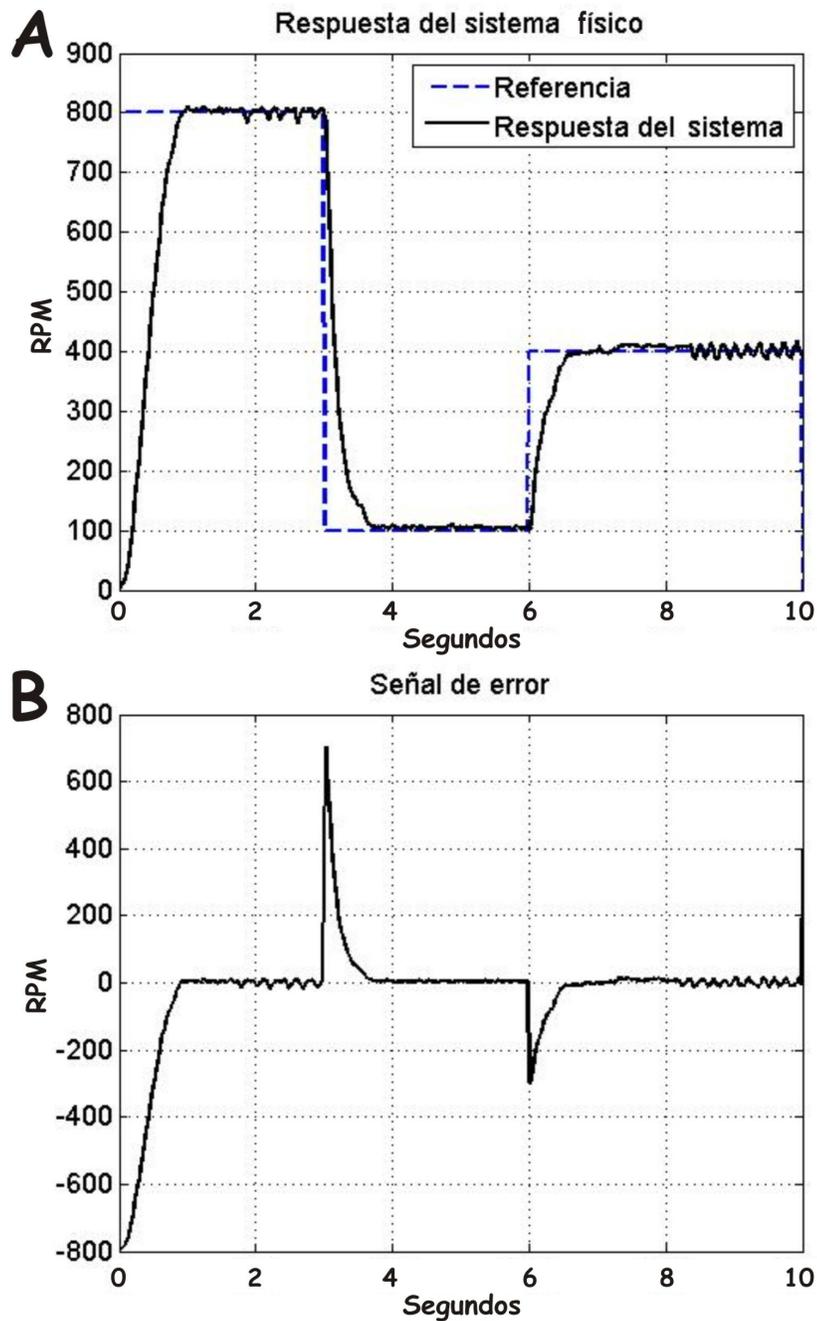


Figura 4.23: En la Figura **A** se describe la respuesta del sistema al control PID-S, la Figura **B** muestra el error de la velocidad del motor con neuronas tipo PIF, entrenadas con el método SpikeProp.

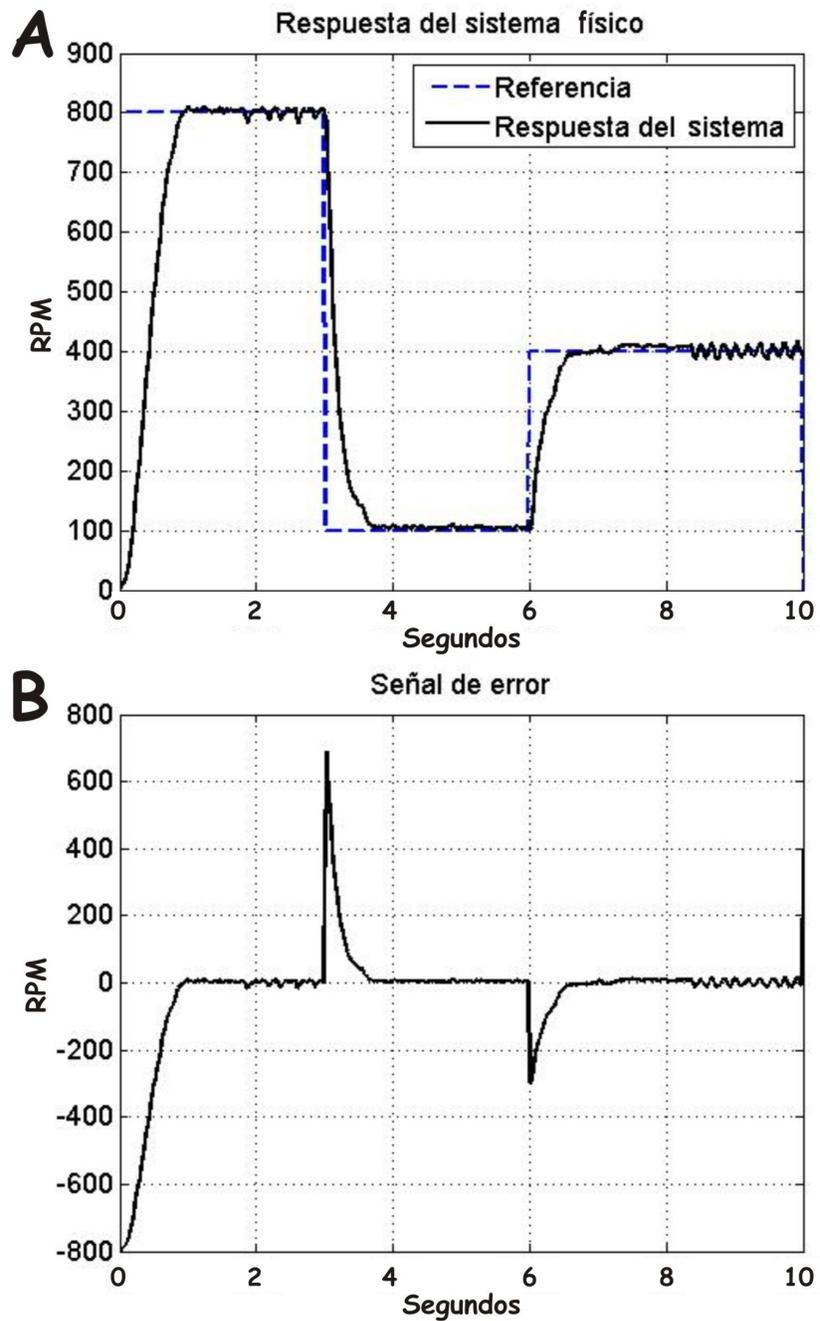


Figura 4.24: En la Figura **A** se describe la respuesta del sistema al control PID-S, la Figura **B** muestra el error de la velocidad del motor con neuronas tipo LIF, entrenadas con el método SpikeProp.

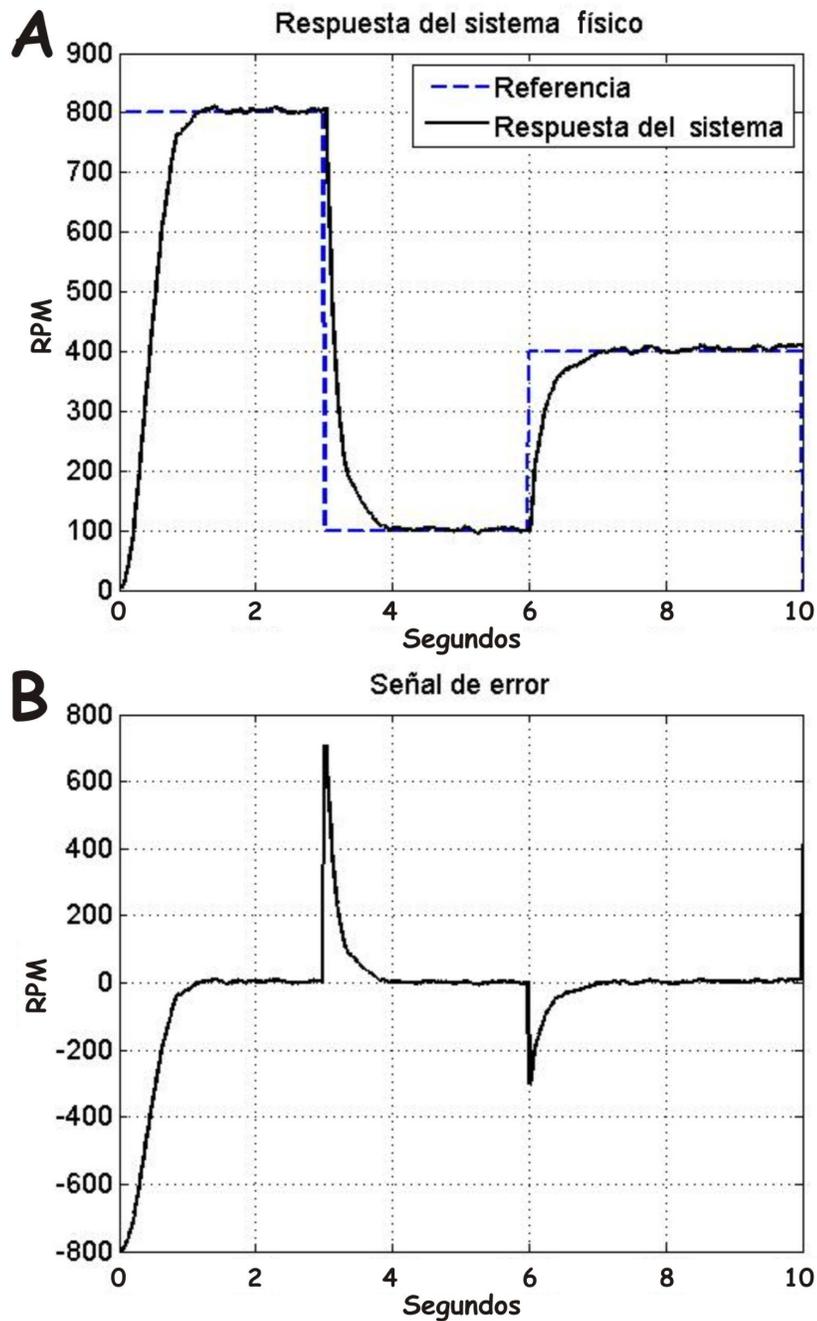


Figura 4.25: En la Figura **A** se describe la respuesta del sistema al control PID-S, la Figura **B** muestra el error de la velocidad del motor con neuronas tipo Izhikevich, entrenadas con el método SpikeProp.

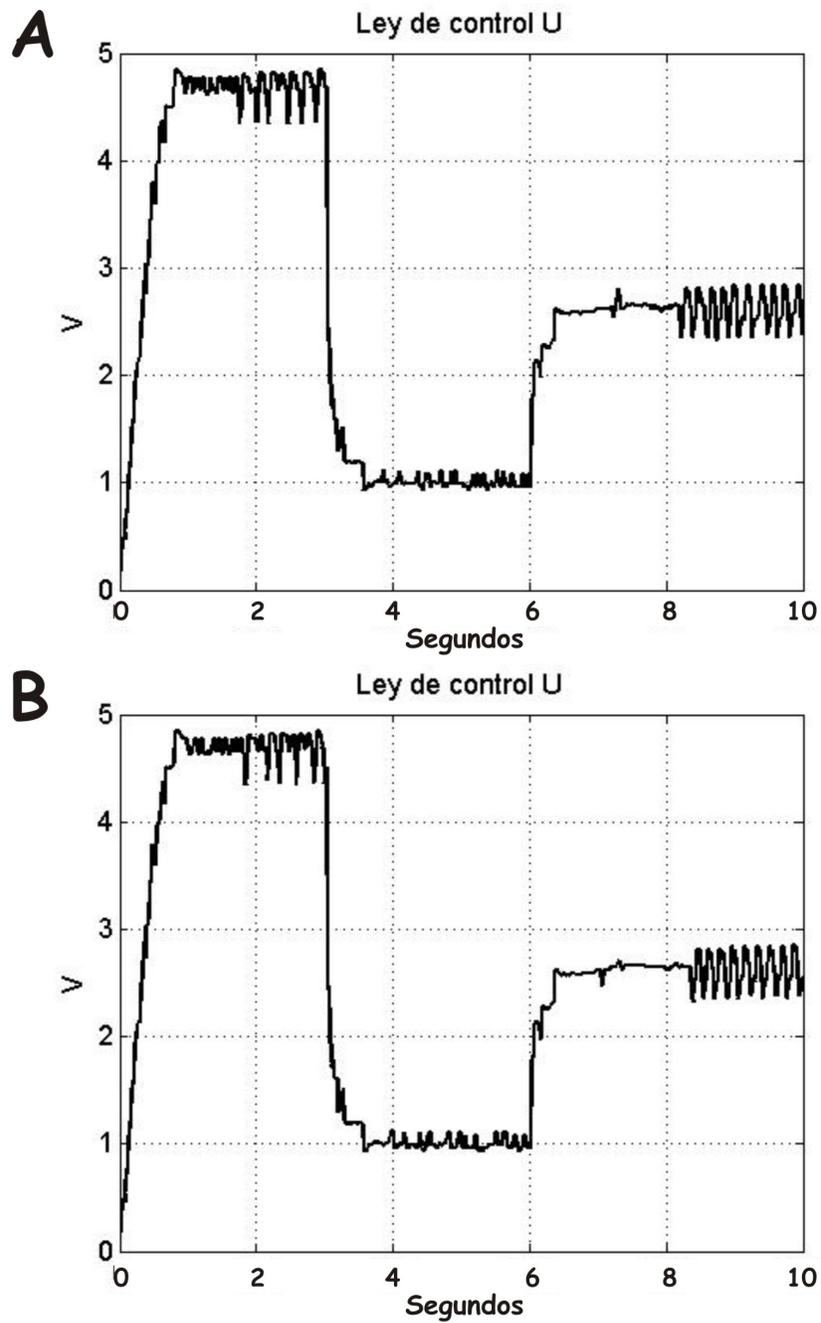


Figura 4.26: En la Figura **A** se describe la ley de control PID-S con neuronas PIF, la Figura **B** muestra la ley de control PID-S con neuronas LIF, entrenadas con el método SpikeProp.

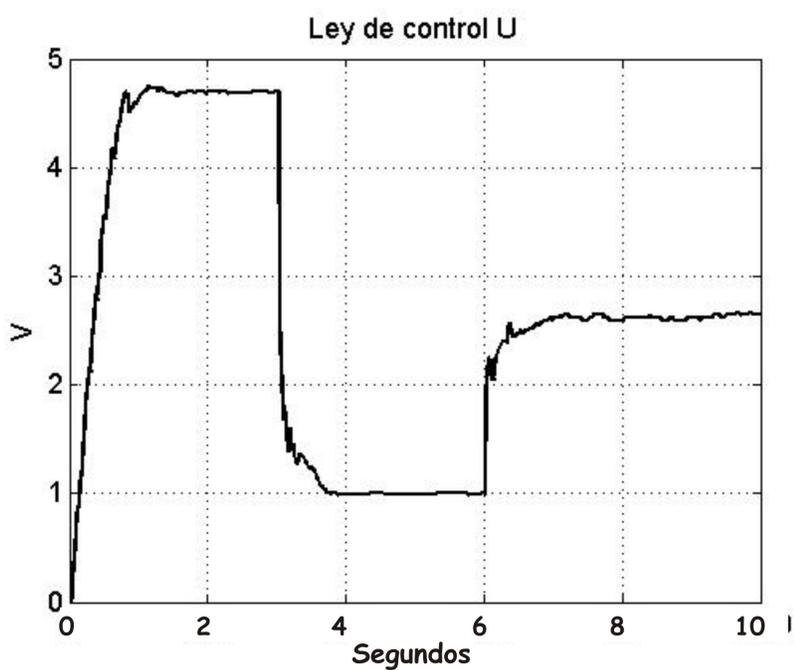


Figura 4.27: En la figura se describe la ley de control PID-S con neuronas Izhikevich, entrenadas con el método SpikeProp.

4.4.2. Sistema no lineal de segundo orden

Al igual que en la sección 4.3, en esta sección se implementa el esquema de control PID basado en redes neuronales artificiales de tipo Spike, para controlar un péndulo invertido.

En la Figura 4.28 se muestra la plataforma experimental, que consta de un péndulo invertido acoplado a un motor de corriente directa, una tarjeta de adquisición de datos de National Instruments, una fuente de poder y una computadora laptop marca Vaio Sony modelo SVF142C29U. El periodo de muestreo empleado en el control es de un milisegundo (ms). Todos los valores de entrenamiento son dados de manera arbitraria sin considerar la dinámica de la planta a controlar.

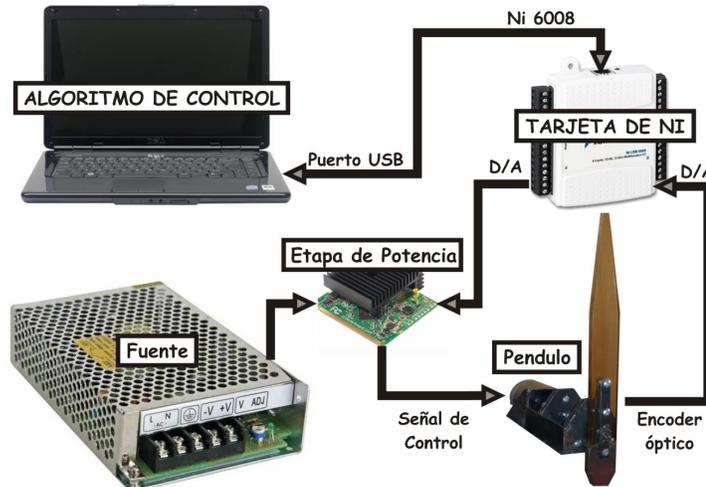


Figura 4.28: Plataforma experimental Péndulo invertido

Para el control del péndulo invertido se considero un tasa de aprendizaje igual a 0.0001 y todos los pesos presinápticos fueron inicializados al azar en un intervalo $(0, 1]$, el valor de reposo y umbral del potencial de membrana para la neurona es de $-60mV$ y $-30mV$ respectivamente, los parámetros de la neurona tipo LIF de resistencia y capacitancia se consideraron de 330Ω y $20\mu F$ respectivamente.

El péndulo se inicializa a 135° de su punto de equilibrio estable, y se desea que la ley de control lo lleve a 180° , posterior a eso, a los 5 segundos se le aplica una perturbación al sistema, y en ambos casos el péndulo llega a la referencia deseada, cabe mencionar que se trabaja en un zona no lineal inestable, por lo que los controles lineales como el PID clásico no realizan un buen control.

