

**IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL DE FLUJO DE AGUA
BASADO EN REDES NEURONALES ARTIFICIALES.**

RUBÉN DARÍO URIBE QUEJADA

**INSTITUCIÓN UNIVERSITARIA DE ENVIGADO
FACULTAD DE INGENIERÍAS
PROGRAMA INGENIERÍA DE ELECTRÓNICA
ENVIGADO
2009**

**IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL DE FLUJO DE AGUA
BASADO EN REDES NEURONALES ARTIFICIALES.**

RUBÉN DARÍO URIBE QUEJADA

Trabajo de grado para optar el título de Ingeniero Electrónico

**Asesor
HUGO DE JESÚS MESA YEPES
Ms. en INGENIERIA DE SISTEMAS**

**INSTITUCIÓN UNIVERSITARIA DE ENVIGADO
FACULTAD DE INGENIERÍAS
PROGRAMA INGENIERÍA DE ELECTRÓNICA
ENVIGADO
2009**

**IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL DE FLUJO DE AGUA
BASADO EN REDES NEURONALES ARTIFICIALES.**

NOTA DE ACEPTACION: _____

Firma del Presidente del jurado

Firma del Jurado

Firma del Jurado

DEDICATORIA

Dedico este trabajo a mi madre Celeny, mi hermana Karen, a la memoria de mi padre Álvaro, porque siempre me han brindado su amor, su apoyo incondicional a lo largo de mi formación como profesional.

AGRADECIMIENTOS

Al Docente Hugo de Jesús Mesa Yepes, por su acompañamiento, apoyo y asesoría profesional.

A mi familia, por estar siempre a mi lado en las buenas y en las malas, motivándome para seguir sin importar los obstáculos que se me presentaran.

Y a todas las demás personas que fueron partícipes directa o indirectamente en este proyecto.

CONTENIDO

	Pág.
INTRODUCCIÓN.	18
1. PRESENTACION DEL TRABAJO DE GRADO.	21
1.1 PLANTEAMIENTO DEL PROBLEMA.	21
1.2 FORMULACION DEL PROBLEMA.	22
1.3 OBJETIVOS.	22
1.3.1 Objetivo General.	22
1.3.2 Objetivos Específicos.	22
1.4 ALCACE.	23
1.5 JUSTIFICACION.	24
1.6 DISEÑO METODOLÓGICO.	25
1.7 CRONOGRAMA DE ACTIVIDADES.	27
1.8 PRESUPUESTO.	28
2. MARCO DE REFERENCIA DE CONTROL BASADO EN REDES NEURONALES ARTIFICIALES.	30
2.1 ANTECEDENTES.	30
2.1.1 Aplicación de las Redes Neuronales para la Identificación de un Sistema no lineal.	30
2.1.2 Implementación de un control digital en un PC para un proceso de flujo.	30
2.1.3 Red neuronal para el control de un servomecanismo.	30
2.1.4 Diseño de un sistema de control con algoritmo predictivo para un proceso de control de flujo de agua.	30
2.2 IDENTIFICACIÓN DE LAS CARACTERÍSTICAS CORESPONDIENTES AL MODELO DEL SISTEMA DE FLUJO DE AGUA EN EL MÓDULO DE FLUJO-NIVEL.	31

2.2.1 DESCRIPCIÓN DEL MÓDULO.	31
2.3 ANÁLISIS DEL PROCESO DE CONTROL.	33
2.3.1 Modelación del sistema.	33
2.3.2 Identificación del sistema.	36
2.3.3 Aproximación a modelos de primer orden con retardo (POR).	37
2.4 APLICACIÓN DE UN CONTROLADOR PID EN EL MÓDULO DE FLUJO.	39
2.4.1 Elementos del control convencional PID.	39
2.4.2 Cálculo del controlador por Ziegler-Nichols.	41
2.4.3 Implementación del controlador por Ziegler-Nichols.	42
2.5 GENERACIÓN DE LA BASE DE DATOS.	44
2.6 ESTRUCTURACIÓN Y ADECUACION DE LOS DATOS.	45
3. EVALUCIÓN DEL COMPORTAMIENTO DE DIFERENTES MODELOS DE RED NEURONAL ARTIFICIAL PARA EL PROCESO DE CONTROL.	47
3.1 INTRODUCCIÓN A LA REDES NEURONALES.	48
3.1.1 Neurona Biológica.	48
3.1.2 Neurona Artificial.	50
3.1.3 Función de activación.	51
3.1.3.1 Limitador fuerte (Hardlim).	51
3.1.3.2 Función de transferencia lineal (purelin).	52
3.1.3.3 Función de transferencia sigmoideal (logsig).	52
3.1.4 Clasificación de las redes neuronales.	53
3.2 ALGUNOS MODELOS DE REDES NERONALES ARTIFICIALES.	54
3.2.1 Perceptrón.	54
3.2.1.1 Regla de aprendizaje.	55
3.2.2 Adaline.	56
3.2.3 Perceptrón multicapa.	57
3.2.3.1 Algoritmo de retropropagación del error o BP.	58
3.2.4 Funciones de base radial (RBF).	59

3.3 ENTRENAMIENTO DE LOS MODELOS DE REDES NEURONALES ARTIFICIALES.	61
3.3.1 Entrenamiento del preceptrón simple.	61
3.3.2 Entrenamiento del preceptrón multicapa (MLP).	63
3.3.3 Entrenamiento del RBF.	74
4. DESARROLLO DEL MODELO DE CONTROL BASADO EN REDES NEURONALES ARTIFICIALES.	83
4.1 Desarrollo del modelo del modelo de RNA's en LabView.	83
5. VALIDACIÓN DEL COMPORTAMIENTO DEL SISTEMAS DE CONTROL EN EL MÓDULO DE FLUJO-NIVEL.	89
6. CONCLUSIONES.	92
8. RECOMENDACIONES.	94
BIBLIGRAFIA.	95
ANEXOS.	97

LISTA DE TABLAS

	Pág.
Tabla 1. Cronograma de actividades.	25
Tabla 2. Presupuesto global del proyecto.	26
Tabla 3. Descripción de los gastos de personal.	26
Tabla 4. Descripción de materiales y suministros.	26
Tabla 5. Descripción de las salidas de campo.	27
Tabla 6. Descripción bibliografía.	27
Tabla 7. Descripción equipos.	27
Tabla 8. Descripción publicaciones.	27
Tabla 9. Parámetros del controlador PID por Ziegler-Nichols.	38
Tabla 10. Funciones de Transferencia.	50
Tabla 11. Valores de la topología MLP 1.	61
Tabla 12. Valores de la topología MLP 2.	63
Tabla 13. Valores de la topología MLP 3.	65
Tabla 14. Pesos sinápticos de las entrada.	66
Tabla 15. Pesos sinápticos de capa oculta.	66
Tabla 16. Bias de capa oculta y capa de salida.	66
Tabla 17. Valores de la topología MLP 4.	68
Tabla 18. Pesos sinápticos de las entrada.	68
Tabla 19. Pesos sinápticos de capa oculta.	69
Tabla 20. Bias de capa oculta y capa de salida.	69
Tabla 21. Valores de la topología RBF 1.	71
Tabla 22. Pesos sinápticos de entrada.	71
Tabla 23. Peso sinápticos de capa oculta.	72
Tabla 24. Bias del capa oculta y capa de salida.	72
Tabla 25. Valores de la topología RBF 2.	73
Tabla 26. Pesos sinápticos de las entrada.	73

Tabla 27. Pesos sinápticos de la capa oculta.	74
Tabla 28. Bias del capa oculta y capa de salida.	74
Tabla 29. Valores de la topología RBF 3.	76
Tabla 30. Pesos sinápticos de las entrada.	76
Tabla 31. Pesos sinápticos de capa oculta.	77
Tabla 32. Bías de capa oculta y capa de salida.	77