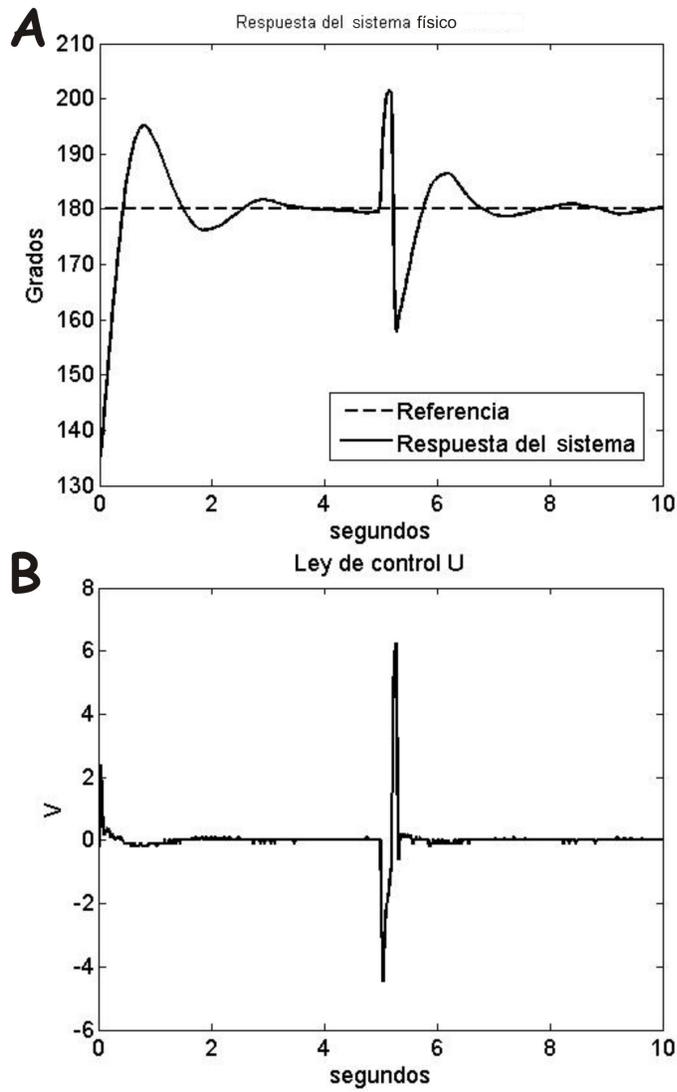


Figura 4.29: En la Figura **A** se muestra la respuesta del péndulo invertido al control PID-S y en la Figura **B** se aprecia la ley de control emitida por la neurona tipo PIF

En la Figura 4.30, se muestra la señal de error del péndulo, como se puede apreciar cuando $t = 5$ segundos se induce una perturbación al sistema y a pesar de eso y de que el sistema es inestable, se consigue que el error llegue a cero.

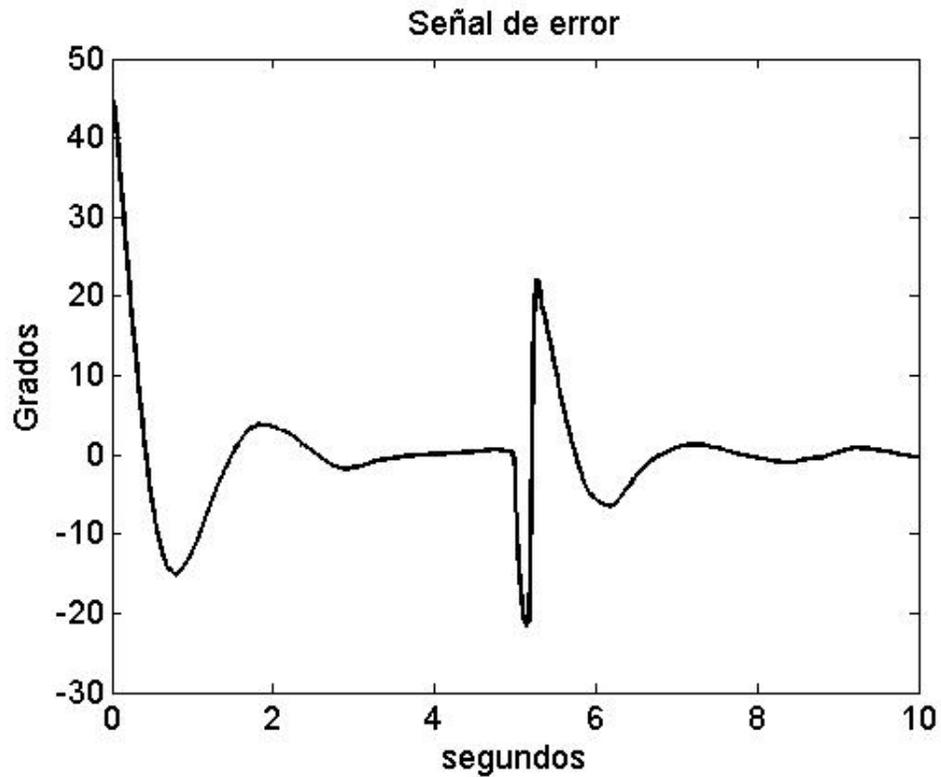


Figura 4.30: Señal de error del control del Pendulo Invertido, neurona PIF

4.5. Conclusiones

- El área de las redes neuronales de tercera generación es nueva en aplicaciones de control automático, su característica principal es que no requieren del modelo matemático del sistema.
- El control PID-S propuesto, representa una alternativa al PID clásico, pues su desempeño es muy similar, con la ventaja de que en el primero no es necesario calcular las ganancias K_p , K_i , y K_d .
- Los resultados en simulación numérica y en tiempo real son aceptables, con lo que se concluye que el objetivo de la tesis se cumplió satisfactoriamente.

4.6. Trabajos a futuro

Algunos de los trabajos futuros que se desprenden de esta tesis son:

- La implementación de las redes neuronales de tercera generación en un tipo de sistemas más complejos por ejemplo en sistemas subactuados, modificando el PID-S y/o diseñando una nueva arquitectura.
- Utilizar las redes neuronales en el área de sistemas con retardos, pues la tasa de aprendizaje va directamente relacionada con el comportamiento del modelo neuronal, retardando su dinámica se podría compensar los retardos intrínsecos de la planta.
- Entrenar a las redes neuronales de tercera generación con un índice de desempeño en lugar del error, para optimizar el control.
- Una de las mayores propiedades de las redes neuronales de tercera generación no es su plasticidad y la capacidad que ésta conlleva para el aprendizaje, sino que tienen la capacidad de poseer memoria, esto ofrece una gran gama de posibilidades, como el utilizarlas para recordar errores y predecirlos cuando las condiciones sean similares, ofreciendo una señal de control predictivo.

Glosario

Actuador. Dispositivo capaz de convertir energía eléctrica, hidráulica u otro tipo de energía en movimiento [42].

Algoritmo. Conjunto ordenado y finito de operaciones que permite hallar la solución de un problema [23].

Aprendizaje no supervisado. También llamado aprendizaje auto-supervisado, no se requiere influencia externa para ajustar los pesos de las conexiones entre sus neuronas. No se recibe ninguna información por parte del entorno que indique si la salida generada en respuesta a una determinada entrada es correcta o no [62].

Aprendizaje supervisado. Tipo de aprendizaje basado en una respuesta preprogramada para una entrada particular. Es decir, cada par de entrenamiento está compuesto por los valores de entrada a la red y los valores de salida deseados para tales entradas [62].

Aprendizaje. Proceso por el cual los parámetros libres de una red neuronal son adaptados a través de estimulaciones generadas por el entorno en el cual la red neuronal esta empotrada [62].

Arquitectura. En redes neuronales artificiales consiste en la organización y disposición de las neuronas formando capas y la forma en que se conectan entre sí. En este sentido, los parámetros fundamentales de la red que constituyen su arquitectura son: el número de capas, el número de neuronas por capa, el grado de conectividad y el tipo de conexiones entre neuronas [62].

Automatización. Ciencia que trata de sustituir en un proceso el operador humano por un determinado dispositivo, generalmente electromecánico [42].

Codificador (Encoder). Dispositivo de retroalimentación que convierte un movimiento mecánico en señales eléctricas, las cuales indican la posición de un actuador [46].

Control adaptable. Esquema de control en el cual los parámetros del controlador se ajustan ante cambios de la planta, de manera que el comportamiento en lazo cerrado conserva las características deseadas de diseño. El control adaptable es inherentemente no lineal. Se aplica generalmente aunque no exclusivamente a sistemas con parámetros que varían lentamente en el tiempo [10].

Control. Proceso mediante el cual un sistema es llevado a parámetros preestablecidos [42].

Energía. Capacidad de realizar un trabajo [10].

Entrenamiento. Se refiere al proceso consiente y planeado de transferir conocimiento, habilidades y capacidades [62].

Época. Presentación completa del conjunto de entrenamiento a la red neuronal [23].

Espacio de estado. Espacio de n dimensiones, donde cualquier estado de un sistema dado puede ser representado como un punto [42].

Estabilidad. Condición en la cual las variables críticas de un sistema dinámico se mantienen invariables o permanecen dentro de unos límites determinados [56].

Estado. Conjunto mínimo de variables de un sistema dado (variables de estado), tal que, conociendo su valor en un instante dado, permiten conocer la respuesta del sistema ante cualquier señal de entrada o perturbación [42].

Función de activación. Sirve para limitar la amplitud de la neurona de salida. La función de activación limita el rango de amplitud permisible de la señal de salida a algún valor finito [23].

Función suave. La función f se dice que es suave si sus derivadas parciales de cualquier orden existen y son continuas.

Grados de libertad. Número de coordenadas que describen de manera única la posición de todos los eslabones de un sistema [10].

Hardware. Se dice de cualquier componente físico relacionado con cierta tecnología [10].

Identificación. Conjunto de métodos para la obtención de modelos matemáticos a partir de datos experimentales de las entradas y salidas de un sistema. Evitándose la obtención de un modelo a partir de complejas leyes [10].

- Manipulador.** Mecanismo formado generalmente por elementos en serie, articulados entre sí, destinado al agarre y desplazamiento de objetos. Es multifuncional y puede ser gobernado directamente por un operador humano o mediante dispositivo lógico [42].
- Modelo matemático.** Es la representación por medio de ecuaciones de la dinámica de un sistema. Es el tipo de modelo más importante para la ciencia y la tecnología [42].
- Neurona** Célula nerviosa, elemento fundamental de la arquitectura nerviosa. Es una unidad de procesamiento de información que es fundamental para la operación de una red neuronal [62].
- Perceptrón.** Dicese de una arquitectura de red neuronal estática multicapa, cuyos nodos ocultos tienen una función de activación suave (sigmoideal), mientras que los nodos de salida poseen una función de activación lineal [23].
- Perturbación.** Variable no deseada aplicada a un sistema y la cual tiende a afectar adversamente el valor de una variable controlada. Si la perturbación se genera dentro del sistema se denomina interna, en tanto que una perturbación externa se produce fuera del sistema y es una entrada [56].
- Punto de equilibrio.** El estado \mathbf{x}^* se dice que es un punto de equilibrio del sistema $\dot{\mathbf{x}} = f(\mathbf{x})$ si $\mathbf{x}(t)$ es igual a \mathbf{x}^* , para todo $t \geq t_0$ [56].
- Redes neuronales artificiales.** Procesador distribuido masivamente paralelo constituido de unidades de procesamiento simples, el cual tiene una tendencia natural a almacenar conocimiento experimental y hacerlo viable para su posterior uso [62].
- Redes neuronales dinámicas.** Redes neuronales artificiales que poseen dispositivos de memoria en sus conexiones internas, sus salidas son funciones de entradas presentes y pasadas [23].
- Redes neuronales estáticas.** Redes neuronales artificiales cuyas ecuaciones no lineales que las caracterizan no tienen memoria, es decir sus salidas son tan solo funciones de las entradas actuales [23].
- Retroalimentación.** En el proceso de control consiste en la reinyección de la salida (o salidas) a la entrada del sistema, con el fin de mantener cierto desempeño [42].
- Retropropagación.** Algoritmo de entrenamiento basado en el método del descenso del gradiente [62].

Robot. Dispositivo generalmente mecánico, que desempeña tareas automáticamente, ya sea de acuerdo a supervisión humana directa, a través de un programa predefinido o siguiendo un conjunto de reglas generales. Generalmente estas tareas reemplazan, asemejan o extienden el trabajo humano, como ensamble en manufactura, manipulación de objetos pesados o peligrosos, trabajo en el espacio, etc. [10].

Ruido. Típicamente es una perturbación aleatoria. Señal indeseada presente en un sistema físico [46].

Sensor. Dispositivo que convierte un parámetro físico (como temperatura, presión, flujo, velocidad, posición) en una señal eléctrica. En algunos casos se le considera un sinónimo de transductor, pero un verdadero sensor contiene un sistema de acondicionamiento de la señal, de manera que es mucho más sencillo realizar una medición [46].

Sinapsis. Son uniones especializadas mediante las cuales las células del sistema nervioso envían señales de unas a otras y a células no neuronales como las musculares o glandulares. Son la estructura elemental y unidad funcional que regulan la interacción entre neuronas [62].

Sistema dinámico. Sistema cuya respuesta depende de entradas presentes y pasadas (e incluso futuras), dicho sistema puede ser representado mediante ecuaciones diferenciales dinámicas [46].

Sistema en tiempo real. Un sistema de tiempo real STR es un sistema informático en el que es significativo el tiempo en el que se producen las acciones. Las acciones deben realizarse dentro de un intervalo de tiempo determinado [46].

Sistema estático. Sistema cuya respuesta depende únicamente de las entradas presentes [46].

Sistema lineal. Se dice que un sistema es lineal si cumple con los principios de homogeneidad y superposición [56].

Sistema no lineal. Se dice que un sistema es lineal si no cumple con los principios de homogeneidad o superposición [56].

Sistema subactuado. Sistema con menos actuadores que grados de libertad [10].

Software. Se dice de todos los componentes intangibles de una computadora, es decir, al conjunto de programas y procedimientos necesarios para hacer posible la realización de una tarea específica [10].

Umbral. Tiene el efecto de incrementar o decrementar la entrada de la red de la función de activación, dependiendo de si es positivo o negativo, respectivamente [62].

Variables de estado. Conjunto más pequeño de variables que determinan el estado de un sistema dinámico [42].

Vector de estado. Si se necesitan n variables para describir el estado de un sistema dado, entonces estas n variables se pueden considerar como los componentes de un vector \mathbf{x} [42].

Bibliografía

- [1] R. Van Rullen A. Delorme, J. Gautrais and S. Thorpe. *SpikeNET: A simulator for modeling large networks of integrate and fire neurons*. Neurocomputing, vol. 26-27, pp. 989-996, 1999.
- [2] Pedro Albertos and Antonio Sala. *Fuzzy logic controllers advantages and drawbacks*. VIII Congreso Latinoamericano de Control Automático, pp. 833-844, Viñas del Mar, 1998.
- [3] Satoshi Matsuda Ammar Mohemmed, Stefan Schliebs and Nikola Kasabov. *Method for Training a Spiking Neuron to Associate Input-Output Spike Trains*. EANN/AIAI 2011, Part I, IFIP AICT 363, pp. 219 - 228, Springer Berlin Heidelberg, 2011.
- [4] Stefan Schliebs Kshitij Dhoble Ammar Mohemmed, Satoshi Matsuda and Nikola Kasabov. *Optimization of Spiking Neural Networks with Dynamic Synapses for Spike Sequence Generation using PSO*. Proceedings of International Joint Conference on Neural Networks, pp. 2969-2974, San Jose, CA, 2011.
- [5] Aurora Plata Anton Labraña and Celia Peña. *Álgebra Lineal. Resolución de sistemas lineales*. Madrid, Síntesis, 1995.
- [6] O. Lezoray B. Meftah and A. Benyettou. *Segmentation and edge detection based on spiking neural network model*. Neural Process Letters, vol. 32, pp. 131-146, 2010.
- [7] S. M. Bohte. *The evidence for neural information processing with precise spike-times: A survey*. Natural Computing, vol. 3, pp. 195-206, 2004.
- [8] O. Booij and H. Tat Nguyen. *A gradient descent rule for spiking neurons emitting multiple spikes*. Information Processing Letters, vol. 95, pp. 552-558, 2005.
- [9] Robert L. Boylestad and Louis Nashelsky. *Electrónica teoría de circuitos*. Cuarta Edición, México, 1989.

-
- [10] Wilson C. and Sadler P. *Kinematics and dynamics of machinery*. McGraw-Hill Interamericana de España S.A., 1997.
- [11] P. Mitra C. Johnson, G. K. Venayagamoorthy. *Comparison of a spiking neural network and an MLP for robust identification of generator dynamics in a multimachine power system*. Neural Networks, vol. 22, pp. 833-841, 2009.
- [12] P. Esquivel Carlos E. Castaneda. *Direct Current Motor Control based on High Order Neural Networks using Stochastic Estimation*. Neural Networks (IJCNN), The 2010 International Joint Conference, pp. 1-4, Barcelona, 2010.
- [13] Chi-Tsong Chen. *Linear System Theory and Design*. Tercera edición, New York, Oxford., 1999.
- [14] Sin-Chung Chi-Chung Cheung and Andrew K. Lui. *Improving the Quickprop algorithm*. World Congress on Computational Intelligence, Australia, 2012.
- [15] R. Chibani and M. Chtourou. *IMC based automatic tuning method of PID controllers using neural networks*. 6th International Multi-Conference on Systems, pp. 1-5, Djerba, Tunisia, 2009.
- [16] Y. Dan and M. Poo. *Spike Timing-Dependent Plasticity of Neural Circuits*. Neuron, vol. 44, pp. 23-30, Berkeley, California, 2004.
- [17] Gérard Dreyfus. *Neural Networks. Methodology and applications*. Segunda edición, Springer-Verlag Berlin Heidelberg, 2005.
- [18] J. H. Schwartz E. R. Kandel and T. M. Jessell. *Principles of Neural Science*. Cuarta Edición, New York: McGraw-Hill, Health Professions Division, 2000.
- [19] J. J. Eggermont. *Is There a Neural Code?* Neuroscience and Biobehavioral Reviews, vol. 22, pp. 355-370, 1998.
- [20] Brian W. Evans. *Arduino Notebook A Beginners Reference Written and compiled*. San Francisco, California, 2007.
- [21] G. Z. Qi F. Amini, H. M. Chen and J. C. S. Yang. *Generalized Neural Network Based Model for Structural Dynamic Identification, pp. 138-142*. Analytical and Experimental Studies, Intelligent Information Systems, Grand Bahama Island, 1997.
- [22] M.A. Anaeib F. Shahrakia and A.R. Arjomandzadeha. *Adaptive System Control with PID Neural Networks*. University of Sistan and Baluchestan, Zahedan, Iran., 2003.

-
- [23] L. Faussett. *Fundamentals of Neural Networks*. Prentice Hall, Englewood Cliffs, New Jersey, USA, 1994.
- [24] A. Gupta and L. N. Long. *Hebbian learning with winner take all for spiking neural networks*. Neural Networks, 2009. IJCNN 2009. International Joint Conference, Atlanta, GA, 2009.
- [25] M. Colley V. Callaghan H. Hagraas, A. Pounds-Cornish and G. *Evolving Spiking Neural Network Controllers for Autonomous Robots*. Robotics and Automation. Proceedings. ICRA. IEEE International Conference, New Orleans, vol. 5, pp.4620-4626, 2004.
- [26] W. Lei N. Hang H. Yuehua, X. Yang and X. Jiujiang W. Huirong. *Research on Direct Current Motor PID Control System Based on BP Neural Networks*. Information Engineering and Computer Science (ICIECS), 2010 2nd International Conference, pp. 1-4, Wuhan, 2010.
- [27] Demuth Hagan and Beale. *Neural network desing*. PWS Publishing Companing, 1995.
- [28] Simon Haykin. *Neural Networks: A Comprehensive Foundations*. Segunda Edición, Prentice Hall, 1944.
- [29] A. L. Hodgkin and A. F. Huxley. *Quantitative description of membrane current and its application to conduction and excitation in nerve*. J. Physiol, vol. 117, pp. 500-544, 1952.
- [30] J. J. Hopfield. *Pattern recognition computation using action potential timing for stimulus representation*. Nature, vol. 376, pp. 33-36, 1995.
- [31] E. M. Izhikevich. *Simple model of spiking neurons*. IEEE Transactions. Neural Networks, vol. 14, pp. 1569-1572, 2003.
- [32] E. M. Izhikevich. *Which model to use for cortical spiking neurons?* Neural Networks, IEEE Transactions, vol. 15, pp. 1063-1070, 2004.
- [33] Jose A. Santos John J. Wade, Liam J. McDaid and Heather M. Sayers. *SWAT: A Spiking Neural Network Training Algorithm for Classification Problems*. IEEE Transactions on neural networks, pp. 1817-1830, 2010.
- [34] T. M. Taha C. N. Vutsinas K. L. Rice, M. A. Bhuiyan and M. C. Smith. *FPGA implementation of izhikevich spiking neural networks for character recognition*. Reconfigurable Computing and FPGAs, 2009. ReConFig. International Conference, Quintana Roo, 2009.

- [35] I. Szkely L. Bako, S. T. Brassai and M. A. Baczo. *Hardware implementation of delay-coded spiking-RBF neural network for unsupervised clustering*. Optimization of Electrical and Electronic Equipment, 2008. OPTIM 2008. 11th International Conference, Brasov, 2008.
- [36] Lennart Ljung and Torkel Glad. *Modeling of Dynamic Systems*. Dit Mosco, 1994.
- [37] R. Jalasutram M. A. Bhuiyan and T. M. Taha. *Character recognition with two spiking neural network models on multicore architectures*. Computational Intelligence for Multimedia Signal and Vision Processing, 2009. CIMSVP. IEEE Symposium, Nashville, TN, 2009.
- [38] W. Maass. *Fast sigmoidal networks via spiking neurons*. Neural Computation, vol. 9, pp. 279-279, MIT Press, 1997.
- [39] W. Maass. *Networks of spiking neurons: The third generation of neural network models*. Elsevier, Neural Networks, vol. 10, pp. 1659-1671, 1997.
- [40] W. Maass and C. M. Bishop. *Pulsed Neural Networks*. Cambridge, Mass.: MIT Press, 1999.
- [41] Van Rossum M.C.W. *A novel spike distance*. Neural Computation vol. 13, num. 4, pp.751-763, 2001.
- [42] Katsuhiko Ogata. *Ingeniería de control moderno*. Cuarta Edición, Madrid, 2003.
- [43] Katsuhiko Ogata. *Ingeniería de Ontrol Moderna*. Ribera del Lioria, Madrid, pp. 745-798, 2003.
- [44] F. Ponulak. *ReSuMe - New supervised learning method for spiking neural networks*. Technical Report, Poznan, 2005.
- [45] Filip Ponulak. *ReSuMe - New supervised learning method for spiking neural networks*. Technical Report, Poznan, 2005.
- [46] Jhon G. Proakis and Dimitris G. Manolakis. *Tratamiento digital de señales*. Cuarta edición, Pearson educación, Madrid, 2007.
- [47] Yesim Oniz Rahib H.Abiyev, Okyay Kaynak. *Spiking Neural Networks for Identification and Control of Dynamic Plants*. International Conference on Advanced Intelligent Mechatronics, pp. 1030-1035, Kachsiung, 2012.
- [48] H. La Poutre S. M. Bohte and J. N. Kok. *Unsupervised clustering with spiking neurons by sparse temporal coding and multilayer RBF networks*. Neural Networks, IEEE Transactions, vol. 22, pp. 355-370, IEEE Computational Intelligence Society, 2002.

-
- [49] L. Dingding S. McKennoch and L. G. Bushnell. *Fast modifications of the Spike-Prop algorithm*. IEEE International Conference on Neural Networks-Conference Proceedings, Vancouver, BC, 2006.
- [50] A. Delorme S. Thorpe and R. Van Rullen. *Spike-based strategies for rapid processing*. Neural Networks, vol. 14, pp.715-725, 2001.
- [51] Joost N. Kok Sander M. Bohte and Han La Poutré. *Error-backpropagation in temporally encoded networks of spiking neurons*. Neurocomputing, Elsevier, vol. 48, pp. 17-37, 2002.
- [52] Benjamin Schrauwen and Jan Van Campenhout. *BSA, a Fast and Accurate Spike Train Encoding Scheme*. Neural Networks, 2003. Proceedings of the International Joint Conference on, vol. 4, pp. 2825-2830, 2003.
- [53] J. H. Schwartz, E. R. Kandel, and T. M. Jessell. *Principles of Neural Science*. McGraw-Hill, Health Professions Division, New York, 2000.
- [54] T. J. Sejnowski. *Time for a new neural code?* Nature, vol. 376, pp. 21-22, 1995.
- [55] Jean-Jacques E. Slotine and Weioing Li. *Applied Nonlinear Control*. Segunda edición, Englewood Cliffs, 1991.
- [56] Jean-Jacques E. Slotine and Weiping Li. *Applied Nonlinear Control*. Segunda edición, Englewood Cliffs, 1991.
- [57] Roberto A. Vazquez. *Training Spiking Neural Models using Cuckoo Search Algorithm*. Evolutionary Computation (CEC), 2011 IEEE Congress, vol. 7, pp.679-686, New Orleans, LA, 2011.
- [58] R. van Steveninck de Ruyter W. Bialek, F. Rieke and D. Warland. *Reading a neural code*. IEEE International Conference on Neural Networks Conference, vol. 252, pp. 1854-1857, 1991.
- [59] W. Gerstner W. M. Kistler and J. L. Van Hemmen. *Reduction of the Hodgkin-Huxley Equations to a Single-Variable Threshold Model*. Neural Computing, vol. 9, pp. 1015-1045, 1997.
- [60] G. Schmitger W. Maass and E. D. Sontag. *On the computational power of sigmoid versus boolean threshold circuits*. Proceedings of the 32nd Annual Symposium on Foundations of Computer Science, vol. 02, pp.767-776, San Juan, 1991.
- [61] Bogdan M. Wilamowski. *An Algorithm for Fast Convergence in Training Neural Networks*. International Joint Conference, vol.3, pp.1778-1782, Washington, DC, 2001.

- [62] Werner M. Kistler Wulfram Gestner. *Spiking neuron Models Single Neurons, Population, Plasticity*. Cambridge University Press, 2002.
- [63] R. Wen Y. Jin and B. Sendhoff. *Evolutionary multi-objective optimization of spiking neural networks*. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 1, pp. 370-379, Springer-Verlag Berlin Heidelberg, 2007.
- [64] Ayse Cisel Aras end Okyay Kaynak Yesim Oniz. *Control of Antilock Braking System Using Spiking Neural Networks*. Industrial Electronics Society, IECON 2013 - 39th Annual Conference of the IEEE, vol. 1, pp.3422-3427, Vienna, 2012.
- [65] You-Yin Jing Zhun Yu, Ying-Bai Xie and Xu-Ao Lu. *Applying neural networks to PID controllers for timedelay systems*. Machine Learning and Cybernetics, 2006 International Conference, pp. 3173-3176, Dalian, China, 2006.

Apéndice A

Desarrollo del control por redes neuronales de tercera generación

El objetivo de este apéndice es describir brevemente la evolución a través del desarrollo de esta tesis, para obtener el control por redes neuronales de tercera generación. La organización de este capítulo es la siguiente: En la Sección A.1 se da una breve explicación del contenido del anexo, en la Sección A.2 se realiza el entrenamiento de una neurona de una forma muy general sin codificación y el cual se divide en dos subsecciones, en la Subsección A.2.1 se describe el procedimiento para comenzar a trabajar con una neurona artificial de tercera generación, posterior a eso en la Subsección A.2.2 se explica el entrenamiento supervisado empleado, para la Sección A.3 se emplean los conocimientos de las secciones previas para identificar un sistema dinámico y finalmente en la Sección A.4 se concluye este anexo con el control aplicando una neurona artificial de tercera generación.

A.1. Introducción

Para la construcción de esta tesis, fue necesario seguir una secuencia lógica de pasos, desde la comprensión de la dinámica de los modelos de neuronas de tercera generación, hasta las características necesarias que debe poseer el entrenamiento neuronal para que el control PID-S propuesto sea eficaz.

A lo largo de este apéndice sólo se hará referencia sobre los métodos, entrenamientos y/o tipos de neuronas, sin ahondar más a fondo sobre ellos, puesto que lo que se desea es, sólo ilustrar el camino que se siguió para concluir con el resultado final esperado de la tesis.

A.2. Control de un valor real

Para atacar el problema del control de un sistema dinámico, se explica el cómo funcionan las redes neuronales de tercera generación, y en primer instancia se entrena una neurona para que la salida que emite ésta, sea igual a la salida deseada.

A.2.1. Inicialización de las variables

Primeramente se decide qué tipo de entrenamiento es el mejor, esto va directamente relacionado con las necesidades de la respuesta, es decir si lo que se desea es que el valor de la salida neuronal se aproxime de forma rápida al valor de la salida deseada o si la precisión de la salida de la neurona es imperativa sobre la velocidad de convergencia. Los entrenamientos que maneja esta tesis son de tipo supervisado, para este ejemplo en particular se implementa el entrenamiento SpikeProp, descrito en la sección 3.4.2.1

Para este ejemplo se eligió un modelo de alta abstracción, aunque es importante tomar en cuenta que mientras más abstracto sea el modelo, es más fácil su implementación computacional aunque carece de similitud biológica e incluso puede interferir en el desempeño de la red neuronal de forma negativa, para este ejemplo en particular se implementa el modelo PIF, de la ecuación (3.12), el cual tiene un nivel de abstracción alto.

Por último se adecua la forma en que se va a manejar la información, esto depende del proceso de aprendizaje, pues cada algoritmo de entrenamiento maneja la información de manera singular, en el caso del aprendizaje tipo SpikeProp se trabaja dentro una ventana de tiempo especificada previamente, en la cual varía el potencial de membrana según el número de pulsos que entren a la neurona, la disposición espacio temporal de los mismos y la fuerza de cada una de las conexiones y/o retardos (pesos sinápticos) entre las neuronas presinápticas y postsinápticas, en este caso la codificación es simplemente el valor temporal en el que ocurre el pulso, ejemplo figura A.1.

La ventana de tiempo propuesta para este ejemplo se propone de mil milisegundos.

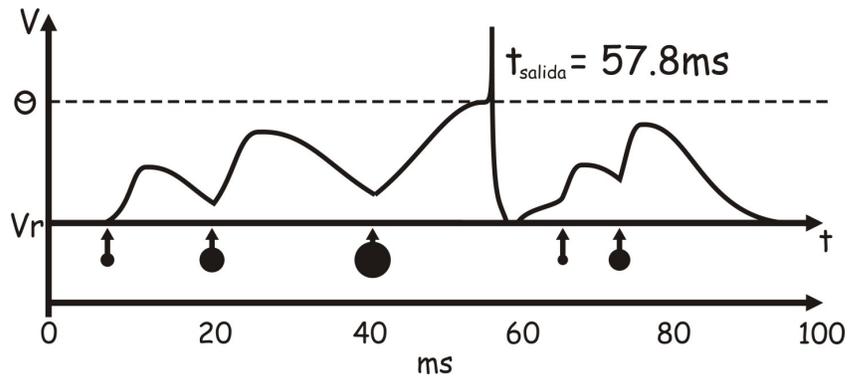


Figura A.1: En esta imagen se muestra la codificación espacio temporal directa, dentro de una ventana de tiempo, la neurona recibe una secuencia de pulsos de entrada, simbolizados por flechas y el valor de cada uno de los pesos sinápticos está representado por círculos de diferentes tamaños, cuando el umbral es rebasado por el potencial de membrana, la neurona dispara, el valor del pulso codificado es directamente el valor temporal en el que ocurre. en este caso el pulso vale 57.8 unidades.

A.2.2. Entrenamiento de una neurona

Una vez habiendo definido el tipo de entrenamiento, el modelo neurona y la codificación que se utilizará se puede realizar el primer ejemplo.

Un esquema general del entrenamiento de una neurona se puede apreciar en la figura(A.2) donde el error generado por la diferencia de la salida menos la entrada modifica los pesos, implementando un algoritmo de entrenamiento supervisado.

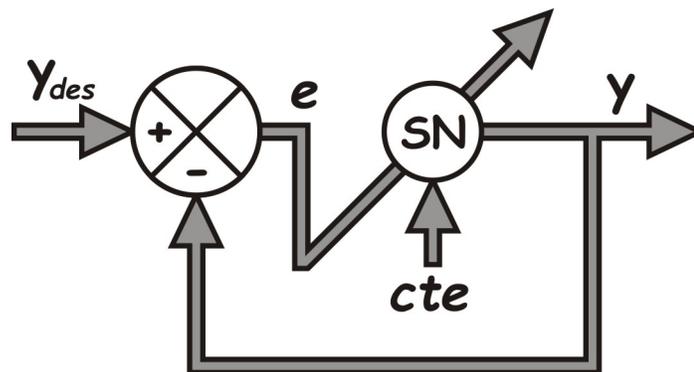


Figura A.2: Esquema general de entrenamientos supervisados con una sola neurona, bajo una entrada constante

A continuación se entrenará una neurona PIF, con un método de entrenamiento

SpikeProp bajo una entrada pre sináptica la cual se presenta constante a los cien milisegundos con ocho señales retardadas cada una con un retardo igual a cincuenta milisegundos como se muestra en la figura(3.4), en una ventana de tiempo de mil milisegundos, para que la neurona arroje un valor constante de $232ms$, y el kernel que transforma los pulsos de entrada a la neurona, en PSP es el que aparece en la ecuación 3.33 donde el valor de tiempo de subida $\tau_s = 25ms$. Esto se puede observar gráficamente en la figura(A.5). El entrenamiento se realizó durante treinta iteraciones, en las cuales el error converge a cero, tal como se muestra en la figura(A.4).

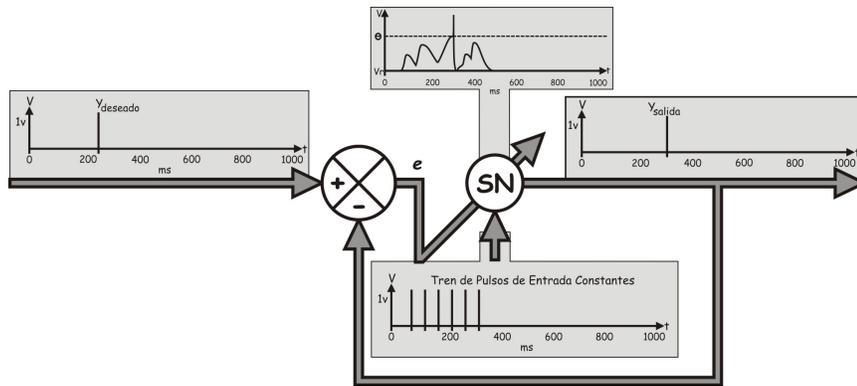


Figura A.3: En esta figura se muestra gráficamente el esquema de modificación de pesos sinápticos, en donde la señal de referencia o señal deseada es igual a doscientos treinta y dos milisegundos, el error es la diferencia entre el tiempo en que ocurre el pulso deseado y el tiempo en que la neurona emite el pulso de salida. La entrada de la neurona en este caso es una entrada constante que ocurre a los cien milisegundos con múltiples retardos de cincuenta milisegundos, los cuales en conjunto con el valor de los pesos sinápticos modifican el potencial de membrana.

En la figura(A.5) se muestra la evolución que presentan los nueve pesos presinápticos a través del proceso de entrenamiento, para este ejemplo se inicializaron todos los pesos en uno, aunque no es imperativo hacer esto, pues como se puede observar en [51], los pesos se pueden inicializar aleatoriamente. Todos los pesos convergen a algún valor y se estabilizan en éste, de acuerdo al comportamiento del error.

Finalmente, posterior al entrenamiento de la neurona, el valor del pulso de salida es igual al pulso deseado, y el potencial de membrana final se observa en la figura(A.6).

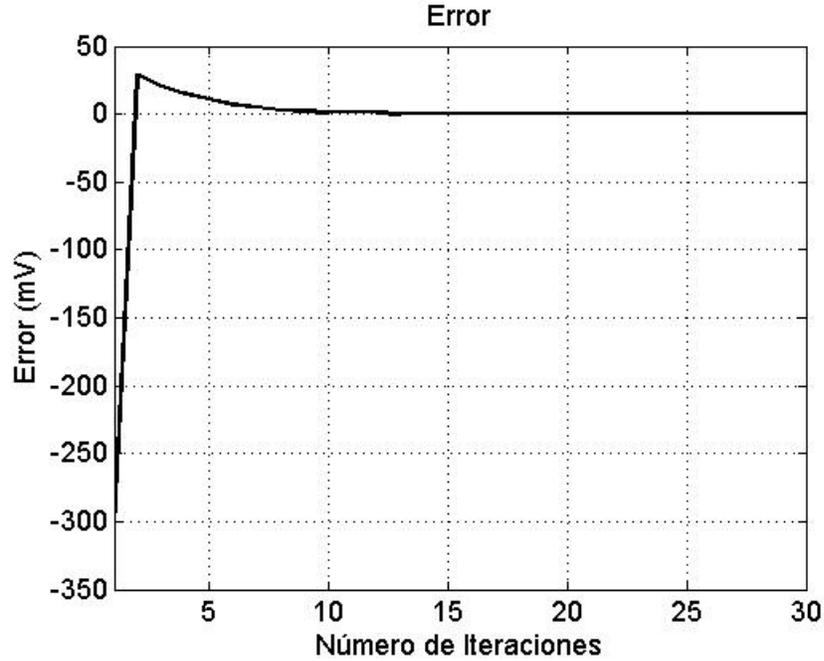


Figura A.4: Evolución del error de la salida neuronal, bajo el entrenamiento SpikeProp y una tasa de aprendizaje de .01 unidades.

A.3. Identificación de un sistema

Con el fin de evaluar el funcionamiento de las redes neuronales de tercera generación, se han realizado numerosos estudios en simulación, para el propósito de identificación y control de sistemas dinámicos.