LISTA DE FIGURAS

	Pág.
Figura 1. Módulo de flujo y nivel de la IUE.	31
Figura 2 .a) tanque de control, b) tanque de reserva de agua, c) motobomba.	32
Figura 3. a) Transmisor de flujo, b) IP, c) válvula, d) compresor.	33
Figura 4. Respuesta del proceso.	34
Figura 5. Escalón desplazado.	35
Figura 6. Validación del modelo.	35
Figura 7. Representación de un sistema dinámico.	36
Figura 8. Forma de aplicar el escalón para obtener la curva de reacción.	38
Figura 9. Curva de reacción para el modelo POR.	38
Figura 10. Diagrama de bloques del proceso de control PID.	43
Figura 11. Conversor de voltaje a corriente.	43
Figura 12. Conversor de corriente a voltaje.	43
Figura 13. Base de datos del proceso de control.	44
Figura 14. Base de datos escalada.	45
Figura 15. Neuronas Biológicas.	48
Figura 16. Proceso químico de una sinapsis.	49
Figura 17. Similitud entre neurona biológica y artificial.	50
Figura 18. Proceso de una red neuronal artificial.	51
Figura 19. Función de transferencia Hardlim.	52
Figura 20. Función de transferencia lineal.	52
Figura 21. Función de transferencia sigmoideal.	52
Figura 22. Clasificación de las redes neuronales.	54
Figura 23. Estructura del perceptrón simple.	54
Figura 24. Estructura del adaline.	56
Figura 25. Estructura del perceptrón multicapa.	57
Figura 26. Estructura de red RBF.	59

Figura 27. Topología de la red perceptrón.	62
Figura 28. Error del perceptrón.	62
Figura 29. Simulación del perceptrón vs. Salida esperada.	63
Figura 30. Topología MLP 1.	64
Figura 31. Error del MLP 1.	65
Figura 32. Simulación del MLP 1.	66
Figura 33. Topología MLP 2.	67
Figura 34. Error del MLP 2.	67
Figura 35. Simulación del MLP 2.	68
Figura 36. Topología MLP 3.	69
Figura 37. Error del MLP 3.	70
Figura 38. Simulación del MLP 3.	71
Figura 39. Topología MLP 4.	72
Figura 40. Error del MLP 4.	73
Figura 41. Simulación del MLP 4.	74
Figura 42. Topología RBF 1.	75
Figura 43. Error del RBF 1.	76
Figura 44. Simulación del RBF 1.	77
Figura 45. Topología RBF 2.	78
Figura 46. Error del RBF 2.	79
Figura 47. Simulación del RBF 2.	79
Figura 48. Topología RBF 3.	80
Figura 49. Error del RBF 3.	81
Figura 50. Simulación del RBF 3.	82
Figura 51. Panel frontal del VI.	83
Figura 52. Panel de diagrama de bloques del VI.	84
Figura 53. Panel icono/conector del VI.	84
Figura 54. Función de Matlab en LabView.	85
Figura 55. Programa de control con RNA's.	85

Figura 56. Configuración del puerto serial.	86
Figura 57. Diagrama de pines del microcontrolador.	87
Figura 58. Circuito del filtro RC.	87
Figura 59. Panel frontal de control con RNA's.	88
Figura 60. Bloque para almacenamiento de datos.	89
Figura 61. Pruebas al control convencional.	90
Figura 62. Pruebas al control basado en RNA's.	90

LISTA DE ANEXOS.

	Pág.
ANEXO A. Programa de identificación.	97
ANEXO B. Programa de estructuración y adecuación de datos.	105
ANEXO C. Programa de entrenamiento de redes neuronales.	108
ANEXO D. Artículo científico.	110

GLOSARIO.

ALGORITMO DE CONTROL: Es una lista bien definida, ordenada y finita de operaciones que permite hallar la solución a un problema en controladores industriales o de cualquier tipo de control dependiendo de la aplicación.

CONTROLADOR PID: El controlador PID (Proporcional, Integral y Derivativo) es un controlador realimentado cuyo propósito es hacer que el error en estado estacionario, entre la señal de referencia y la señal de salida de la planta, sea cero.

CONVERSIÓN ANÁLOGO/DIGITAL: Consiste en la transcripción de señales analógicas en señales digitales, con el propósito de facilitar su procesamiento.

FLUJO: Caudal de un fluido desplazándose por un ducto.

IDENTIFICACION DE SISTEMAS: conjunto de estudios, algoritmos, información necesaria, cuyo objetivo principal es obtener modelos (ecuaciones) matemáticos, mediante datos obtenidos experimentalmente.

NEURONA: Cuerpos celulares que forman parte constituyente del cerebro. Especializadas para comunicaciones y computación.

SINAPSIS: Se corresponde con el lugar donde es transferida la información de una neurona a otra.

RED NEURONAL (RN): Es un conjunto de neuronas dispuestas en el cerebro (unidades de procesamiento) con memoria local propia, interconectadas mediante canales de comunicación (dendritas y axones) por los cuales se conducen impulsos mediante los neurotransmisores que son activados en la sinapsis que ocurre entre neuronas.

RED NEURONAL ARTIFICIAL (RNA): Es una red de unidades de procesamiento que tienen memoria local propia, interconectadas mediante canales de comunicación por los cuales se conducen datos numéricos. Cada unidad ejerce operaciones a través de las conexiones que son excitatorias o inhibitorias controladas por los pesos sinápticos.

RESUMEN.

En este trabajo presenta el diseño de un controlador para un sistema de flujo de agua basado en redes neuronales artificiales. Para el desarrollo de este trabajo fue necesaria la implementación de un controlador PID, para ello fue necesario obtener el modelo dinámico de la planta mediante la aplicación de una serie de escalones a la válvula de control, de este proceso se adquirieron los datos necesarios para la identificación del sistema que rige el comportamiento del proceso de flujo de agua; luego de realizar esto, se efectúan los cálculos matemáticos que dan como resultado la parámetros de la función de transferencia del controlador.

Con el controlador calculado, se procede a implementarlo en el modulo de flujo de agua, para hacer una nueva adquisición de datos, pero esta vez con el proceso controlado, después, la base de datos generada por el proceso, es sometida a una estructuración y adecuación para el posterior entrenamiento de las redes neuronales artificiales, ya que las redes tiene mejores respuestas en el entrenamiento con valores escalados entre ciertos rangos como -1 y 1; luego, se procese con el entrenamiento de las redes y se valora su desempeño, dependiendo de este desempeño se escoge la mejor red mejor red; esta mejor red es la que será implementada para controlara el sistema de flujo de agua, con la red neuronal ya implementada se validara que tan eficiente es comparándola con el controlador PID anteriormente implementado.

ABSTRACT

This paper presents the design of a controller for a water flow system based on artificial neural networks. For the development of this work was necessary to implement a PID controller, it was necessary to obtain the dynamic model of the plant by applying a series of steps to the control valve of this process of acquiring the necessary data for the identification of the system governing the behavior of water flow process, then to accomplish this, the mathematical calculations are performed that result in the parameters of the transfer function of the controller.

With the controller calculated, we proceed to implement it in the water flow module, to make a new data acquisition, but this time with the controlled process, then the database generated by the process, is subjected to a structured and suitability for further training of artificial neural networks because the networks have better answers in training with scaled values between certain ranges as -1 and 1, then is processed with the training of the networks and their performance is evaluated, depending This performance was better network chooses the best network, this network is the best that will be implemented to control water flow system with the neural network is implemented and how effective is validated by comparing with the PID controller previously implemented.

INTRODUCCIÓN

En la actualidad se busca generar una solución para los problemas de control que son frecuentemente solucionados con métodos tradicionales de control, por esto se optó por buscar una solución fuera de lo convencional como lo son las redes neuronales artificiales que ofrecen una alternativa novedosa que simplifica y permite comparar su eficiencia de este nuevo enfoque respecto a las metodologías tradicionales, a través de conocimientos matemáticos de un nivel aceptable para comprender el fundamento y desarrollo del problema que se quiere resolver. Las redes neuronales artificiales, son una imitación del proceso de una neurona biológica, e imitan las características que hacen tan llamativas y prácticas, como lo son la tolerancia a fallas, procesamiento paralelo, son robustas, flexibles, consumen poca energía, lo que le da una ventaja sobre los computadores convencionales que se basan en la arquitectura de Von Neumann¹.