El problema de identificación involucra la relación entre las entradas y salidas del sistema. En la figura(A.7) se muestra el esquema de identificación propuesto por [47]. La entrada a la neurona encargada de la identificación es la misma entrada externa del sistema, el error que modifica los pesos sinápticos es generado por la diferencia entre la salida $Y_n(k)$ propuesta por la neurona y la salida del sistema. La dificultad de la identificación por redes neuronales radica en determinar los pesos de la neurona para los cuales la diferencia entre la salida de la neurona $y_n(k)$ y la salida del sistema $y(k)$ sea mínima para cualquier entrada $u(k)$.

La siguiente planta no lineal representada por la ecuación A.1, es considerada como un ejemplo para el problema de identificación

$$x = 0.9038x + 0.9511 \left[1.02u \left(1 - \frac{1}{\exp(.75u)} \right) \right] \quad (\text{A.1})$$

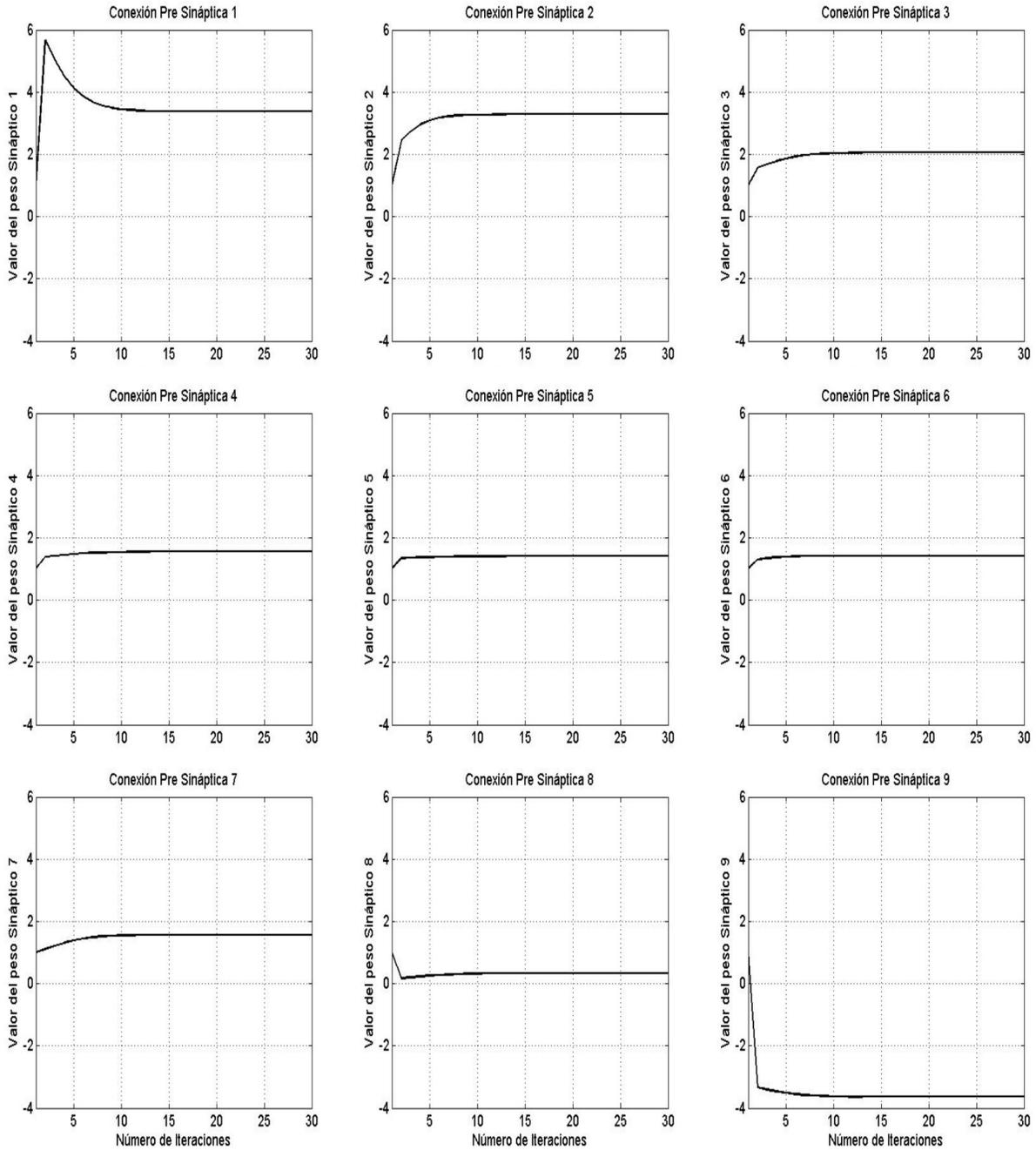


Figura A.5: Evolución de los nueve pesos presinápticos, inicializados en uno y con una tasa de aprendizaje de 0.01, durante un entrenamiento de treinta iteraciones.

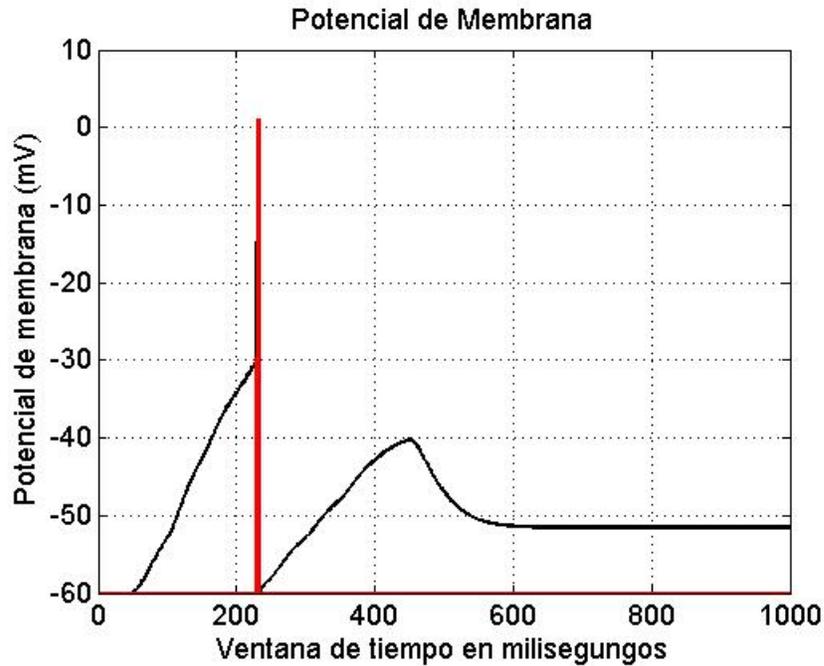


Figura A.6: Potencial de membrana final, de una neurona de tipo PIF, posterior al entrenamiento SpikeProp, bajo un tren de pulsos transformados por el kernel de la ecuación(3.33)

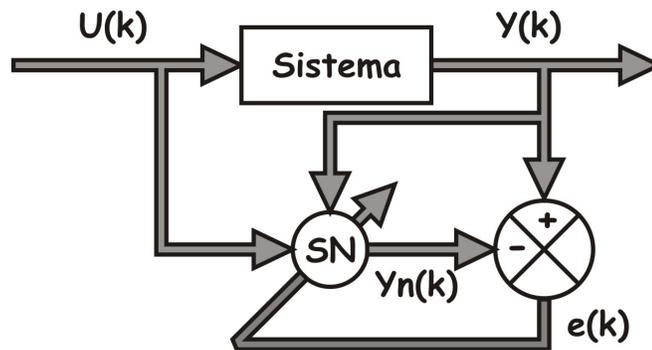


Figura A.7: El esquema de identificación para el entrenamiento SpikeProp maneja una entrada con múltiples retardos uniformes, seis por cada entrada, para este ejemplo en particular, esto quiere decir que la neurona recibe, $Y(k), Y(k - 1) \dots Y(k - 6)$, de igual forma la señal de control del sistema $U(k), U(k - 1) \dots U(k - 6)$.

Se utiliza una neurona para realizar el proceso de identificación, donde el estado actual de la planta depende tanto de los estados previos como de la entrada externa actual. La inicialización de los pesos presinápticos fue realizada al azar, en un intervalo cerrado de cero a uno.

En una gran cantidad de problemas en ingeniería se trabaja con valores analógicos, por lo que se debe encontrar un método eficaz de codificar la información del mundo real a un lenguaje apropiado para las redes neuronales de tercera generación. Este problema es resuelto con una codificación llamada, codificación por retardos implementada en [25],[11] y [52].

La codificación por retardos utilizada para relacionar las redes neuronales de tercera generación con el sistema a controlar, es una interpolación lineal [47], [64].

$$t_i(x) = t_{max} - \text{round}\left(t_{min} + \frac{(x_i - x_{min})(t_{max} - t_{min})}{(x_{max} - x_{min})}\right) \quad (\text{A.2})$$

y la ecuación que corresponde a la decodificación de la salida neuronal a la señal de entrada al sistema es [64]:

$$x_i = x_{min} + \frac{(t_{max} - t_{min} - t_i)(x_{max} - x_{min})}{(t_{max} - t_{min})} \quad (\text{A.3})$$

donde $t_i(x)$ es la variable de proceso del sistema dentro de la ventana de tiempo neuronal, t_{min} y t_{max} son el mínimo y máximo valor de la ventana de tiempo neuronal respectivamente, y los valores x , x_{min} y x_{max} son el valor real, mínimo y máximo de la variable de procesos correspondiente del sistema.

Usando la regla de aprendizaje, los pesos son actualizados usando la ecuación A.4 como una función de excitación al sistema, el entrenamiento de la neurona se lleva a cabo durante doscientas épocas con una ventana de tiempo de mil milisegundos por cada época, figura(A.8), la tasa de aprendizaje es de 0.003 y el modelo neuronal implementado es el propuesto por Izhikevich.

$$u(k) = \frac{.3(\sin(.07k) + .1\sin(.02k) + .65\sin(.09k))}{3.5} + .6 \quad (\text{A.4})$$

La función de error del entrenamiento durante la etapa de identificación se observa en la figura(A.9)

Una vez entrenada a la neurona Izhikevich se probó bajo diferentes entradas al sistema: ecuación(A.5).

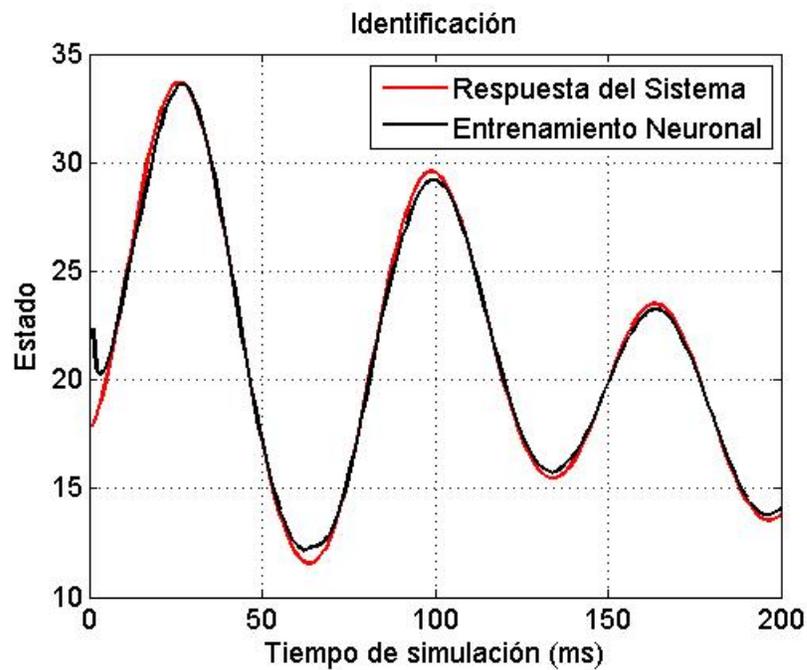


Figura A.8: Para entrenar a la neurona y que esta identifique al sistema se propuso la ecuación A.4, la cual excita al sistema por doscientos milisegundos, y durante ese tiempo los pesos neuronales son adaptados.

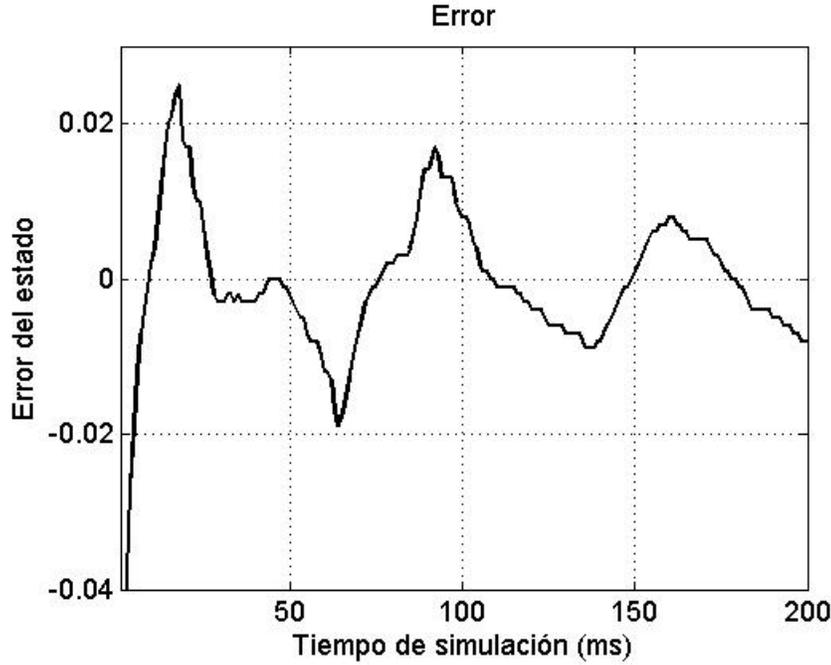


Figura A.9: La función de error que evoluciona durante el entrenamiento de identificación se minimiza mientras el tiempo de simulación aumenta.

$$u(k) = \begin{cases} .2\sin(.05k) + .60 & \text{si } 0 < k \leq 250 \\ .78 & \text{si } 250 < k \leq 500 \\ .42 & \text{si } 500 < k \leq 750 \\ \frac{.3(\sin(.07k) + .1\sin(.02k) + .65\sin(.09k))}{3.5} + .6 & \text{si } 750 < k \leq 1000 \end{cases} \quad (\text{A.5})$$

En la figura(A.10) se muestra la respuesta de la neurona Izhikevich (entrenada previamente) y la respuesta del sistema ambos excitados por la ecuación A.5, como se puede apreciar en la misma figura existe un error casi constante a lo largo de la simulación, esto se debe a que la neurona ya no modifica sus pesos, y sólo responde de acuerdo al entrenamiento que se realizó, en el sistema no lineal propuesto.

Finalmente en la figura(A.11) se muestra el potencial de membrana final, en la prueba de la identificación del sistema.

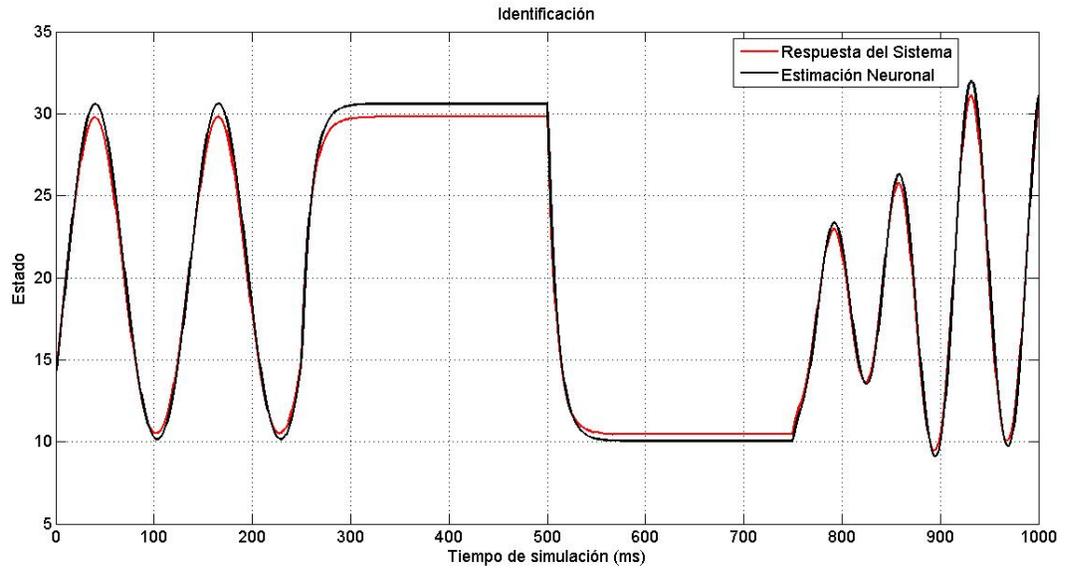


Figura A.10: Respuesta de la red neuronal, y del sistema bajo la misma entrada $u(k)$, para esta prueba los pesos sinápticos de la neurona no se modifican, sino que permanecen constantes, tal como quedaron posteriores al entrenamiento.

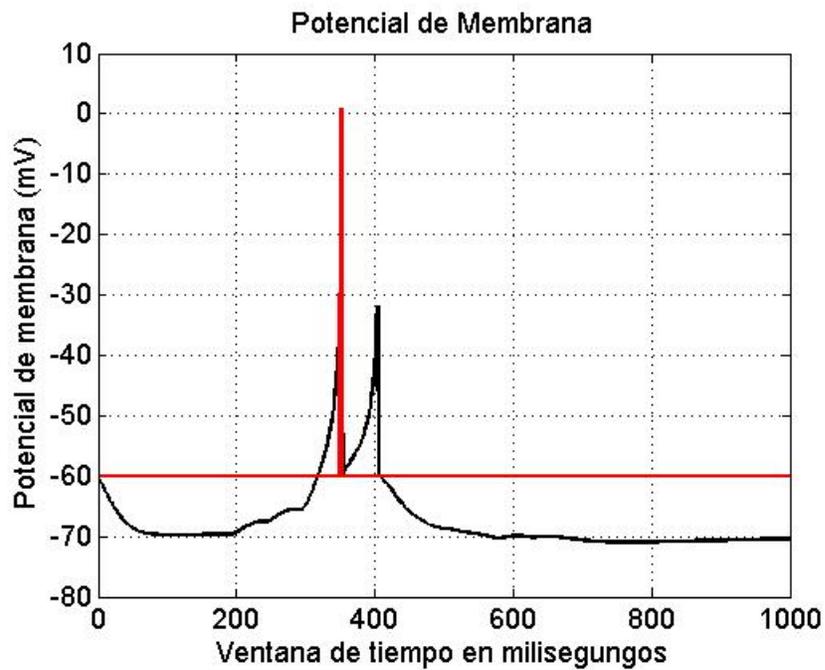


Figura A.11: Potencial de membrana final, de una neurona de tipo Izhikevich, posterior al entrenamiento de la identificación del sistema

A.4. Control de un sistema dinámico

Para realizar el control de un sistema dinámico se implementó el esquema mostrado en la figura(A.12), la entrada a la neurona es una constante y el entrenamiento se realiza con la señal de error, como se ha ido manejando a lo largo de esta tesis, solo que en este caso el error es generada por la diferencia entre la salida del sistema y la referencia deseada. Al igual que en la etapa de identificación, es necesaria una codificación y decodificación para entrelazar la neurona con el sistema y viceversa, por lo que se empleó la misma interpolación lineal que en la sección A.3.

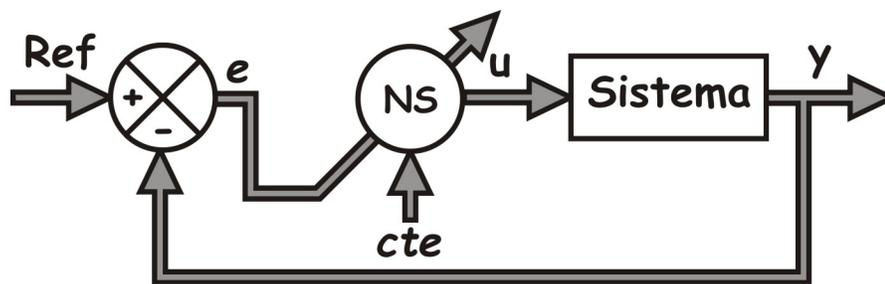


Figura A.12: Esquema general de control de un sistema dinámico utilizando entrenamientos supervisados con una sola neurona, bajo una entrada constante.

El entrenamiento que se utilizó para modificar los pesos sinápticos es aquel que implementa la relación de pulsos entrada salida descrito en la sección 3.4.2.2, la entrada constante a la neurona consta de múltiples retardos, tal como se muestra en la figura(3.4) los cuales abarcan toda la ventana de tiempo, esto con el propósito de que a pesar de ser una entrada constante la que tiene la neurona, no exista algún espacio dentro de la ventana de tiempo en el cual no pueda existir un pulso, de otra forma se limitaría la ley de control, la neurona implementada fue una neurona tipo LIF, con una tasa de aprendizaje igual a cero punto uno.

En la figura(A.13) se muestra la respuesta del sistema propuesto en la ecuación A.1, aplicando la ley de control neuronal, con tres referencias diferentes para probar qué tan susceptible es este tipo de control inteligente a los cambios, recordemos que sólo es una capa con una neurona la que realiza la tarea de control.

En la figura(A.14) y (A.15) se muestra el error del sistema y la ley de control a lo largo de la simulación respectivamente.

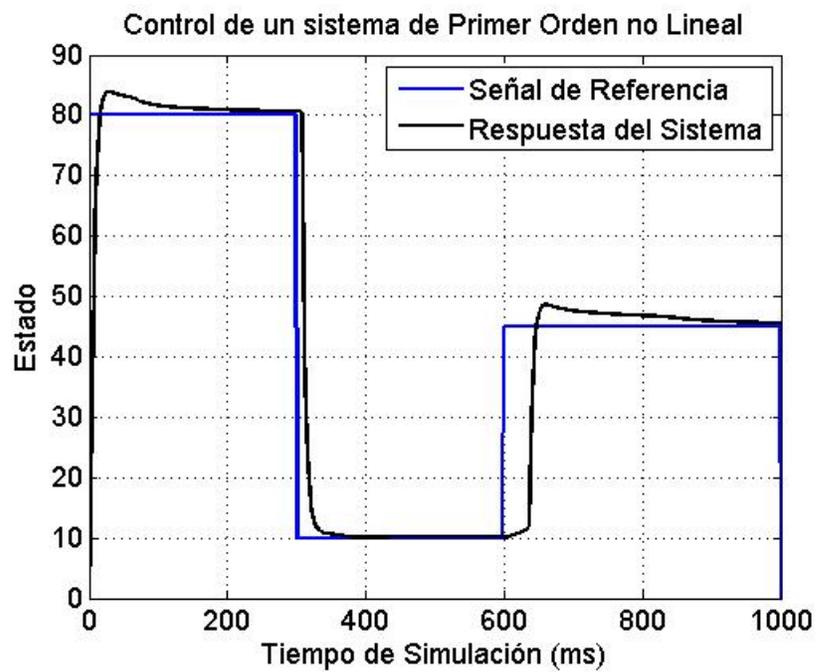


Figura A.13: Control de un sistema no lineal de primer orden, utilizando una neurona artificial de tercera generación tipo LIF y un entrenamiento de relación de pulsos de entrada salida, se observa cómo aunque la referencia cambia, la plasticidad de la neurona lleva al sistema a la referencia.

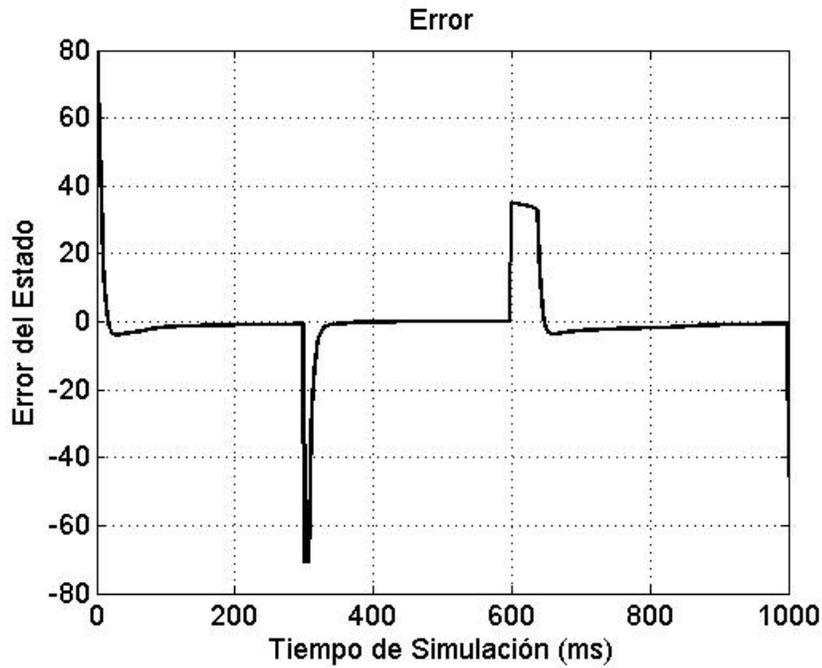


Figura A.14: En esta figura se muestra la señal de error del sistema ante, en todos los cambios de referencia el error converge a cero.

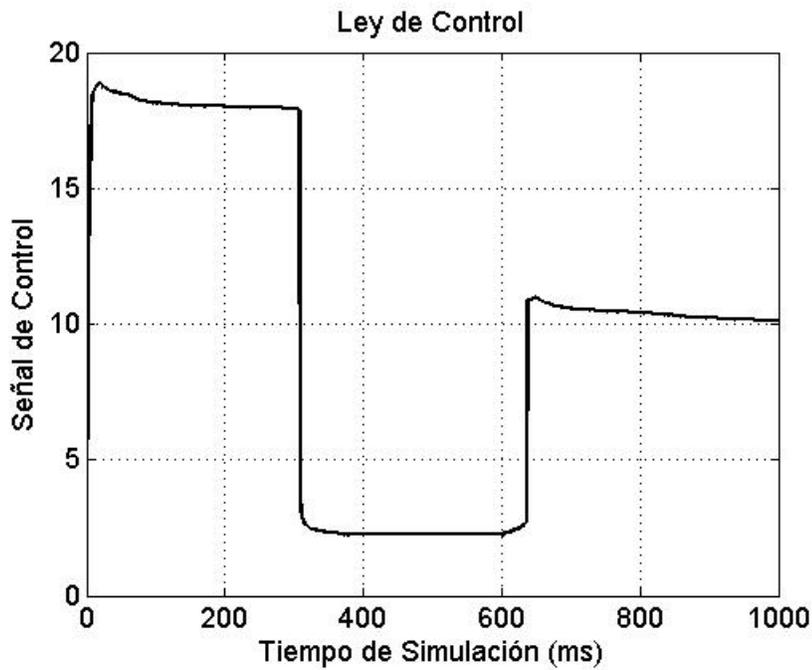


Figura A.15: En esta figura se aprecia la ley de control emitida por la neurona, esta señal fue codificada previamente para adaptarse al sistema.

Apéndice B

Comparación entre el PID-S y el PID

El objetivo de este apéndice es realizar una comparación entre el PID clásico y el PID-S, la cual sólo se realiza de una forma muy general, comparando la respuesta del sistema ante ambos controles, y comparando la ley de control emitida por cada uno de ellos. La organización de este capítulo es la siguiente: En la Sección B.1 se realiza una breve descripción de lo que es PID, en la Sección B.2 se muestran los resultados obtenidos en la simulación numérica, y en la Sección B.3 se describen los resultados obtenidos experimentalmente con la plataforma descrita en la sección 4.4.

B.1. Introducción

El control PID es uno de los controles más populares en la industria, debido a su eficiencia y simplicidad, en la literatura se han propuesto muchos tipos diferentes de reglas de sintonización, que permiten llevar a cabo una sintonización delicada y fina de los controladores PID en el sitio. se han desarrollado métodos automáticos de sintonización y algunos de los controladores PID poseen capacidad de sintonización automática en línea. Actualmente se usan en la industria formas modificadas del control PID, tales como el control I-PD y el control PID con dos grados de libertad. Es posible obtener muchos métodos prácticos para una conmutación sin choque (desde la operación manual hasta la operación automática) y una programación del aumento. El esquema de control del PID se muestra en la figura 4.1.

En este anexo se realiza una comparación entre el PID y el control PID-S. De la cual se muestran los resultados tanto en simulación numérica con una representación matemática linealizada de un motor de corriente directa, así como los resultados obtenidos físicamente en laboratorio.

Los experimentos llevados a cabo en simulación numérica, se realizaron utilizando los mismos parámetros y modelos matemáticos que se implementaron en el capítulo 4.

B.2. Resultados en simulación numérica

En la figura(B.1) se muestra la respuesta del sistema descrito por la ecuación 4.4 tanto para un control PID, como para el control PID-S. La referencia deseada se describe en la ecuación 4.3. El PID fue sintonizado por el método propuesto por Siegler-Nichols, donde las ganancias K_p , K_i y K_d son 0.005, 0.001 y 0.005 respectivamente.

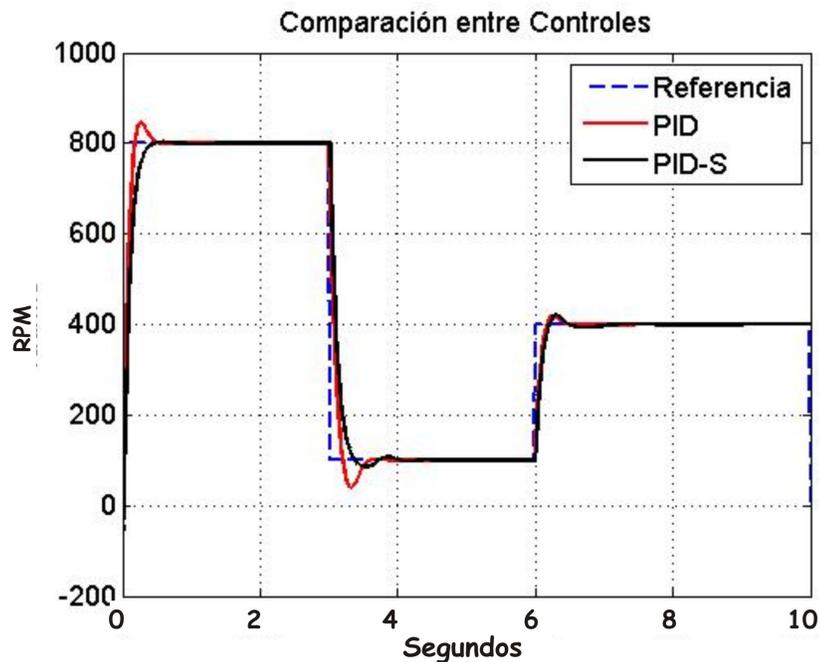


Figura B.1: En esta figura se muestra la respuesta de un mismo sistema con diferentes controles, la línea roja describe el comportamiento implementando una ley de control emitida por el PID clásico, sintonizado por Siegler-Nichols, y la línea negra utilizando el PID-S, con neuronas tipo PIF, finalmente la línea punteada describe la referencia deseada.

En la figura(B.2) se describe la comparación entre la ley de control del PID y la ley de control del PID-S, como se puede observar en la misma figura la salida del

control neuronal no demanda excesos de energía, en comparación con el PID, el cual llega hasta los 8.8v y -3.0433v.

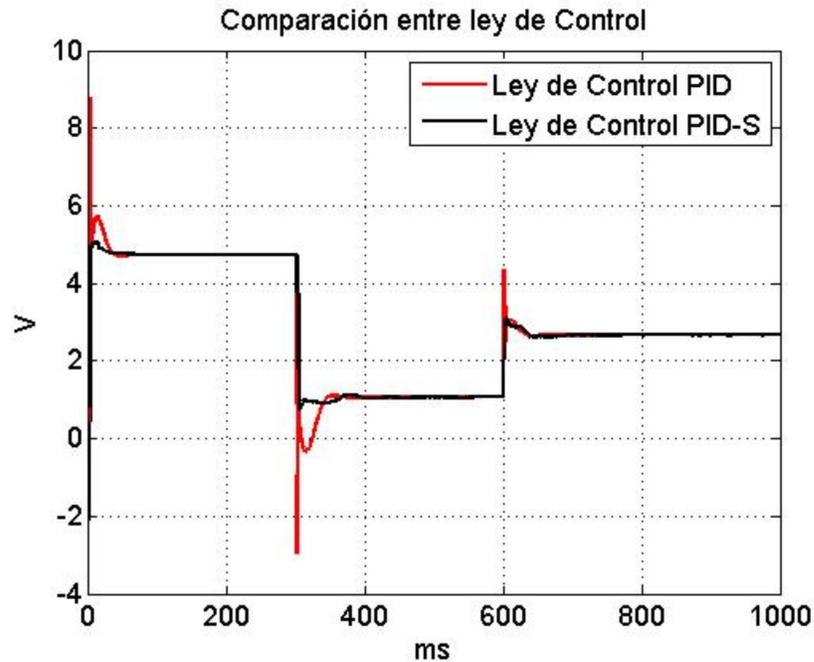


Figura B.2: En esta figura se muestra la comparación entre las leyes de control emitidas por el PID y el PID-S, la línea roja es la señal de control respuesta del PID, y la línea negra representa la señal emitida por el PID-S con neuronas tipo PIF.

B.3. Resultados de laboratorio

En la figura(B.3) se muestra la respuesta del motor de CD tanto para un control PID, como para el control PID-S. La referencia deseada se describe en la ecuación 4.3. El tiempo de muestreo es de un milisegundo y la plataforma implementada se describe en la figura 4.17

En la figura(B.2) se describe la comparación entre la ley de control del PID y la ley de control del PID-S, como se puede observar en la misma figura la salida del control neuronal no demanda excesos de energía, tal como ocurre en simulación, además de que la ley de control emitida por el PID-S es más suave que la emitida por el PID a pesar de que sea el mismo sistema.

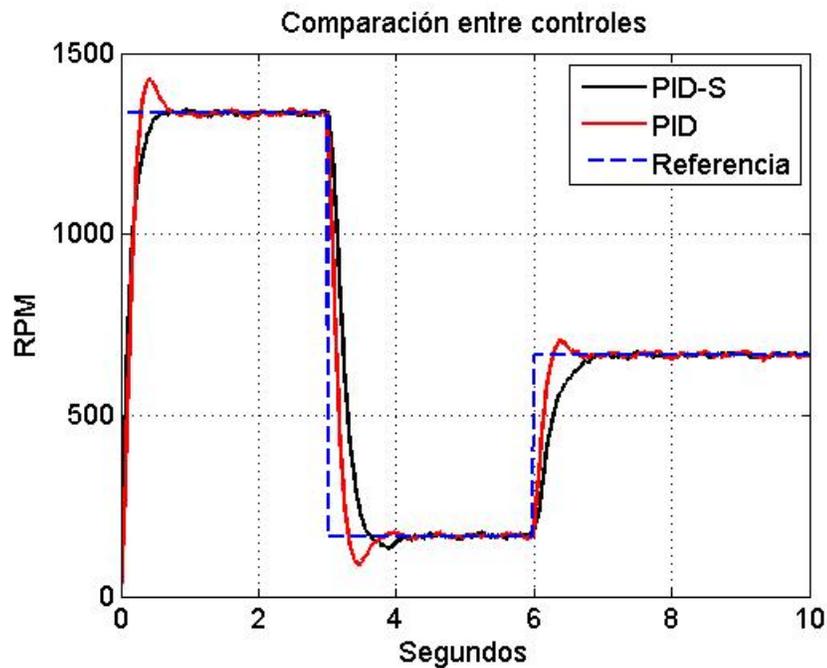


Figura B.3: En esta figura se muestra la respuesta de un motor de CD con diferentes controles, la línea roja describe el comportamiento implementando una ley de control emitida por el PID clásico, sintonizado por Siegler-Nichols, y la línea negra utilizando el PID-S, con neuronas tipo PIF, finalmente la línea punteada describe la referencia deseada.

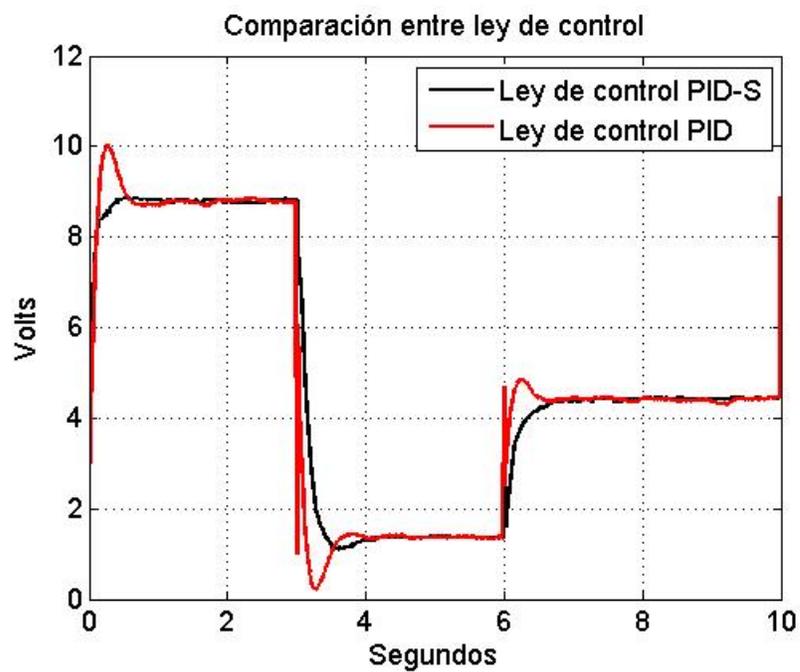


Figura B.4: En esta figura se muestra la comparación entre las leyes de control emitidas por el PID y el PID-S, la línea roja es la señal de control respuesta del PID, y la línea negra representa la señal emitida por el PID-S con neuronas tipo PIF.

Apéndice C

Artículo propuesto a revisión



Control Proporcional Integral Derivativo Basado en Redes Neuronales Spike

Luis Rey Lechucga¹, L. E. Ramos Velasco*, Joel Suárez Cansino

^aCentro de Investigación en Tecnologías de Información y Sistemas (CITIS), Universidad Autónoma del Estado de Hidalgo, Carretera Pachuca-Tulancingo, Km. 4.5, Mineral de la Reforma, Hidalgo, México. Tel/Fax:(+52) 7717172000 ext. 6734.