La capacidad del cerebro humano de pensar, recordar y resolver problemas ha inspirado a muchos científicos intentar o procurar modelar en el ordenador el funcionamiento del cerebro humano.

Los profesionales de diferentes campos como la ingeniería, filosofía, fisiología y psicología han unido sus esfuerzos debido al potencial que ofrece esta tecnología y están encontrando diferentes aplicaciones en sus respectivas profesiones. En este caso las aplicaciones se enfocaría en los problemas que tengan que ver con en campo del control de procesos, como lo son control de nivel, flujo, temperatura y presión que son procesos que se presentan con frecuencia en la industria lo que indica que hay posibilidades casi ilimitadas de aplicación de las RNA para el control.

En el primer capítulo, se trata de todo lo que tiene que ver con al origen del problema que se quiere tratar, el objetivo general de la investigación, objetivos específicos, la justificación del por qué es necesario el desarrollo del proyecto, que alcance tendrá y que metodología se aplicará.

En el segundo capítulo, se hace una documentación necesaria para el desarrollo del inicio del proyecto, como lo son los antecedentes referentes al control de flujo de agua y controles donde utilizan redes neuronales, además, se hace una identificación de las características del módulo de flujo de agua, la dinámica del

¹ la arquitectura de Von Neumann, se refiere a las arquitecturas de computadoras que utilizan el mismo dispositivo de almacenamiento tanto para las instrucciones como para los datos.

proceso y el desarrollo de un controlador convencional PID (Proporcional Integral Derivativo).

En el tercer capítulo, se documenta sobre el modelo neuronal biológico y artificial, las funciones a activación de las RNA's, algunos modelos de RNA's, además, se hace el proceso de entrenamiento de las RNA's, para una posterior implementación, este proceso se lleva a cabo con el Toolbox de Matlab para redes neuronales.

En el cuarto capítulo, se implementa la RNA que mejor comportamiento tuvo en el proceso de entrenamiento, se desarrolla la red en un programa de interfaz grafica como lo es LabView y utilizando en este funciones de Matlab.

En el quinto capítulo, se hacen pruebas al controlador convencional y al controlador basado en RNA's para determinar cuál de los controladores es más eficiencia controlando el sistemas de flujo de agua.

IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL DE FLUJO DE AGUA BASADO EN REDES NEURONALES ARTIFICIALES.

1.1 PLANTEAMIENTO DEL PROBLEMA.

Los sistemas de control se encuentran utilizados en diferentes procesos todos los sectores de la industria, tales como control de calidad de los productos manufacturados, líneas de ensamble automático, control numérico de máquinas-herramientas, tecnología espacial, sistemas de armas, sistemas de transporte, sistemas de potencia, robótica, entre otros.²

Además, en los últimos 20 años la tecnología de los sistemas de control automático convencionales se ha caracterizado por la sustitución de los lazos de control análogo por sistemas de control digital, utilizando el computador en el control de procesos. Esto permite resolver problemas específicos de regulación y de seguimiento de dispositivos con una mejor relación de funciones de supervisión, monitoreo y tratamiento de datos con un reducido costo adicional.³

Los métodos de control convencionales como el PID (Proporcional Integral Derivativo) se enfocan en la simplificación del análisis y diseño de controladores para sistemas lineales, pero dichos sistemas lineales no existen en la práctica, dado que todos los sistemas físicos son no lineales en algún grado, por lo que un sistema de control de este tipo está limitado por intervalos en los cuales los sistemas físicos exhiben una característica lineal. Por estas particularidades del control convencional, se requiere hacer modelación a través de ecuaciones diferenciales, análisis vectorial y matricial, análisis de circuitos y mecánica, lo que aumenta el grado de complejidad del diseño, a pesar de esto, son los sistemas de control más utilizados desde el ámbito académico.⁴

En el programa de ingeniería electrónica de la Institución Universitaria de Envigado, en las asignaturas referentes al control como lo son control análogo, control digital y el grupo de investigación Automatización y Robótica han enfocado sus trabajos e investigación en sistemas de control convencionales PID para el manejo de los módulos didácticos presentes en el laboratorio de Automatización y Robótica de la IUE como lo es el módulo de flujo-nivel de agua, sin tener en

² KUO, Benjamin C. Sistemas de control automático. Séptima edición. México: Prentice Hall Hispanoamericana S.A., 1996. p.2.

³ GARCÍA JAIMES, Luís Eduardo. Control Digital: Teoría y práctica. Segunda edición. Medellín: Politécnico Colombiano Jaime Isaza Cadavid. 2009. p.2, [online].

⁴ KUO. OP.CIT. p. 15.

cuenta la posibilidad profundizar sus trabajos e investigaciones en otros tipos de control, por ello es conveniente obtener un mejor aprovechamiento de las herramientas que se encuentran disponibles en los laboratorios de la IUE . Por esto se ve como una necesidad el ofrecimiento del conocimiento y las herramientas necesarias de manera que los estudiantes interesados puedan profundizar en temáticas acerca el manejo, desarrollo e implementación de algoritmos de control basados en técnicas no convencionales, que pueden aplicar en dichos módulos como el de flujo-nivel de agua, además, siendo egresados de la IUE con los conocimientos obtenidos en este tipo de control avanzado tendrían las bases para escalar hacia los procesos industriales.

1.2 FORMULACIÓN DEL PROBLEMA.

¿Como profundizar en el manejo un sistema de control de flujo de agua basado en redes neuronales artificiales en el modulo de nivel-flujo del laboratorio de la IUE.?

1.3 OBJETIVOS DE LA INVESTIGACIÓN.

1.3.1 Objetivo general. Implementar un sistema de control de flujo de agua basado en redes neuronales artificiales en el modulo de flujo-nivel del laboratorio de la IUE.

1.3.2 Objetivos específicos.

- Identificar las características correspondientes al modelo del sistema de flujo de agua en el módulo de flujo-nivel.
- Evaluar el comportamiento de diferentes modelos de red neuronal artificial para el proceso de control.
- Desarrollar el modelo de control basado en redes neuronales artificiales.
- Validar el comportamiento del sistema de control desarrollado en el modulo de flujo-nivel.

1.4 ALCANCE.

El alcance del proyecto fue implementar un sistema de control de flujo de agua basado en redes neuronales artificiales con una interfaz grafica desarrollada en una computadora que se comunica al proceso (al módulo) a través de un micro controlador que interpreta la información que proviene del computador y actura sobre el módulo, además, el módulo en donde se va a desarrollar el proyecto hace parte de la dotación del laboratorio de Automatización y Robótica de la IUE, este es un módulo didáctico, en el modulo se pueden llevar a cabo el control de dos variables que son el nivel y el flujo de liquido que en este caso se escogió agua por su bajo costo, fácil de conseguir y fácil manipulación, pero puede ser otro liquido con características físicas similares. Se espera que el proyecto les sirva de guía para que aquellos estudiantes que se interesen en áreas como el control y sistemas inteligentes como lo son la redes neuronales artificiales.

Por otro lado, con el desarrollo de este proyecto logra fortalecer la competitiva de la IUE, muestra adelantos tecnológicos e innovaciones, que le permitan ganarse cierto prestigio frente a las otras universidades que ofrezcan programas afines a la Ingeniería electrónica.

También se espera despertar el interés de la comunidad universitaria por el área de la Automatización y la robótica, siendo este un tema del cual se profundiza muy poco y tiene una gran proyección en todos los procesos de producción.

Con este proyecto también se pretendió fortalecer, aumentar e integrar los conocimientos adquiridos en el ciclo de formación y aprendizaje ofrecido por la Institución tanto teórica como práctica.