Resumen

En este artículo se propone la sinergia entre el control Proporcional, Integral y Derivativo (PID) y las redes neuronales de tercera generación. Se utiliza el error, su derivada temporal y su integral para entrenar a tres neuronas, donde la sumatoria de las salidas de éstas se convierte en la señal de control. La codificación y decodificación entre la información espacio-temporal de la red neuronal y la información del sistema se realiza bajo una interpolación lineal. El esquema de control propuesto es la contribución original de este artículo y se denota como PID-S, éste se prueba bajo una combinación de diferentes modelos de activación neuronal utilizando el entrenamiento *SpikeProp*, mostrando resultados sumamente interesantes en simulación numérica y de experimentos en laboratorio en los que se emplea como planta un motor de Corriente Directa (CD). Copyright©2014 CEA. Publicado por Elsevier España, S.L. Todos los derechos reservados.

Palabras Clave:

Controlador PID, Redes neuronales *spike*, Entrenamiento *SpikeProp*, Modelo Izhikevich, Modelo PIF, Motor de CD.

1. Introducción

Los controladores más utilizados actualmente en la industria son los controladores PID (Aström, 1997) o alguna de sus modificaciones, dada la facilidad de su implementación y los recursos tecnológicos que requiere. Sin embargo, el empleo del algoritmo PID lineal presenta limitaciones de robustez y convergencia cuando el proceso a controlar está definido por dinámicas complejas, zonas muertas y características altamente no lineales. En términos generales, el funcionamiento del controlador PID se basa en una actuación en forma proporcional, integral y derivativa sobre la señal de error $e(t)$, definida como la diferencia entre la señal de referencia $y_d(t)$ y la señal de salida del proceso $y(t)$, con la finalidad de acondicionar la señal de control $u(t)$ a partir de la suma de estas tres acciones de control, teniendo efectos de convergencia a la salida de la planta como se ve en la Figura 1.

En la misma figura, los parámetros k_p , k_i y k_d son las ganancias del controlador a ser sintonizadas. Existen distintas técnicas analíticas y experimentales con el fin de sintonizar estas ganancias (Aström and Hägglund, 2006). Una alternativa es

auto-sintonizar en línea dichas ganancias, tal como se hace en (Ramos-Velasco et al., 2013), donde se utilizan redes neuronales artificiales *wavenet* para identificar la planta y calcular esas ganancias.

La propuesta de control descrita en este trabajo resulta ser una técnica alternativa a las técnicas existentes, y en ella se hace uso de redes neuronales artificiales de tercera generación para llevar a cabo la tarea de control, sin hacer uso del modelo matemático de la planta, ni de los parámetros de la misma, y para demostrar esto se presentan resultados en simulación numérica y de experimentos en laboratorio para el seguimiento de trayectorias, para lo cual se emplea como planta un motor de CD.

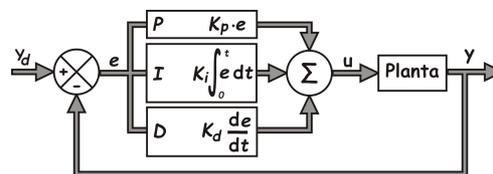


Figura 1: Esquema del controlador PID clásico

Las redes neuronales artificiales de tercera generación son los modelos que mejor emulan el comportamiento de las redes neuronales biológicas, este tipo de modelos tiene un sustento matemático basado en ecuaciones diferenciales, siendo algunas de ellas incluso no lineales (Gerstner and Kistler, 2002), por lo que de manera natural se ha propuesto la idea de integrar las re-

* Autor en correspondencia.

Correos electrónicos: l_1123581321@hotmail.com (Luis Rey Lechucga), lramos@uaeh.edu.mx (L. E. Ramos Velasco), jsuarez@uaeh.edu.mx (Joel Suárez Cansino)

¹Rey Lechuga actualmente es estudiante de la Maestría en Automatización y Control que se ofrece en el CITIS.

des neuronales artificiales de tercera generación a los sistemas dinámicos que se tiene interés en controlar, obteniéndose de esta forma un controlador adaptable, en el sentido de que el uso de este tipo de control requiere de herramientas matemáticas orientadas hacia el entrenamiento supervisado o no supervisado de la red.

Desde el punto de vista del aprendizaje supervisado, esto quiere decir que el error con el cual se adaptan los pesos sinápticos es generado por la diferencia entre la salida del sistema dinámico y la referencia deseada, por lo que la adaptabilidad del control a los requerimientos del sistema dinámico se logra a través de los métodos de entrenamiento supervisado apropiados a las redes neuronales artificiales de tercera generación.

La aplicación de las redes neuronales artificiales de pulso asume primeramente la definición de un modelo de neurona biológica, un modelo en el que se encuentre presente la idea de potencial de neurona, la definición de una codificación de información en pulsos, la definición de una arquitectura de red de neuronas y la definición de un algoritmo de entrenamiento apropiado. De acuerdo con esto, el artículo se encuentra organizado de la siguiente manera: la Sección 2 presenta un resumen sobre las redes neuronales artificiales de tercera generación, comentando brevemente los modelos neuronales más empleados en la literatura, así como el correspondiente método de entrenamiento supervisado. La Sección 3 presenta el esquema de control PID-S y en la Sección 4 se presentan los resultados obtenidos a través de una simulación numérica en lazo cerrado, empleando un modelo de un motor de CD, mientras que en la Sección 5 se presentan los resultados de los experimentos de laboratorio. En la Sección 6 se muestran los resultados experimentales al aplicar un control PID clásico haciendo una comparación de los resultados obtenidos entre los controladores PID clásico y el PID-S propuesto en este artículo. Por último, en la Sección 7 se presentan las conclusiones y el trabajo futuro.

2. Redes Neuronales de Tercera Generación

Las redes neuronales de tercera generación también conocidas como redes neuronales *spike* o de pulso, son una parte importante de las tendencias actuales dentro del campo de la Inteligencia Artificial y han sido usadas extensamente en diferentes tareas de ingeniería (Abiyev et al., 2012; Oniz et al., 2013a,b; Nichols et al., 2013). Wolfgang en uno de sus trabajos publicados (Jalasutram et al., 2009), ubica una red neuronal artificial en una de tres generaciones posibles, basándose en qué tanto se acerca ésta al comportamiento de una red de neuronas biológicas reales.

La primera generación de redes neuronales artificiales fue usada por McCulloch-Pits en 1943. En este modelo no existe la posibilidad de aprendizaje. Por otro lado, la segunda generación de redes neuronales artificiales está caracterizada por su capacidad de aprender, aunque sin considerar explícitamente en ello la forma en que se codifica la información que se transmiten entre sí las neuronas artificiales (Schnitger et al., 1991; Haykin, 1999). Finalmente, las redes neuronales artificiales de *Spike* son conocidas como redes neuronales artificiales de tercera generación ya que, adicionalmente a incluir las ideas y técnicas asociadas

a la segunda generación, lo hacen siguiendo un paradigma de transmisión de información interneuronal sumamente diferente a como lo hacen las dos generaciones anteriores. La idea esencial se basa en el hecho de que la forma del potencial de acción es siempre la misma, independientemente de la neurona que lo transmite, deduciendo con ello que la codificación de información no se basa en la geometría del potencial, sino seguramente en la frecuencia con que se activa éste, el tiempo en que un potencial emitido llega a cierto lugar, o bien por la forma en que se sincronizan en su activación una población de neuronas.

El primer esquema se denomina *frequency rate code* o *codificación basada en frecuencia*, el segundo se denomina *time-delay encoding* o *codificación basada en tiempo de llegada*, mientras que la tercera recibe el nombre de *synchronous population encoding* o *codificación basada en sincronización de población*. De esta forma, la información se maneja en forma espacio-temporal (Gerstner and Kistler, 2002).

2.1. Modelos neuronales

Los modelos matemáticos que representan el comportamiento de las neuronas de tercera generación se pueden dividir en dos grandes categorías de acuerdo a su nivel abstracción, el cual se encuentra definido por las propiedades computacionales que presenta cada uno (Gerstner and Kistler, 2002; Izhikevich, 2004). Así, existen modelos basados en conducta y modelos basados en umbral.

Los modelos basados en conducta incorporan el comportamiento detallado de los canales de iones de una célula biológica, tal como lo hace el modelo de Hodgkin-Huxley por ejemplo, mientras que los modelos de umbral representan el comportamiento dinámico de las neuronas biológicas reales de una manera más abstracta, en la que la neurona dispara cada vez que el potencial de la membrana sobrepasa un valor de umbral característico. En este artículo se emplea el modelo de umbral y disparo, por su relativa sencillez.

2.1.1. Modelos umbral y disparo

Por su fácil implementación y bajo costo computacional, el modelo de integración o umbral y disparo es uno de los modelos más extensamente usado (Rullen et al., 1999). 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 presinápticas al potencial de membrana, y si el potencial de membrana sobrepasa un umbral fijado previamente, entonces la neurona postsináptica dispara.

La ecuación diferencial de primer orden asociada al comportamiento de los modelos de integración y disparo describe la dinámica desde las condiciones de reposo del potencial de membrana hasta antes del momento en que la neurona dispara (instante en que se emite el pulso de salida), por lo que la condición de la actividad de disparo debe ser planteada. Esto se expresa a través de las condiciones (1), donde se representa el momento justo, el tiempo de disparo t^d , en el que el valor del potencial de membrana, $v(t)$, supera el umbral θ previamente especificado, lo cual es garantizado por el hecho de que su derivada en ese instante sea estrictamente positiva

$$t^{(f)} : v(t^{(f)}) = \theta \quad \text{y} \quad \left. \frac{dv(t)}{dt} \right|_{t=t^{(f)}} > 0 \quad (1)$$

Los modelos denominados Integración y Disparo Perfecto (Perfect Integrate and Fire, PIF) e Integración y Disparo con Resistencia (Leaky Integrate and Fire, LIF) pertenecen a esta categoría.

El modelo PIF es sumamente sencillo y generalmente se emplea con fines de introducir las características mínimas necesarias para el funcionamiento de una neurona de pulso. Su análisis se lleva a cabo asumiendo que consiste únicamente de un circuito capacitivo; sin embargo, esta misma sencillez hace interesante investigar su desempeño en un ambiente de control como se verá más adelante. Por su misma definición, la capacitancia resulta ser el parámetro de importancia.

Por otro lado, el modelo LIF es más complejo que el modelo PIF ya que el análisis de su funcionamiento se lleva a cabo sobre un circuito RC en paralelo. En este caso, la capacitancia y la resistencia son los parámetros de importancia.

Por supuesto, en estos y en otros casos se tienen que definir también parámetros adicionales como son, por ejemplo, el potencial de reposo, el potencial de umbral, entre otros.

2.1.2. El modelo Izhikevich

Izhikevich (Izhikevich, 2003) propuso 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 (Gerstner and Kistler, 2002) y la eficacia computacional de los modelos umbral y disparo. Esto quiere decir que se reduce el modelo de Hodgkin-Huxley, el cual es más real pero con mayor demanda de recursos computacionales, a un modelo de dos ecuaciones diferenciales acopladas de primer orden, dadas por:

$$\dot{v} = 0.04v^2 + 5v + 140 - z + I \quad (2)$$

$$\dot{z} = a(b \cdot v - z) \quad (3)$$

La variable v representa el potencial de membrana de la neurona y z representa la variable de recuperación de membrana, la cual está definida por la corriente de iones de activación de potasio, K , y la inactivación de la corriente de iones de sodio, Na . Así mismo, I representa la señal. Este modelo puede exhibir las propiedades más importantes sobre el comportamiento neuronal con las variables apropiadas a , b , c y d , como se muestra en la Figura 2. Este modelo tiene un umbral dinámico que depende del estado previo del potencial de membrana anterior al pulso. Los parámetros a , b , c , d son completamente especificados y están relacionados con algunas propiedades físicas y químicas de la neurona biológica.

El modelo de Izhikevich dado por (2) y (3) es empleado en cada una de las neuronas *spike* (NS) del esquema de control de la Figura 3.

2.2. Métodos de Aprendizaje

Tomando en cuenta que el entrenamiento supervisado se lleva a cabo en línea, en este trabajo los criterios seguidos para adoptar un algoritmo de aprendizaje supervisado en particular

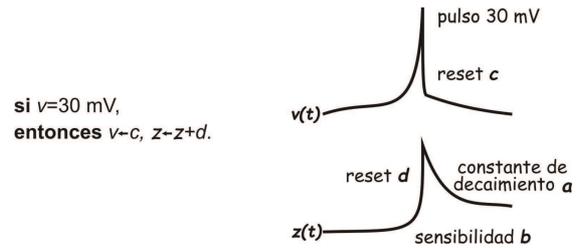


Figura 2: Comportamiento del modelo de Izhikevich dado por las ecuaciones (2) y (3). Si el potencial de membrana de la neurona v es igual a 30 mV, entonces se presenta un pulso y la variable v toma el valor de c , además la variable de recuperación de la membrana z toma el valor de $z + d$

consideran características de convergencia y estabilidad, así como el comportamiento mostrado en las simulaciones numéricas para el control de sistemas una entrada-una salida.

2.2.1. SpikeProp

Durante el entrenamiento y en la iteración k , los pesos w_{ij}^k entre la neurona pre-sináptica i y la neurona post-sináptica j no necesariamente han adquirido valores estables. El cambio de estos valores está directamente relacionado con la salida neuronal y el manejo de información por trenes de pulsos. Si la sinápsis se fortalece, entonces se le conoce como LTP (Long Term Potentiation) y si, por lo contrario, la sinápsis se debilita, entonces 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 estar dentro de una de dos posibles categorías; a saber, aprendizaje supervisado o no supervisado. En este artículo sólo se emplea el aprendizaje supervisado.

Algunos investigadores (Schwartz et al., 2000) han propuesto 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 artificiales con múltiples conexiones, las cuales tienen múltiples retardos. La arquitectura de la red es del tipo feedforward, con una o más capas ocultas.

Sin embargo, estos investigadores (Schwartz et al., 2000) implementan una variante del Modelo de Respuesta al Pulso (SRM) para describir el potencial de membrana producido por el conjunto de pulsos presinápticos; a saber, el tren de pulsos asociado a cada respuesta neuronal no incluye un término de refracción.

El propósito del algoritmo de entrenamiento propuesto por ellos, es que la red neuronal sea entrenada de acuerdo al error que se genera por la diferencia entre los tiempos de disparo específicos que codifican a las salidas deseadas, y los tiempos de disparo específicos producidos por la red en la salida durante la iteración k en el entrenamiento. Los detalles del algoritmo se pueden encontrar en la referencia (Schwartz et al., 2000).

2.2.2. Regla modificada de Widrow-Hoff o Regla Delta Generalizada a Spikes

Similar a otros algoritmos de entrenamiento supervisado, los pesos sinápticos de la red neuronal se ajustan iterativamente basados en una relación de entrada-salida deseada (Mohammed et al., 2011). Se comienza a partir de la regla Widrow-Hoff común para la modificación del peso de una sinápsis i , en otras palabras

$$\Delta w_i^{WH} = \lambda x_i (y_d - y) \quad (4)$$

donde $\lambda \in \mathbb{R}$ es un coeficiente de aprendizaje positivo con valores reales, x_i es la entrada a través de la neurona pre-sináptica por medio de la sinápsis i y, finalmente, y_d e y se refieren a la salida deseada y a la salida real emitida por la neurona, respectivamente. Debe recordarse que estas entradas y salidas están dadas por una codificación convenientemente definida por los potenciales de membrana.

Mediante la integración de Δw_i^{WH} se obtiene una Δw_i escalar que se utiliza para actualizar el peso de la sinápsis i :

$$\Delta w_i = \lambda \int x_i(t)(y_d(t) - y(t))dt \quad (5)$$

Los pesos se actualizan a lo largo de un proceso iterativo, usando la regla de aprendizaje $w_i(\mathbf{e} + 1) = w_i(\mathbf{e}) + \Delta w_i$, donde \mathbf{e} es la época actual del proceso de aprendizaje.

En términos generales, el entrenamiento compara trenes de pulsos en el tiempo (deseado y actual), para lo cual asume un potencial de disparo de la forma (Rossum, 2001),

$$\alpha(t) = e^{\frac{1}{\tau_s} t e^{-\frac{t}{\tau_s}}} \Theta(t) \quad (6)$$

donde $\Theta(t)$ se refiere a la función de Heaviside y τ_s es el tiempo donde ocurre el valor máximo de la función sináptica. Finalmente, para completar la actualización de los pesos se aplica la ecuación (5).

3. PID-S

Los modelos matemáticos que representan el comportamiento de las neuronas biológicas son en su mayoría formulaciones no lineales (Schwartz et al., 2000), y los algoritmos de aprendizaje supervisado existentes (Maass and Bishop, 1999; Ponulak, 2005) funcionan sin mayores problemas con ellos. De acuerdo con esto, surgió la idea de controlar sistemas dinámicos con entrenamientos supervisados, diseñados exclusivamente para redes neuronales de tercera generación. Ciertamente los entrenamientos no fueron diseñados para el control de sistemas dinámicos, para esto se contemplan las propiedades del control lineal PID clásico (Ogata, 2003), dando como resultado el esquema de control PID-S mostrado en la Figura 3, cuyo esquema de control es una de las contribuciones originales de este artículo.

De la Figura 3 se puede observar que la entrada a cada una de las tres neuronas es la señal de error, la integral del error y la derivada del error, respectivamente. Un análisis más detallado del proceso de codificación con múltiples retardos y decodificación se muestra en la Figura 4. La codificación que se utiliza

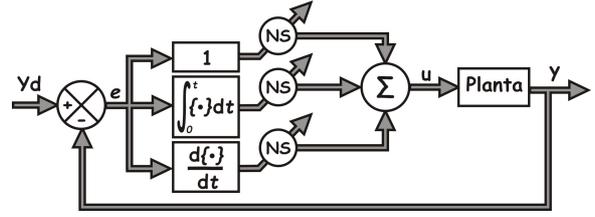


Figura 3: Esquema de control PID-S propuesto en este artículo

para relacionar las redes neuronales de tercera generación con el sistema a controlar, está definida por la interpolación lineal (Abiyev et al., 2012; Oniz et al., 2013a),

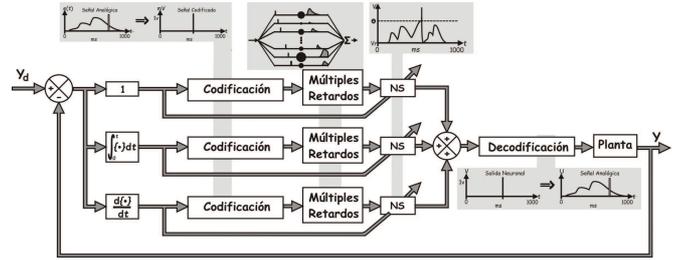


Figura 4: Diagrama a bloques del proceso de codificación y decodificación empleadas en el algoritmo de control

$$t_i(x_e) = t_{max} - \text{round} \left(t_{min} + \frac{(x_e - x_{emin})(t_{max} - t_{min})}{(x_{emax} - x_{emin})} \right) \quad (7)$$

donde round es la función de redondeo realizada en los intervalos de las ventanas de tiempo, dichas ventanas son definidas por el diseñador tomando en cuenta qué tan suave se quiere la señal de control y con qué hardware se cuenta para la realización de los cálculos, además $t_i(x_e)$ es tiempo donde se emite el pulso dentro de la ventana de tiempo neuronal, t_{min} y t_{max} son el mínimo y máximo valor de la ventana de tiempo neuronal respectivamente, y los valores x_e , x_{emin} y x_{emax} son el valor actual, mínimo y máximo del error, integral del error y derivada del error para cada uno de los bloques de codificación correspondiente.

La ecuación que corresponde a la decodificación de la salida neuronal a la señal de entrada a la planta es (Oniz et al., 2013a):

$$u_i = u_{min} + \frac{(t_{max} - t_{min} - t_i)(u_{max} - u_{min})}{(t_{max} - t_{min})} \quad (8)$$

donde u_i , u_{min} y u_{max} son el valor actual, mínimo y máximo de la señal de control que actúa sobre la planta.

Algo importante de resaltar es que antes de comenzar la secuencia de control, la salida de la neurona debe inicializarse en el cero codificado, es decir la señal de control inicial emitida por cada una de las tres neuronas debe ser cero.

4. Resultados en Simulación Numérica

El esquema de control PID-S que se propone en este artículo se utiliza para controlar un motor de CD en una simulación

numérica y en experimentos de laboratorio, la selección de este motor como planta a controlar se debe principalmente a la facilidad de encontrarlo en el mercado, además de ser ampliamente utilizado en la industria moderna como actuador en los procesos industriales (Klafter et al., 1989). Es importante destacar que en la simulación y experimentación se realizaron diferentes pruebas cambiando los modelos neuronales y los métodos de aprendizaje supervisado.

La representación entrada-salida del motor de CD está dada por la función de transferencia siguiente

$$\frac{Y(s)}{U(s)} = \frac{83.3}{9.98s + 0.997} \quad (9)$$

la cual se obtiene experimentalmente aplicando una entrada de 4 volts al motor de CD. Como se puede observar, es un sistema lineal de primer orden estable.

4.1. Entrenamiento SpikeProp

En las Figuras 5 y 6 se muestra la respuesta del sistema utilizando el control PID-S bajo el entrenamiento *SpikeProp* y los modelos neuronales PIF e Izhikevich, respectivamente. Los valores de entrenamiento de cada terna de neuronas en ambas pruebas son -0.2 , -0.01 y -0.001 , para cada una de las tres neuronas PIF entrenadas con e , $\frac{de}{dt}$ y $\int edt$, respectivamente; mientras que los valores -0.05 , -0.0008 y -0.00001 son para cada una las tres neuronas Izhikevich entrenadas con e , $\frac{de}{dt}$ y $\int edt$, respectivamente. Todos los valores de entrenamiento son dados de manera arbitraria sin considerar la dinámica de la planta a controlar.

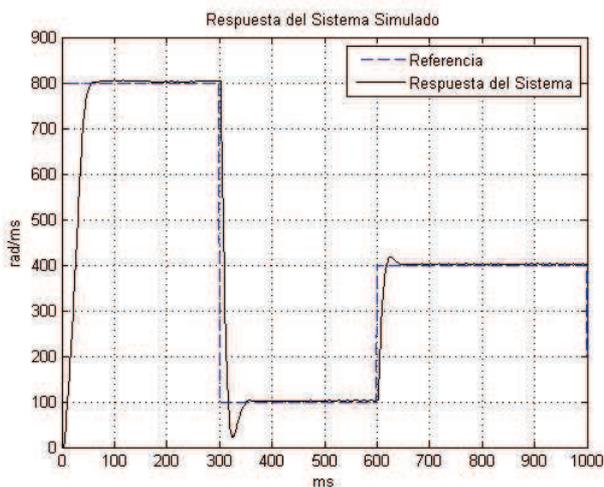


Figura 5: Control del sistema utilizando un modelo PIF

A partir de la Figura 5 se muestran los resultados en lazo cerrado empleando el control PID-S con el modelo PIF, de donde se puede observar que al paso del tiempo las oscilaciones decrecen, esto se debe al aprendizaje del PID-S propuesto, dando como resultado que el error de seguimiento mostrado en la Figura 7 tiende a reducirse en los instantes de cambio de la señal de referencia. Este mismo fenómeno se presenta cuando se emplea el modelo Izhikevich cuyos resultados están dados en las

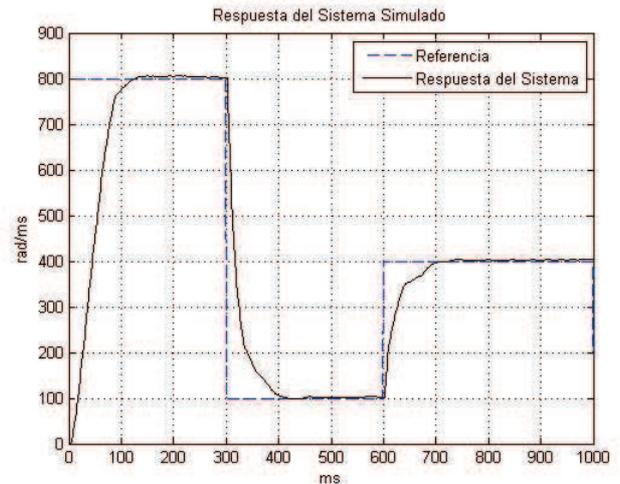


Figura 6: Control del sistema utilizando un modelo Izhikevich

Figuras 6 y 8. Las señales de control para ambos modelos PIF e Izhikevich son mostradas en las Figuras 9 y 10, respectivamente. Ambas señales están acotadas y no presentan oscilaciones que puedan dañar al motor de CD con el paso del tiempo.

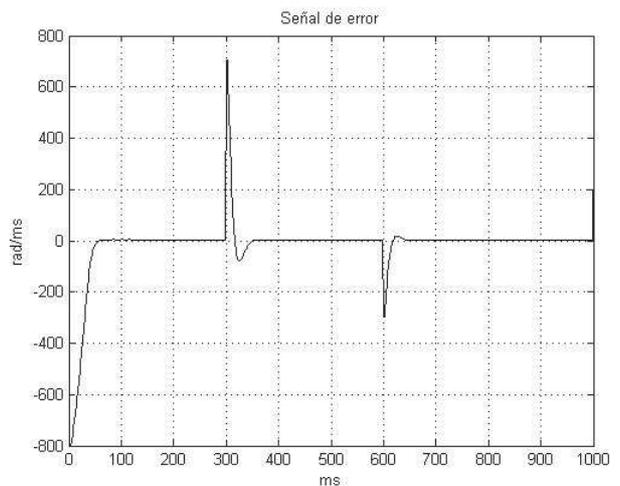


Figura 7: Señal de error utilizando un modelo PIF

5. Resultados de Laboratorio

En la Figura 11 se muestra la plataforma experimental, que consta de un motor de corriente directa, una tarjeta de adquisición de datos de National Instruments, una fuente de poder y una computadora laptop marca Vaio Sony modelo SVF142C29U. El periodo de muestreo empleado en el control es de un milisegundo (ms).

5.1. Entrenamiento SpikeProp

En las Figuras 12 y 13 se muestra la respuesta del sistema para cambios en la señal de referencia, para esto se emplea el

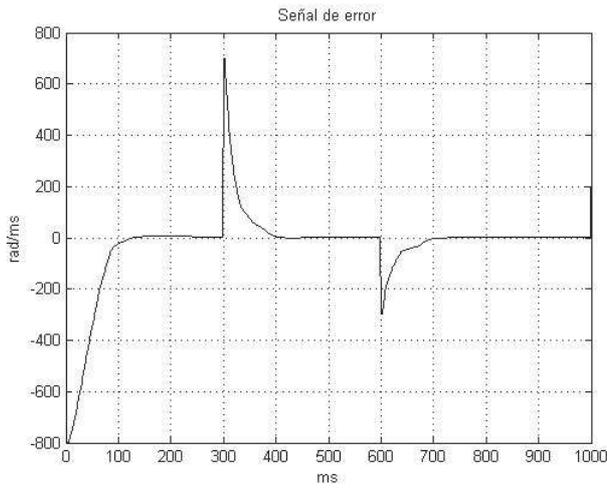


Figura 8: Señal de error utilizando un modelo Izhikevich

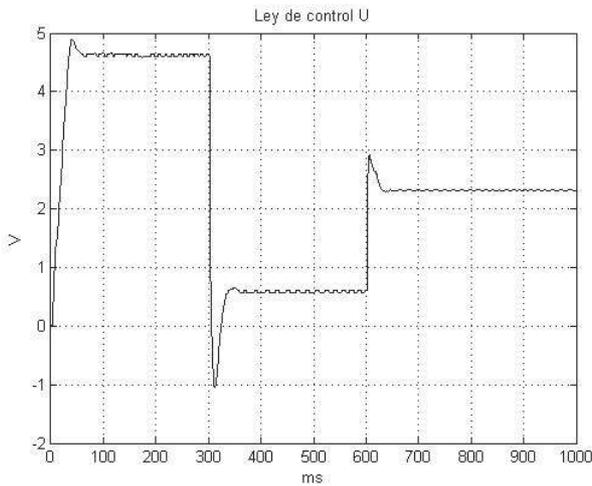


Figura 9: Señal de control utilizando un modelo PIF

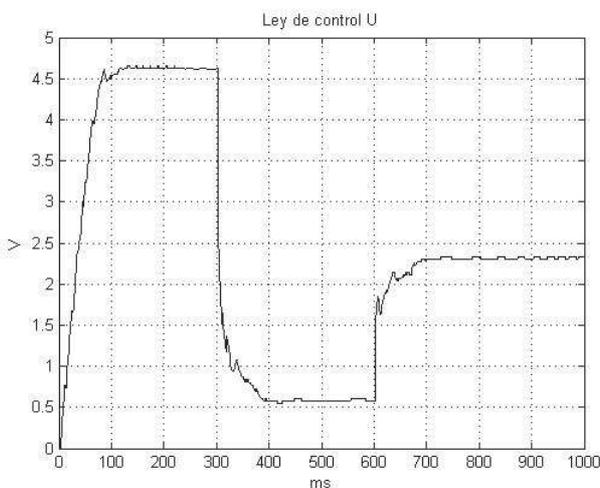


Figura 10: Señal de control utilizando un modelo Izhikevich

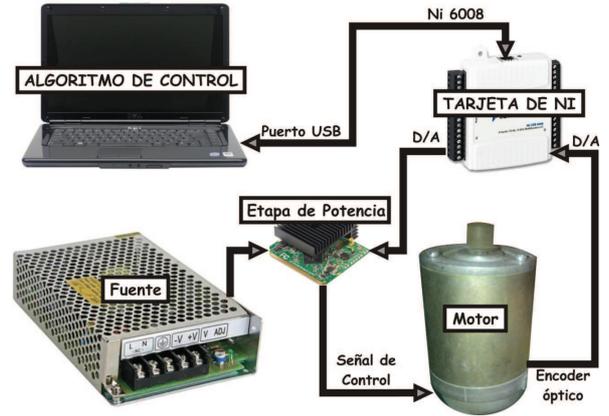


Figura 11: Plataforma experimental

control PID-S bajo el entrenamiento SpikeProp y los modelos neuronales PIF e Izhikevich, respectivamente. Al igual que en las simulaciones numéricas, para la aplicación en el laboratorio se emplean los valores de entrenamiento de cada terna de neuronas los que, en ambas pruebas, son -0.2 , -0.01 y -0.001 para las tres neuronas PIF entrenadas con e , $\frac{de}{dt}$ y $\int edt$, respectivamente y -0.05 , -0.0008 y -0.00001 para las tres neuronas Izhikevich entrenadas con e , $\frac{de}{dt}$ y $\int edt$, también respectivamente. Dichos valores son dados de manera arbitraria sin considerar ninguna condición del sistema bajo control. Las señales de error para cada uno de los casos anteriores se presentan en las Figuras 14 y 15, mientras que las señales de control se dan en las Figuras 16 y 17.

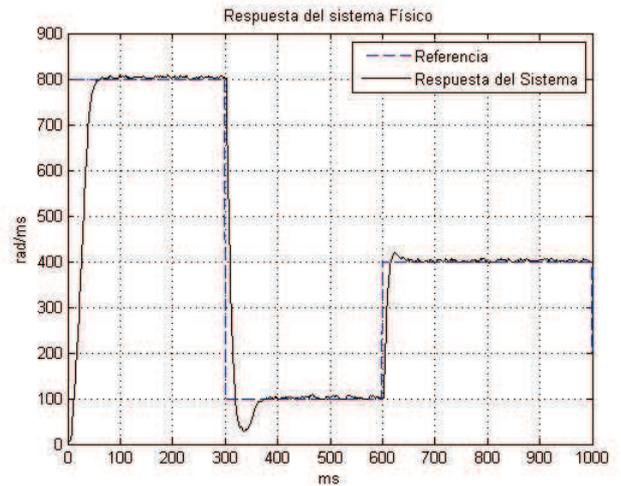


Figura 12: Respuesta del sistema utilizando un modelo PIF

5.1.1. Entrenamiento mediante la regla modificada de Widrow-Hoff

Aquí se presentan los resultados experimentales obtenidos en el laboratorio cuando se emplea el entrenamiento mediante la regla modificada de Widrow-Hoff. Se consideran los casos de las neuronas LIF e Izhikevich.

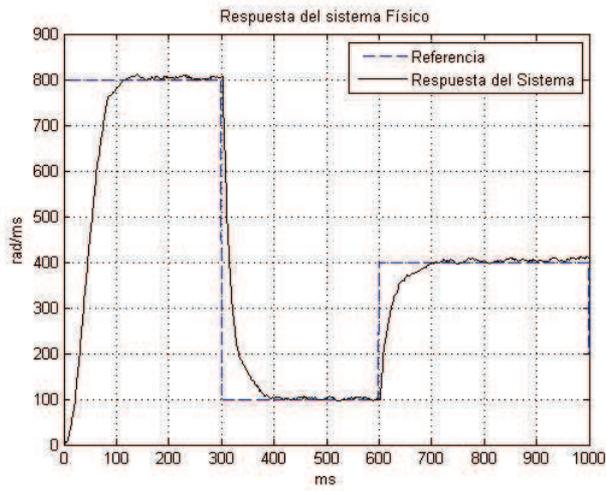


Figura 13: Respuesta del sistema utilizando un modelo Izhikevich

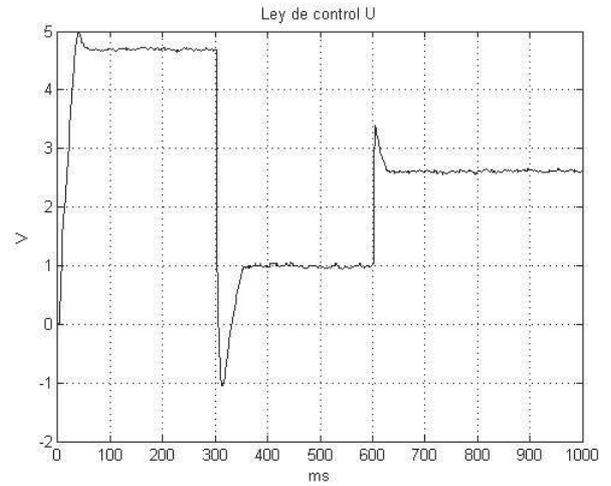


Figura 16: Señal de control utilizando un modelo PIF

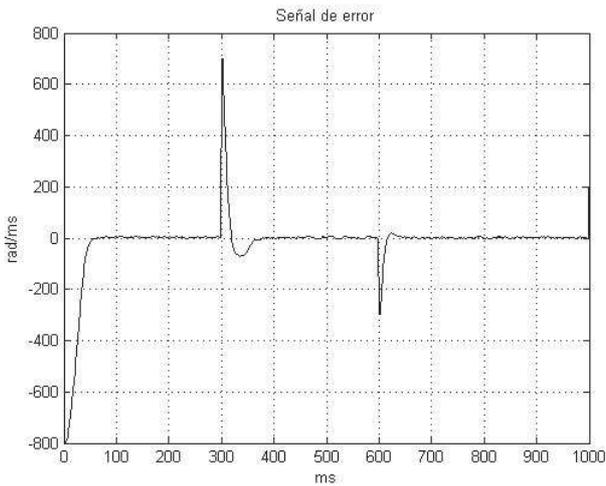


Figura 14: Señal de error utilizando un modelo PIF

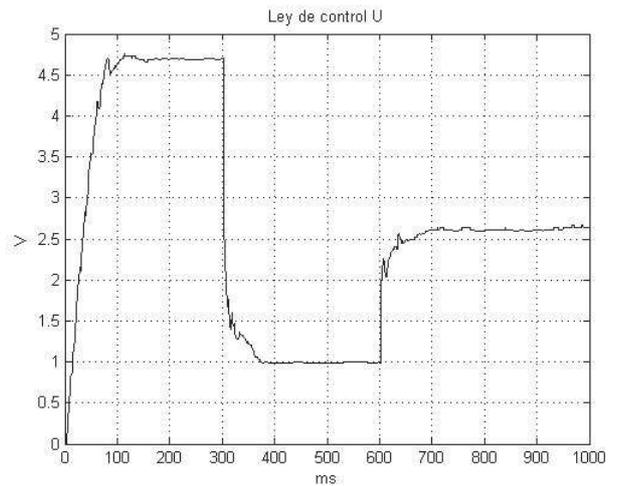


Figura 17: Señal de control utilizando un modelo Izhikevich

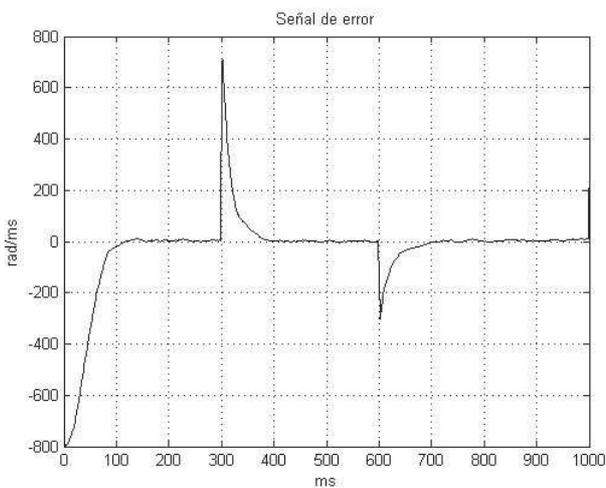


Figura 15: Señal de error utilizando un modelo Izhikevich

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

En las Figuras 18 y 21 se muestra la respuesta del sistema para cambios en la señal de referencia, para esto se emplea el control PID-S bajo el entrenamiento mediante la regla modificada de Widrow-Hoff y los modelos neuronales LIF e Izhikevich, respectivamente. Se emplean los valores de las tasas de aprendizaje de cada terna de neuronas, son 1, 0.05 y 0.001 para las tres neuronas LIF entrenadas con e , $\frac{de}{dt}$ y $\int edt$, respectivamente y 0.01, 0.001 y 0.0001 para las tres neuronas Izhikevich entrenadas con e , $\frac{de}{dt}$ y $\int edt$, también respectivamente. Dichos valores son dados de manera arbitraria sin considerar ninguna condición del sistema bajo control. Las señales de error para cada uno de los casos anteriores se presentan en las Figuras 19 y 22, mientras que las señales de control se dan en las Figuras 20 y 23.