Se espera además que los estudiantes de la facultad se interesen en temas de investigación de esta clase, como son los controladores basados en sistemas inteligentes, adentrándose tema de poca profundización en la IUE y que tenga gran proyección para aplicarse en los procesos de producción.

Los resultados del proyecto se plasmarán en una serie de artículos, los cuales se someterán a evaluación y posible presentación en diferentes eventos nacionales que se realizará en el área de Automatización y control e inteligencia computacional. Los eventos en los cuales se pretende participar son:

Nacionales:

- Semana académica de ingeniería de control de UNAL.

Revistas:

- Revista de la facultad de ingenierías de IUE.

Para cada uno de estos eventos y estas revistas, se escribirá un artículo, con los resultados más relevantes del proyecto. Esto permitirá realizar una retroalimentación de los conocimientos adquiridos y el método propuesto.

1.5 JUSTIFICACIÓN.

El desarrollo del proyecto se justifica desde tres pilares básicos, los cuales son el aspecto académico, el económico y el industrial.

Desde el aspecto académico, el diseño y desarrollo de sistemas de control no convencionales como las redes neuronales artificiales le permitirán al estudiante la ampliación de sus conocimientos tanto teórica como práctica, y dándole así los mecanismos para proponer nuevas opciones de investigación, fortalecimiento de esta manera de las líneas investigación de inteligencia computacional y Automatización y control, lo cual a su vez se traduce en el fortalecimiento del Grupo de investigación en Sistemas Inteligentes de la IUE.

En el aspecto económico, porque La competencia en muchos sectores industriales así como el creciente interés social por los problemas medioambientales relacionados con los procesos de producción provoca la necesidad de disponer de técnicas fiables que permitan la operación del proceso con gran eficiencia y alto grado de flexibilidad. Actualmente los sistemas de control en la industria de procesos deben satisfacer criterios económicos, asociados con el mantenimiento de las variables de proceso en sus referencias minimizando dinámicamente una función de coste de operación, criterios de seguridad y medioambientales, y de calidad en la producción, la cual debe satisfacer ciertas especificaciones sujetas a una demanda normalmente variable.⁵

En la industria, ya teniendo los conocimientos en el diseño y desarrollo de un sistemas de control basado en redes neuronales artificiales para un proceso de flujo de agua del modulo de flujo-nivel de la IUE, el estudiante tendrá las capacidades para emprender un proyecto de mayor magnitud, ya no aplicado a algo didáctico sino algo real, y no solo para un proceso de flujo sino también a procesos para controlar temperatura, presión y entre otros.

⁵ ALBA BORDONS, Carlos. Control Predictivo: metodología, tecnología y nuevas perspectivas. España: Universidad de Sevilla. I Curso de Especialización en Automática, 2000, [online].

1.6 DISEÑO METODOLÓGICO.

Metodología:

El desarrollo del proyecto se realizará mediante la aplicación de los enfoques metodológicos de proyectos de desarrollo tecnológico, ya que se hace uso de conocimientos propios de áreas afines y de otras ciencias, para el diseño y desarrollo de sistemas de control de procesos, en este caso, control basado en redes neuronales artificiales. Para el logro de los objetivos, se recurre específicamente a conocimientos de las áreas de adquisición, acondicionamiento de señales, procesamiento, desarrollo de sistemas basados en redes neuronales y validación del sistema. Conocimientos propios de las redes neuronales artificiales, los cuales se deben combinar con la ciencia del control de procesos, para lograr la construcción de un sistemas óptimo para el control basado en redes neuronales para un proceso de flujo de agua.

El desarrollo técnico se realizará mediante la aplicación de las metodologías propias del desarrollo de sistemas inteligentes basados en redes neuronales artificiales, los cuales comprenden la etapa de diseño, adquisición de datos históricos, entrenamiento y validación del sistema a través del Toolbox de redes neuronales artificiales de MATLAB.

El proyecto se desarrollará en una serie de etapas, las cuales a su vez se estructurarán mediante un conjunto de tareas, que se integrarán para el logro de los objetivos. Las etapas a desarrollar y las tareas son:

Etapas 1. Identificación de los parámetros en el modelo del sistemas de flujo de agua en el modulo de flujo-nivel.

En esta etapa se analizará el proceso de control de flujo, en el modulo flujo-nivel, identificando cuales variables que intervienen en el proceso, el comportamiento físico de dichas variables identificadas previamente, las características del modelo matemático que representa la dinámica del sistema para un control convencional, además, se desarrolla una tarjeta de adquisición con la cual se generara la base de datos con el proceso de control convencional, para ser usado en el entrenamiento de la RNA (Redes Neuronales Artificiales).

Etapas 2. Evaluación del comportamiento de diferentes modelos de red neuronal artificial para el proceso de control.

En esta etapa se utilizarán los datos obtenidos y procesados en la etapa anterior para el entrenamiento de las redes neuronales, además del entrenamiento se

observara el comportamiento de dichas redes, para determinar cuál de ellas es la mejor para el control de flujo de agua.

Etapa 3. Desarrollo del modelo de control basado en redes neuronales artificiales. Después de haber en trenado la RNA y determinar cuál es la mejor para el sistema de control de flujo de agua en la etapa anterior, se implementa el control basado en RNA.

Etapa 4. Validación del comportamiento del sistema de control desarrollado en el modulo de flujo-nivel.

En esta etapa se realizaran pruebas al sistemas de control basado en RNA para el modulo de flujo de agua y se analizará su comportamiento de dicho control, el resultado de las pruebas se comparara con el control convencional que se desarrolló en etapa 1.

Etapa 5: Documentación final y divulgación.

En esta etapa se elaborará el informe final del proyecto, además se prepararán y presentarán varios artículos científicos a ser publicados en diferentes eventos y revistas. Adicionalmente, se recolectarán todos los datos, planos, diseño, procesos de normalización y estandarización, con miras a realizar este control en otro tipo de procesos.

1.7 CRONOGRAMA DE ACTIVIDADES.

Tabla 1. Cronograma de actividades.

Nombre de tarea	Duración	Comienzo	Fin	mayo 2009							junio 2009							julio 2009							agosto 2009							septiembre 2009																				
				10	13	16	19	22	25	28	31	04	07	10	13	16	19	22	25	28	31	03	06	09	12	15	18	21	24	27	30	03	06	09	12	15	18	21	24	27	30	02	05	08	11	14	17	20	23	26	29	01
- Etapa 1. Identificación de los parámetros	30 días	lun 13/04/09	vie 22/05/09																																																	
Identificación de variables.	10 días	lun 13/04/09	vie 24/04/09																																																	
Análisis del proceso de control.	10 días	lun 27/04/09	vie 08/05/09																																																	
Generación de una base	5 días	lun 11/05/09	vie 15/05/09																																																	
Estructuración, adecuación y validación	5 días	lun 18/05/09	vie 22/05/09																																																	
- Etapa 2. Evaluación del comportamiento de RNA	26 días	lun 25/05/09	lun 29/06/09																																																	
Definir las topologías de las RNA	6 días	lun 25/05/09	lun 01/06/09																																																	
Entrenamiento de las RNA.	14 días	mar 02/06/09	vie 19/06/09																																																	
Determinar cual es la mejor RNA.	6 días	lun 22/06/09	lun 29/06/09																																																	
- Etapa 3. Desarrollo del control basado en RNA	30 días	mar 30/06/09	lun 10/08/09																																																	
Desarrollo del modelo de RNA.	10 días	mar 30/06/09	lun 13/07/09																																																	
Implementación de sistemas en el módulo.	10 días	mar 14/07/09	lun 27/07/09																																																	
Adquisición y procesamiento de los datos.	10 días	mar 28/07/09	lun 10/08/09																																																	
- Etapa 4. Validación del comportamiento del sistema	28 días	mar 11/08/09	vie 11/09/09																																																	
Pruebas del sistema de control basado en RNA.	8 días	mar 11/08/09	mié 19/08/09																																																	
Análisis del comportamiento del control desarrollado	8 días	jue 20/08/09	vie 28/08/09																																																	
Comparación del control convencional vs. Control RNA	6 días	sáb 29/08/09	vie 04/09/09																																																	
Interpretación y validación de los datos.	6 días	sáb 05/09/09	vie 11/09/09																																																	
- Etapa 5: Documentación final y divulgación.	114 días	lun 13/04/09	vie 11/09/09																																																	
Preparación de informes	114 días	lun 13/04/09	vie 11/09/09																																																	
Presentación de informes y entrega de resultados.	7 días	vie 04/09/09	vie 11/09/09																																																	
Divulgación de resultados.	7 días	vie 04/09/09	vie 11/09/09																																																	

1.8 PRESUPUESTO

Tabla 2. Presupuesto global del proyecto.

PRESUPUESTO GLOBAL DEL TRABAJO DE GRADO				
RUBROS	FUENTES			TOTAL
	Estudiantes	IUE	Externa	
Personal	\$400.000	\$750.000		\$1.150.000
Material y suministros	\$392.000	\$1.500.000		\$1.892.000
Salidas de campo	\$ 80.000	\$0.00		\$ 80.000
Bibliografía	\$0.00	\$124.720		\$124.720
Equipos	\$0.00	\$13.920.000		\$13.920.000
Publicación	\$ 250.000	\$0.00		\$ 250.000
TOTAL	\$1.122.000	\$16.294.720		\$17.416.720

Tabla 3. Descripción de los gastos de personal.

DESCRIPCIÓN DE LOS GASTOS DE PERSONAL						
NOMBRE DEL INVESTIGADOR	FUNCIÓN EN EL PROYECTO	DEDICACIÓN horas/semana	FUENTES			TOTAL
			Estudiante	IUE	Ext.	
Hugo Mesa Yepes	Asesor	2		\$750.000		\$ 750.000
Rubén Darío Uribe	Estudiante	10	\$ 400.000			\$ 400.000
TOTAL			\$400.000	\$750.000		\$1.150.000

Tabla 4. Descripción de materiales y suministros.

DESCRIPCIÓN DEL MATERIALES Y SUMINISTROS				
Material	FUENTES			TOTAL
	Estudiante	IUE	Externa	
PC		\$1.500.000		\$1500.000
Resma de papel	\$17.000			\$17.000
Impresiones	\$300.000			\$300.000
CD	\$5.000			\$5.000
Circuito impreso	\$30.000			\$30.000
Suministros electrónicos	\$40.000			\$40.000
TOTAL	\$392000	\$1500.000		\$1.892.000

Tabla 5. Descripción de las salidas de campo.

DESCRIPCIÓN DE LAS SALIDAS DE CAMPO				
DESCRIPCIÓN DE LAS SALIDAS DE CAMPO	FUENTES			TOTAL
	Estudiantes	IUE	Externa	
Bibliotecas	\$ 80.000			\$ 80.000
TOTAL	\$ 80.000			\$ 80.000

Tabla 6. Descripción bibliografía.

DESCRIPCIÓN BIBLIOGRAFIA				
DESCRIPCIÓN BIBLIOGRAFIA	FUENTES			TOTAL
	Estudiantes	IUE	Externa	
Redes Neuronales artificiales: Un enfoque práctico.		\$41.520		\$41.520
Redes Neuronales y Sistemas Difusos.		\$43.200		\$43.200
Inteligencia Artificial y Minirobots		\$40.000		\$40.000
TOTAL		\$124.720		\$124.720

Tabla 7. Descripción equipos.

DESCRIPCIÓN EQUIPOS				
DESCRIPCIÓN EQUIPOS	FUENTES			TOTAL
	Estudiantes	IUE	Externa	
Modulo de flujo y nivel		\$12.000.000		\$12.000.000
Múltmetro		\$95.000		\$95.000
Accesorios(cables, entre otros)		\$25.000		\$25.000
Software de controladores		\$1.800.000		\$1.800.000
TOTAL		\$13.920.000		\$13.920.000

Tabla 8. Descripción publicaciones.

DESCRIPCIÓN PUBLICACIONES				
DESCRIPCIÓN PUBLICACIONES	FUENTES			TOTAL
	Estudiantes	IUE	Externa	
Publicación en revista	\$ 250.000			\$ 250.000
TOTAL	\$ 250.000			\$ 250.000

2. MARCO DE REFERENCIA DE CONTROL BASADO EN REDES NEURONALES ARTIFICIALES.

2.1 ANTECEDENTES.

2.1.1 Aplicación de las Redes Neuronales para la Identificación de un Sistema no lineal. Autores: Jesús Acosta, Josep Fernandez, Lenin Becerra. En el presente trabajo se obtuvieron y analizaron modelos de un sistema de reacción continuo (hidrólisis del anhídrido acético) utilizando las Redes Neuronales (RN) con el fin de determinar si los mismos realmente presentan un comportamiento dinámico que sea representativo del proceso real, sirviéndonos de guía para el proyecto ya que aplican los conceptos de redes neuronales aplicadas en un procesos de control.⁶

2.1.2 Implementación de un control digital en un PC para un proceso de flujo. Autores: Rafael Esteban Vásquez M, Juan Miguel Vásquez C. El Grupo de Investigación en Automática y Diseño A+D de la Universidad Pontificia Bolivariana -Medellín, ha implementado un proceso industrial a escala de laboratorio que sirve de plataforma de estudio y desarrollo de sistemas de control no lineal. Se considera como referencia para el proyecto, porque en éste la variable utilizada en el proceso es el flujo de agua al igual que en este proyecto, dándonos conceptos de cómo realizar algunas tareas del trabajo.⁷

2.1.3 Red neuronal para el control de un servomecanismo. Autores: Marcelo Chávez Durán, Reynaldo Hurtado Morató, Josefina Juaniquina Flores. Se considera la utilización de una red neuronal como elemento de control, para ello se modela un servomecanismo que posiciona un elemento óptico a una determinada altitud angular. El servomecanismo utiliza originalmente un dispositivo PID como elemento de control, la red neuronal es entrenada siguiendo el comportamiento del dispositivo PID para finalmente reemplazarlo. Los resultados obtenidos con la red neuronal como elemento de control son comparados con los obtenidos con el dispositivo PID.⁸

2.1.4 Diseño de un sistema de control con algoritmo predictivo para un proceso de control de flujo de agua. Autora: Angélica María Ochoa Ramírez. En este trabajo se realizó el diseño de un sistema de control con algoritmo predictivo para un proceso de control de flujo de agua trabajado en el módulo de flujo-nivel de IUE. Este trabajo sirve de gran ayuda, ya que se trabajó en el mismo módulo

⁶ ACOSTA, Jesús; FERNANDEZ, Joseph; BECERRA, Lenin. Aplicación de las Redes Neuronales para la Identificación de un Sistema no lineal. España: Comité Español de Automática, [online],

⁷ VÁSQUEZ M, Rafael Esteban; VÁSQUEZ C, Juan Miguel. Implementación de un control digital en un PC para un proceso de flujo. Medellín: Universidad Pontificia Bolivariana.

⁸ CHAVEZ DURÁN, Marcelo; HURTADO MORATÓ, Reynaldo; JUANIQUINA FLORES, Josefina. Red neuronal para el control de un servomecanismo. Bolivia: Universidad Mayor De San Andrés, [online]

en donde se desarrollará en proyecto, dando a conocer las características de dicho modulo.⁹

2.2 IDENTIFICACIÓN DE LAS CARACTERÍSTICAS CORESPONDIENTES AL MODELO DEL SISTEMA DE FLUJO DE AGUA EN EL MÓDULO DE FLUJO-NIVEL.