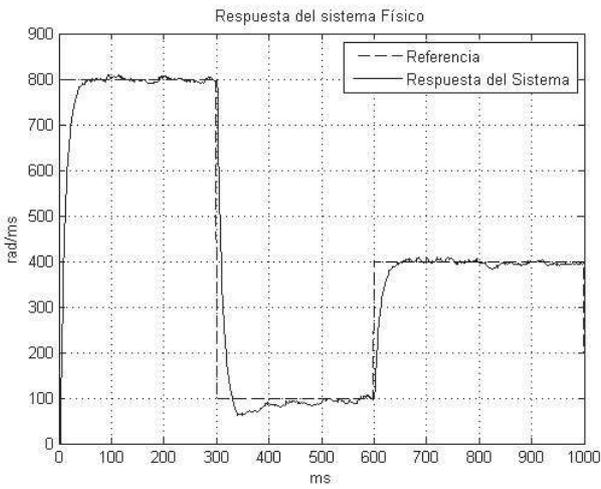


Figura 18: Respuesta del sistema utilizando un modelo LIF

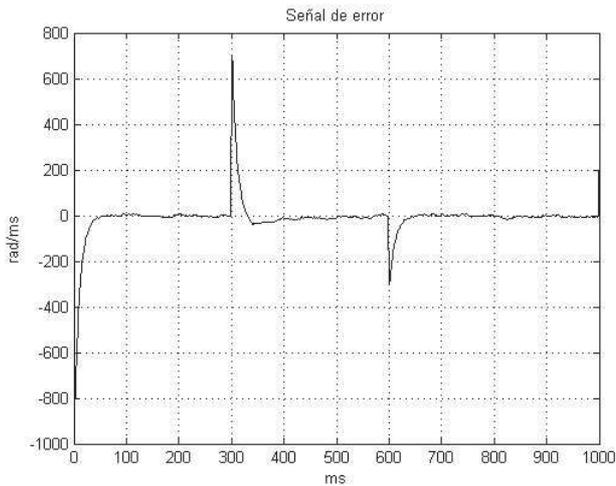


Figura 19: Señal de error utilizando un modelo LIF

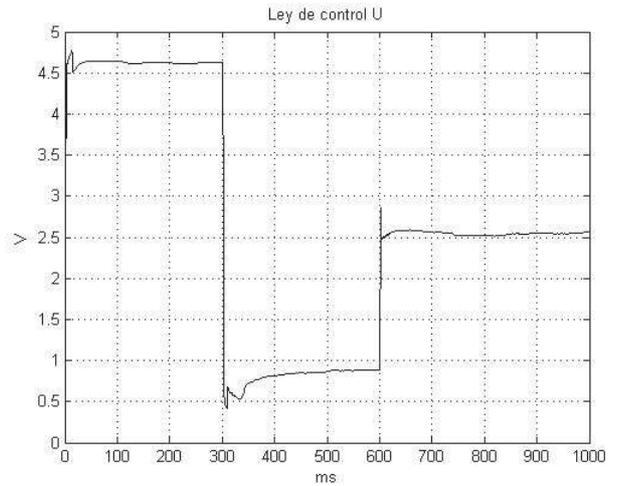


Figura 20: Señal de control utilizando un modelo LIF

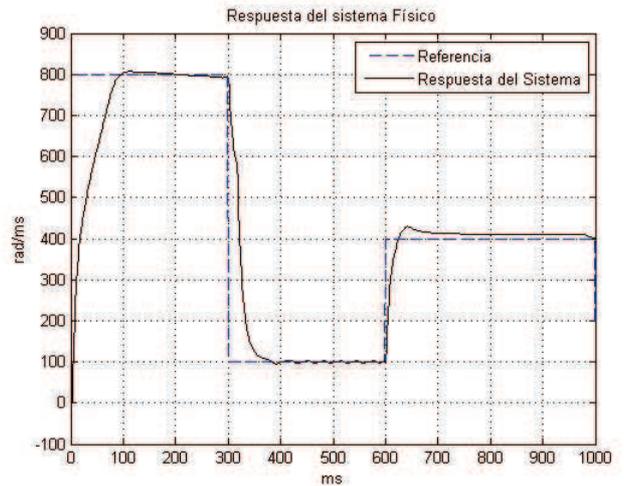


Figura 21: Respuesta del sistema utilizando un modelo Izhikevich

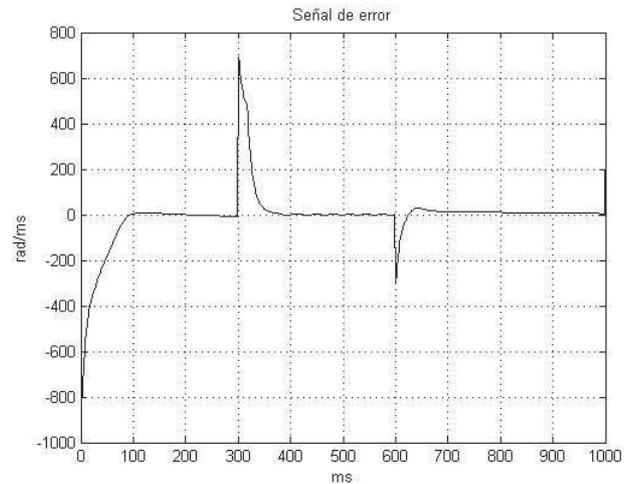


Figura 22: Señal de error utilizando un modelo Izhikevich

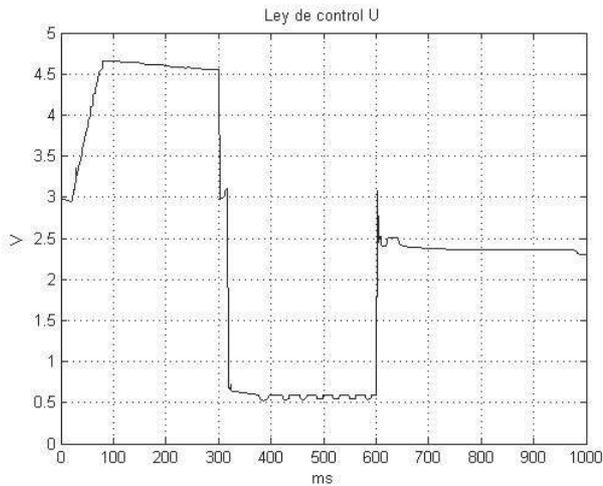


Figura 23: Señal de control utilizando un modelo Izhikevich

6. Control PID clásico

En esta sección se presentan los resultados obtenidos al aplicar un PID clásico sintonizado con base a las reglas de Ziegler-Nichols (Aström and Hägglund, 2006; Ogata, 2003). Para lograr esto, se emplea una entrada de voltaje de 4 Volts a los bornes del motor de CD, obteniendo las ganancias $K_p = 0.005$, $K_i = 0.001$ y $K_d = 0.005$. Los resultados obtenidos en lazo cerrado utilizando este control se muestran en las Figuras 24, 25 y 26. Como se puede observar a partir de los resultados experimentales obtenidos, se tienen sobrepulsos en los instantes en que cambia la señal de referencia; sin embargo, éstos están dentro del rango del 10 % del valor máximo de la señal de referencia.

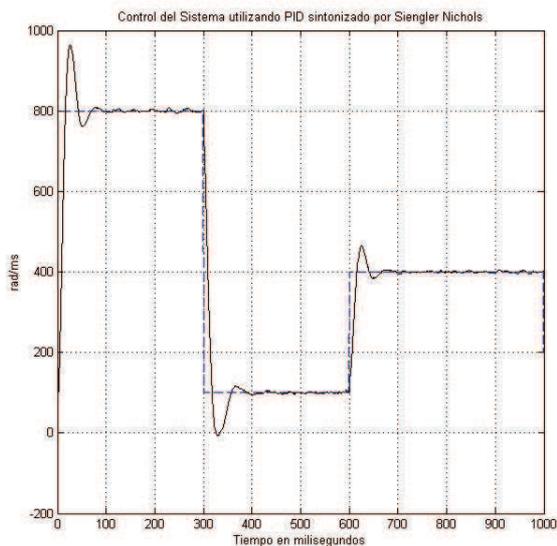


Figura 24: Señal de referencia y respuesta del motor en lazo cerrado

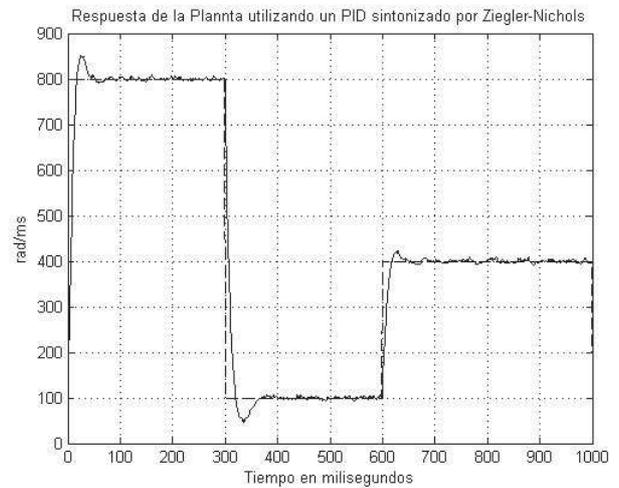


Figura 25: Señal de error

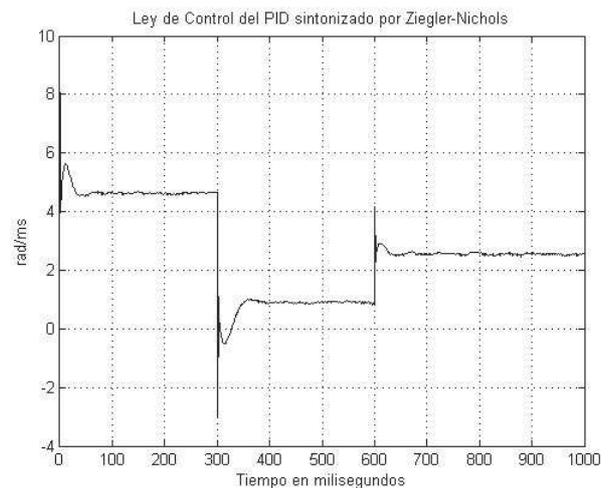


Figura 26: Señal de control utilizando un PID clásico

6.1. Comparación entre el PID clásico y el PID-S propuesto

En este apartado se presentan los resultados experimentales en laboratorio, al aplicar por separado ambos controladores al motor de CD. Como se puede observar en las Figuras 27 y 28 el PID-S presenta una mejora en el seguimiento de la señal de referencia; además, la señal de control del PID-S es más suave que la del PID clásico, esto sucede en los instantes en que cambia la señal de referencia, ver la Figura 28.

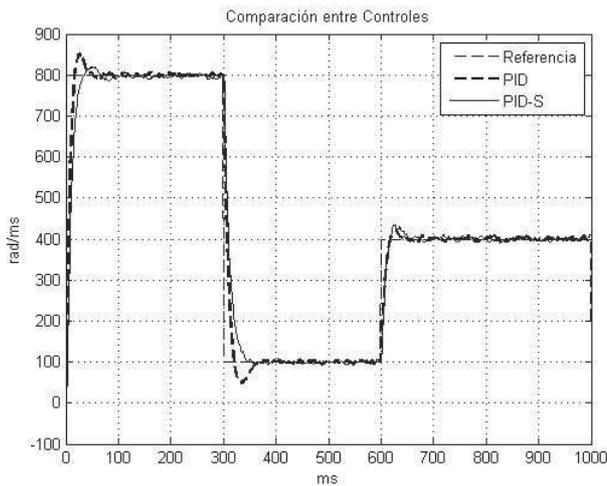


Figura 27: Respuesta del sistema en lazo cerrado ante cambios en la señal de referencia

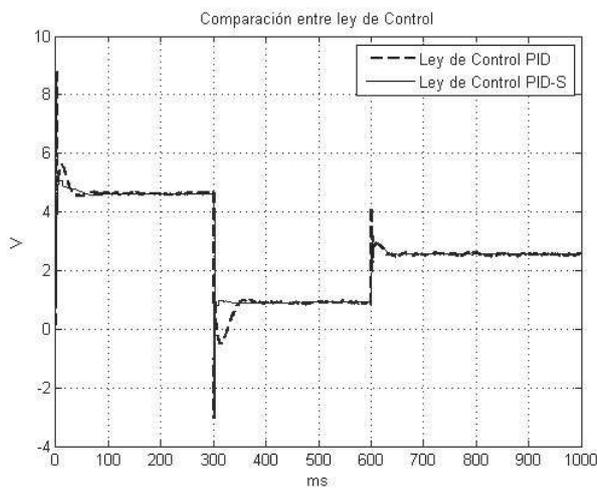


Figura 28: Señal de control utilizando ambos controladores

7. Conclusiones y Trabajo Futuro

A partir de los resultados de simulación y experimentales ya presentados en este artículo, se concluye que el esquema de control propuesto, basado en redes neuronales artificiales de pulso, presenta aspectos sumamente interesantes y relevantes relacionados principalmente con los tiempos de respuesta

deseables y con la arquitectura de red. En esta arquitectura, se emplean tres neuronas *spike*, las cuales son entrenadas considerando el error, la integral y la derivada del error, con la intención de lograr el objetivo de control deseado. Estas tres neuronas *spike* son cambiadas en cada simulación y experimentación para propósitos de comparación. Específicamente, PIF e Izhikevich son los modelos neuronales utilizados, cuyos entrenamientos se hacen con los métodos denominados *SpikePro* y regla delta generalizada para neuronas de pulso. Ambos métodos de entrenamiento emplean el método de codificación descrito por (7) y (8). Todo esto se hace en línea. Los beneficios de un esquema de control como el propuesto en este artículo incluyen el hecho de que permiten que la señal de control se adapte ante cambios en la señal de referencia, sin tener la necesidad de sintonizar nuevamente el controlador. Adicionalmente al hecho de que la señal de control no presenta oscilaciones que puedan afectar la vida útil del motor de CD.

Una de las desventajas que presenta el PID-S propuesto es que no tiene buen desempeño para sistemas lineales inestables por naturaleza. Sin embargo, se puede hacer una modificación al algoritmo propuesto, la cual sí funciona para sistemas inestables; es decir, poner otra neurona a la salida del sumador que entrega la señal de control u mostrado en la Figura 3. Actualmente, se está realizando el análisis de estabilidad del control PID-S en lazo cerrado y se espera reportarlo en otro artículo.

8. Agradecimientos

El primer autor agradece a CONACYT por la beca de maestría otorgada durante el periodo septiembre 2012 - julio 2014, con número de registro 489658.

English Summary

Proportional Integral and Derivative Control Based on Spike Neural Networks.

Abstract

This paper presents a novel control scheme for SISO systems. The work proposes a synergy between the Proportional, Integral and Derivative (PID) control and third generation neural networks. The proposal uses the error, its time derivative and integral to train three spiking neurons, where the sum of the neuron's outputs thereof becomes the control signal. The encoding and decoding between the spatiotemporal neural network information and system information is under a linear interpolation. PID-S denotes the proposed control scheme and its implementation is tested with two different spiking neuron models, which are trained with two different training methods for spiking artificial neural nets, named SpikeProp and generalized Widrow-Hoff's rule for spiking neurons. Numerical simulations and real experiments show good results in all the cases.

Keywords:

PID controller, Spike neural networks, SpikeProp training, Izhikevich model, Haptic interface, PIF model, DC motor.

Referencias

- Abiyev, R. H., Kaynak, O., Oniz, Y., 2012. Spiking neural networks for identification and control of dynamic plants. The 2012 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Kaohsiung, Taiwan, 2012.
- Aström, K., 1997. Computer-Controlled Systems, Theory and Design. Prentice-Hall, New York.
- Aström, K., Hägglund, T., 2006. Advanced PID Control. ISA, The Instrumentation, Systems, and Automation Society.
- Gerstner, W., Kistler, W. M., 2002. Spiking Neuron Models, Single Neurons, Population, Plasticity. MIT Press, Cambridge.
- Haykin, S., 1999. Neural Networks: A Comprehensive Foundation. Prentice-Hall, New York.
- Izhikevich, E. M., 2003. Simple model of spiking neurons. IEEE Transactions on Neural Networks, 1569–1572.
- Izhikevich, E. M., 2004. Which model to use for cortical spiking neurons? IEEE Transactions on Neural Networks, 1063–1070.
- Jalalutram, R., Bhuiyan, M. A., Taha, T. M., 2009. Character recognition with two spiking neural network models on multicore architectures. IEEE Symposium Computational Intelligence for Multimedia Signal and Vision Processing.
- Klafter, R. D., Chmielewski, T. A., Negin, M., 1989. Robotic Engineering An Integrated Approach. Prentice Hall, New Jersey.
- Maass, W., Bishop, C. M., 1999. Pulsed Neural Networks. MIT Press, Cambridge, Mass.
- Mohammed, A., Schliebs, S., Matsuda, S., Kasabov, N., 2011. Method for training a spiking neuron to associate input-output spike trains. 12th INNS EANN-SIG International Conference, EANN 2011 and 7th IFIP WG 12.5 International Conference, AIAI 2011, Corfu, Greece, September 15-18, 2011, Proceedings Part I, 219–228.
- Nichols, E., McDaid, L., Siddique, N., February 2013. Biologically inspired snn for robot control. IEEE Transactions on Cybernetics 43 (1), 115–128.
- Ogata, K., 2003. Ingeniería de Control Moderna. Pearson, Madrid.
- Oniz, Y., Aras, A. C., Kaynak, O., 2013a. Control of antilock braking system using spiking neural networks. 39th Annual Conference of the IEEE Industrial Electronics Society, IECON 2013, 3422–3427.
- Oniz, Y., Kaynak, O., R. Abiyev, 2013b. Spiking neural networks for the control of a servo system. 2013 IEEE International Conference on Mechatronics (ICM), 94–98.
- Ponulak, F., 2005. Resume - new supervised learning method for spiking neural networks. Technical Report, Poznan.
- Ramos-Velasco, L. E., Ramos-Fernández, J. C., Islas-Gómez, O., García-Lamont, J., Espejel-Rivera, M. A., Márquez-Vera, M. A., Julio-Septiembre 2013. Identificación y control wavenet de un motor de CA. Revista Iberoamericana de Automática e Informática Industrial 10 (3), 269–278.
- Rossum, M. V., 2001. A novel spike distance. Neural Computation, 751–763.
- Rullen, R. V., Delorme, A., Gautrais, J., Thorpe, S., 1999. SpikeNET: A simulator for modeling large networks of integrate and fire neurons. Neurocomputing, 989–996.
- Schmitzer, G., Maass, W., Sontag, E. D., 1991. On the computational power of sigmoid versus boolean threshold circuits. 32nd Annual Symposium on Foundations of Computer Science.
- Schwartz, J. H., Kandel, E. R., Jessell, T. M., 2000. Principles of Neural Science. McGraw-Hill, Health Professions Division, New York.