2.2.1 Descripción del módulo. El módulo didáctico de flujo-nivel de agua que se encuentra presente en el laboratorio de Automatización y Robótica fue elaborado en la Institución Universitaria de Envigado, el diseño y la elaboración fue supervisada por el docente en las áreas de control digital y seminario III (control avanzado) Luis Eduardo García Jaimes Ms. en automatización y control. El Módulo para monitoreo y control de flujo y nivel que se muestra en la figura 2, puede utilizarse para realizar prácticas de instrumentación industrial e implementar una gran variedad de estrategias de control análogo y control digital de procesos.

Figura 1. Módulo de flujo y nivel de la IUE.



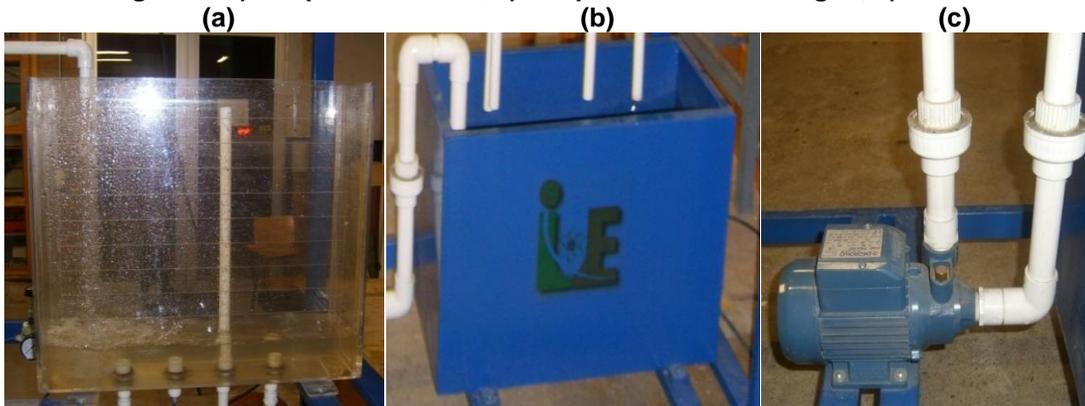
⁹ OCHOA RAMÍREZ, Angélica María. Diseño de un sistema de control con algoritmo predictivo para un proceso de control de flujo de agua. Trabajo de grado Ingeniero electrónico. Envigado

Entre la gran variedad de características que tiene el módulo se encuentran las siguientes:

- Medición de nivel.
- Control on-off de nivel.
- Control análogo de nivel utilizando control P, PI y PID.
- Control digital de nivel utilizando control P, PI, PID y otros algoritmos.
- Medición de flujo.
- Control on-off de flujo.
- Control análogo de flujo utilizando control P, PI y PID.
- Control digital de flujo utilizando control P, PI, PID y otros algoritmos.
- Identificación de sistemas.
- Aplicación de técnicas de control predictivo basado en modelos.
- Aplicación de técnicas de control adaptativo.
- Implementación de sistemas de control borroso, neuronal y neuro-borroso.

El módulo para monitoreo y control de flujo y nivel es una réplica de un proceso de producción real en una escala menor que consta de dos tanques: uno en la parte inferior que se utiliza como tanque de reserva de agua y otro tanque en la parte superior, sobre el cual se realiza control de nivel.

Figura 2 .a) tanque de control, b) tanque de reserva de agua, c) motobomba.

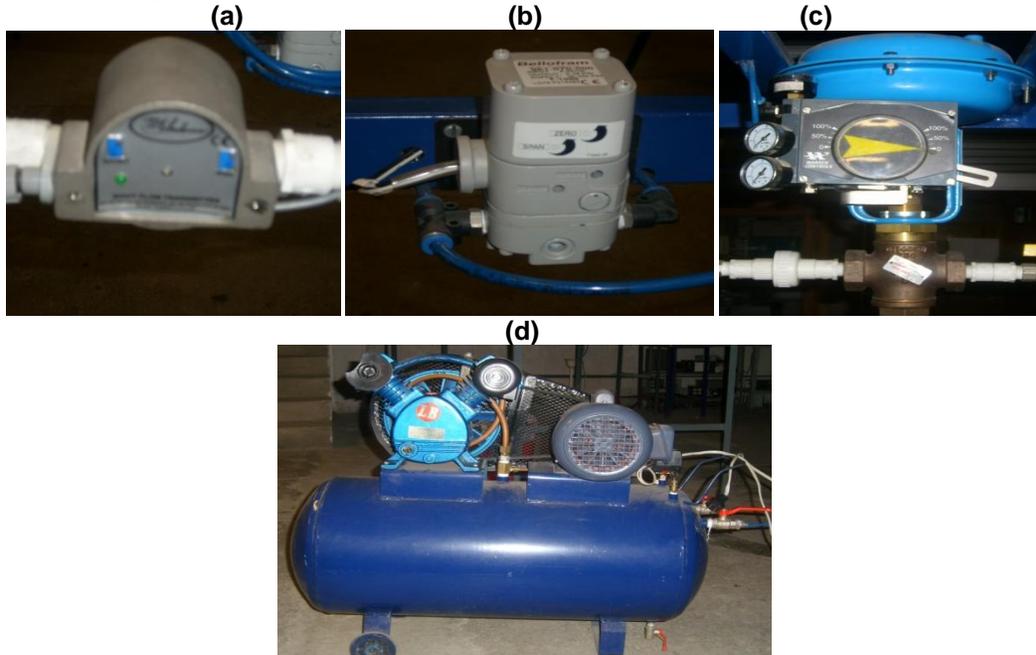


El tanque de control (tanque superior) tiene tres ductos de drenaje de agua hacia el tanque inferior, el primero tiene una válvula de solenoide, la cual permite hacer control on-off de nivel, el segundo ducto es utilizado para evitar desbordamiento de agua en caso de una falla en el sistema de control y el tercer ducto tiene una válvula manual para introducir perturbaciones al lazo de control de nivel. Con ayuda de una motobomba se transporta el agua del tanque inferior (suministro de agua) al tanque superior.

En el control de flujo intervienen directamente los siguientes dispositivos: el transmisor de flujo que entrega una salida en corriente de 4 a 20mA (miliamperios), IP (convertidor de corriente a presión) que sirve para convertir la

señal de control que proviene del un controlador electrónico PID, de una tarjeta de adquisición para hacer control por computador o de un PLC, también está la válvula neumática que es el elemento final de control, esta válvula solo recibe señales de presiones de 3 a 15 PSI por lo que tiene que estar conectada el IP (convertidor de corriente a presión) y para el suministro del aires a presión para que funciones el módulo esta el compresor de aire.

Figura 3. a) Transmisor de flujo, b) IP, c) válvula, d) compresor.



2.3 ANÁLISIS DEL PROCESO DE CONTROL

Para este análisis se utilizará como herramienta primordial los conocimientos obtenidos en el área de control digital en la que se llevo un proceso similar. En este proceso se realizó un control manual, donde la válvula neumática se comunica desde un computador que cuente con el programa LabView y con el software desarrollado en el mismo, para ir aplicando señales de control de tipo escalón para ir abriendo la válvula paso por paso, con el fin de lograr cambios en la dinámica del flujo de agua que se estará midiendo a través del transmisor de flujo que a su vez ira comunicado con la computadora para su almacenamiento y posterior procesamiento para la modelación del sistema.

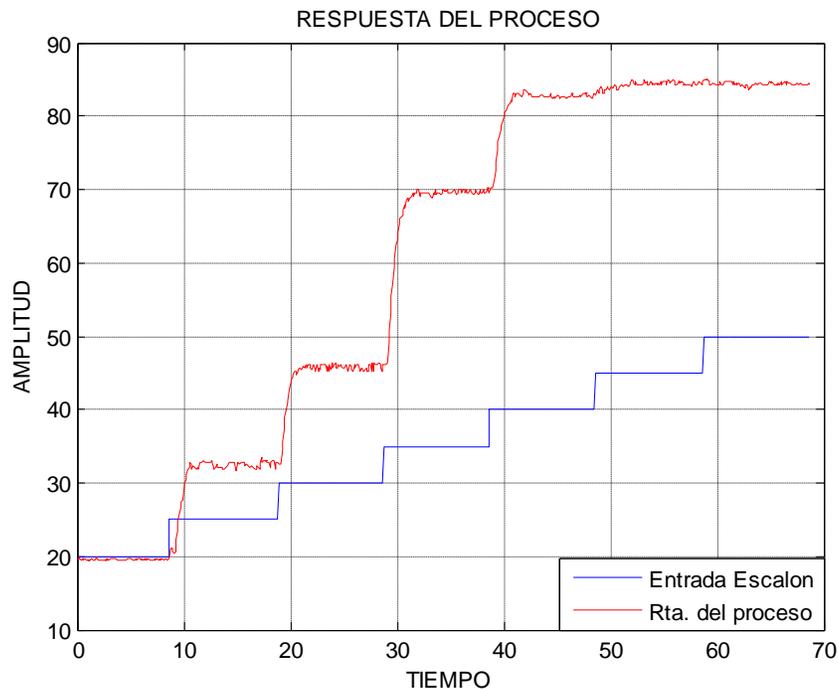
2.3.1 Modelación del sistema. Con los datos obtenidos del módulo de flujo con el sensor de flujo y que se almacenaron en un archivo plano con extensión .dat, con este archivo que se procesará en software desarrollado en Matlab para calcular los parámetros del proceso, la dinámica del sistema y con base a esta se obtendrá la ecuación que representa el modelo del módulo de flujo (ecuación de la planta)

para el desarrollo del algoritmo de control convencional que es el punto de referencia para la implementación del control basado en redes neuronales artificiales.

Con el software en Matlab se muestra paso por paso el desarrollo de los cálculos, de manera que se entienda el procedimiento y los resultados obtenidos. A continuación se mostrará el proceso llevado a cabo en el software para la identificación del modelo matemático del sistema.

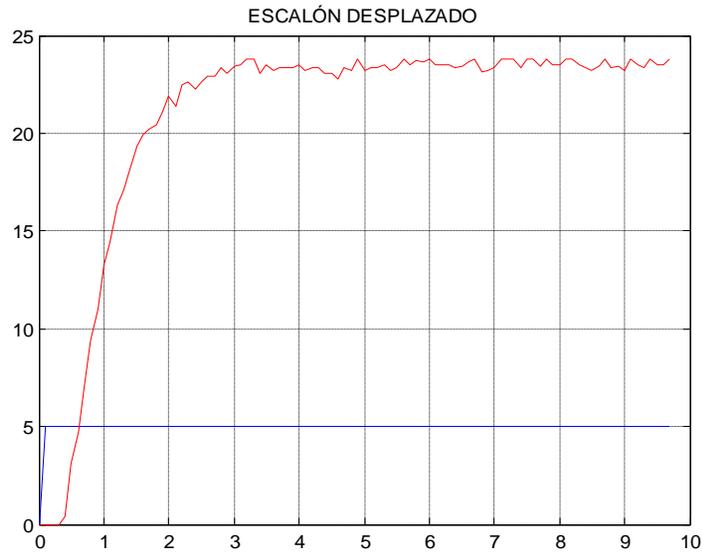
Al iniciar el procesamiento se muestra graficados los datos que se almacenaron anteriormente en la operación manual de la válvula con la señal de entrada tipo escalón y la respuesta del sistema que en este caso sería la variación del flujo con dicha entrada, como lo muestra la figura 3.

Figura 4. Respuesta del proceso.



Como se ve en la figura 4, hay varios escalones, pero para este caso se toma el escalón con mejor comportamiento frente a la entrada aplicada para obtener un mejor resultado, este escalón es el que se encuentra entre los tiempos de 20 (tiempo inicial) y 40 (tiempo final), después de que se escoge en el software el escalón deseado este se amplía y se desplaza de manera que el tiempo inicial que era de 20 se mueva hasta el origen (hasta cero) como lo muestra la figura 5.

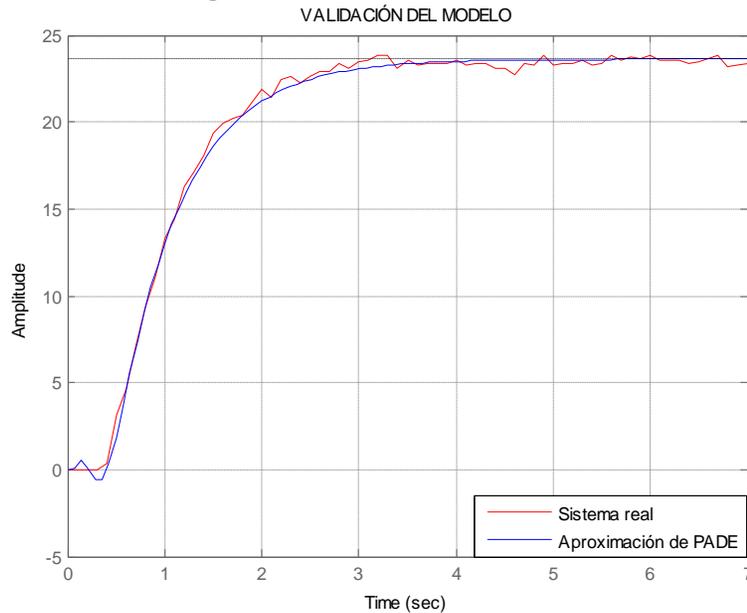
Figura 5. Escalón desplazado.



Esta ampliación y este desplazamiento se hace para hallar los tiempos correspondientes al 28.3% y 63.2% para una aproximación a un sistema de primer orden con retardo (POR), que se explicará con detalle más adelante.

Los resultados obtenidos en Matlab para el 28.3%=0.7294 y para 63.2%=1.4081, además, en el software también se obtiene el modelo del sistema aproximado de primer orden con retardo que se observa en la figura 6 (validación del modelo) y su representación matemática en la ecuación.

Figura 6. Validación del modelo.



$$G(s) = \frac{4.7227t^{-0.459841S}}{0.67585S+1} \quad \text{Ec. 1}$$

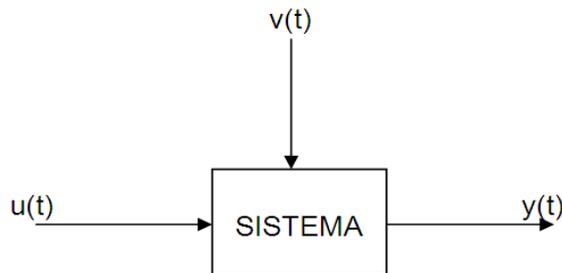
Por otro lado, se obtiene un rango del periodo de muestreo $0.061975 < T_s < 0.092963$, para el cual se escogió 0.19 con lo que se tiene una función de transferencia de pulso de la siguiente manera:

$$HG(Z) = \frac{1.6441Z + 1.5195}{Z^2 - 0.33012Z} \quad \text{Ec. 2}$$

2.3.2 Identificación del sistema. La identificación de un sistema tiene por objetivo la obtención de su modelo matemático a partir de datos experimentales. Se dan dos formas de aproximación a los sistemas: uno se conoce como POR que es la aproximación a modelos de primer orden con retardo y el otro es SOR, aproximación a modelos de segundo orden.

La representación conceptual de un sistema dinámico. El sistema es comandado por variables de entrada $u(t)$ y por perturbaciones $v(t)$. El usuario puede controlar las variables de entrada $u(t)$, pero no las perturbaciones $v(t)$. Las señales de salida $y(t)$ son variables que suministran información útil acerca del sistema, esta representación se puede observar más detallada en la siguiente figura.

Figura 7. Representación de un sistema dinámico.



El procedimiento para la identificación del modelo se fundamenta en tres pasos primordiales que son: recolección de los datos, selección del modelo, validación del modelo.

- **Recolección de los datos:** Los datos son obtenidos por la experimentación realiza en el proceso, lo que se busca es obtener dichos datos lo más confiable posible y que brinden la mejor información del sistema.

- **Selección del modelo:** Es una de las etapas más complejas, porque es necesario tener conocimiento del proceso en cuestión, para poder obtener de manera adecuada el modelo que va a regir el comportamiento del proceso.
- **Validación del modelo:** Como lo indica su nombre, en esta etapa se verifica que los datos recolectados sean lo más aproximados a las características del modelo seleccionado por el investigador.

2.3.3 Aproximación a modelos de primer orden con retardo (POR). Este sistema está representado por una función de transferencia de la siguiente manera.

$$G(s) = \frac{y(s)}{u(s)} = \frac{Kt^{\theta's}}{\tau s + 1} \quad \text{Ec. 3}$$

K : Ganancia de la planta.

θ' : Retardo del sistema.

τ = Constante de tiempo.

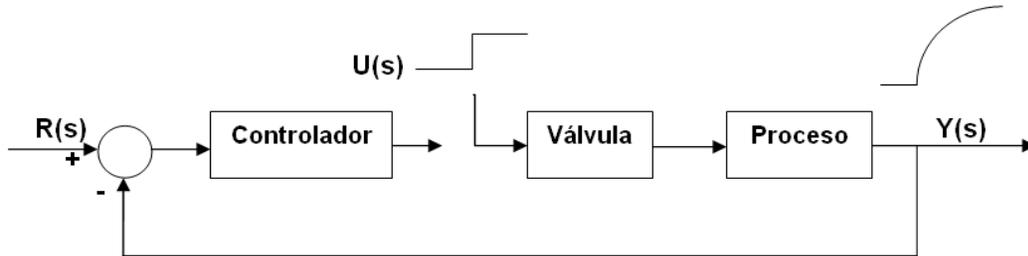
$y(t)$ = Es la salida del sistema.

$u(t)$ = Es la entrada del sistema.

El proceso experimental para obtener la aproximación del modelo consiste en abrir el lazo del control, lo que quiere decir que el controlador queda en modo manual antes del elemento final de control (válvula), con este control manual se crea pequeños y rápidos cambios en forma de escalón en el proceso. La respuesta obtenida se grafica y sobre esta grafica se hace el análisis para estimar los valores de la ganancia (K), la constante de tiempo (τ) y el retardo (θ'). Para logra esto se procede en un principio a determinar el punto de operación de la planta, luego se aplica al sistema en lazo abierto un cambio en escalón de la escala apropiada. Esta operación se debe llevar a cabo varias veces cubriendo así toda la zona lineal del proceso y luego se hace un promedio de valores obtenidos para tener una información más confiable.¹⁰

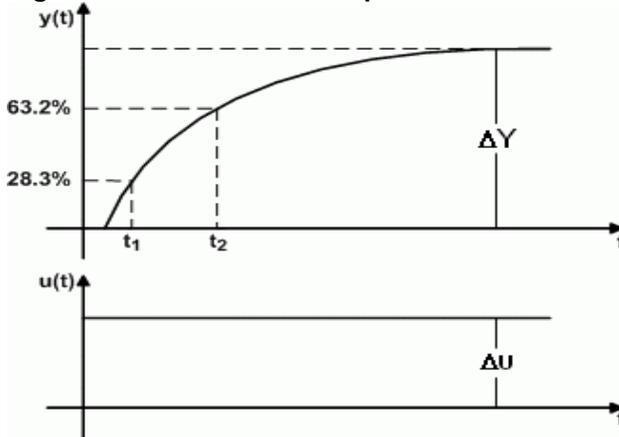
¹⁰ GARCÍA. OP.CIT. p.155-159, [online]

Figura 8. Forma de aplicar el escalón para obtener la curva de reacción.



En las curvas que se obtienen como respuesta, se elige dos puntos en los cuales la respuesta alcanza el 28.3% y el 63.2% de su valor final.

Figura 9. Curva de reacción para el modelo POR.



Donde:

ΔY = Máxima variación de la salida.

ΔU = Entrada aplicada al sistema.

t_1 = Tiempo en donde la respuesta alcanza el 28.3% de la salida.

t_2 = Tiempo en donde la respuesta alcanza el 63.2% de la salida.

De la gráfica se obtienen los valores de t_1 y t_2 para solucionar las siguientes ecuaciones se despejan los valores de θ' y τ , si al resolver las ecuaciones el valor de θ' es negativo, se asume que el sistema no tiene retardo, por lo que se hace $\theta' = 0$ y por lo tanto $\tau = t_2$.

$$\theta' + \frac{\tau}{3} = t_1 \quad \text{Ec. 4}$$

$$\theta' + \tau = t_2 \quad \text{Ec. 5}$$

El valor de la ganancia se obtiene mediante el cociente entre el cambio de la variable de salida y el cambio en la variable de entrada (valor del escalón de entrada).

$$K = \frac{\Delta Y}{\Delta U} \quad \text{Ec. 6}$$

El modelo de la planta se obtiene reemplazando los valores de K , θ' y τ en la ecuación de $G(s)$.

2.4 APLICACIÓN DE UN CONTROLADOR PID EN EL MÓDULO DE FLUJO.

Una parte de la investigación de este proyecto se fundamenta en la comparación de un modelo de control convencional como lo es el PID en contra de un modelo de control basado en Redes Neuronales Artificiales (RNAs), por lo que se hace necesario el diseño e implementación de un controlador convencional por lo que se hicieron los anteriores cálculos en el software de Matlab para llegar la función de transferencia $G(s)$. Con esta ecuación se procede a diseñar el controlador para la planta; en este caso se optó por un controlador PID por Ziegler–Nichols, por ser uno de los métodos más confiables por facilidad de cálculos, y mejores resultados del controlador.

2.4.1 Elementos del control convencional PID.

Definición. Un controlador es un dispositivo que compara el valor de la variable controlada con el valor deseado, establece la diferencia o error y la utiliza para realizar la acción de control, con el fin de reducir el error a 0 o a un valor mínimo aceptable.

Ecuación que rige el control PID. La ecuación que rige el controlador PID (proporcional integral derivativo), se basa en:

$$m(t) = K_c * \left[\ell(t) + \frac{1}{T_i} \int \ell(t) dt + T_d * \frac{d\ell(t)}{dt} \right] + M_o \quad \text{Ec. 7}$$

Donde:

K_c = Ganancia del controlador.

T_i = Tiempo integral (medido en seg. O min.).

T_d = Tiempo derivativo (medido en seg. O min.).

$\ell(t)$ = Error (set point-variable).

$m(t)$ = Salida del controlador

M_o = Salida del control cuando el error sea 0.

T = Periodo de muestreo.

Utilizando técnicas de métodos numéricos para evaluar las integrales y las derivadas de la ecuación anterior se obtiene el algoritmo de control PID, dado de la siguiente manera en la Ecuación (6)

$$m(k) = q_0 l(k) + q_1 l(k - 1) + q_2 l(k - 2) + m(k - 1) \quad \text{Ec. 8}$$

La ecuación (7) se obtiene de aplicar la transformada Z a la ecuación (6)

$$D(Z) = \frac{M(Z)}{E(Z)} = \frac{q_0 Z^2 + q_1 Z + q_2}{z(z-1)} \quad \text{Ec. 9}$$

En donde:

$$q_0 = k_c \left[1 + \frac{T}{2Ti} + \frac{Td}{T} \right] \quad \text{Ec. 10}$$

$$q_1 = -k_c \left[1 - \frac{T}{2Ti} + \frac{Td}{T} \right] \quad \text{Ec. 11}$$

$$q_2 = \frac{k_c Td}{T} \quad \text{Ec. 12}$$

Los términos de cada ecuación son:

q_0, q_1, q_2 = Ganancias del controlador

k_c = Ganancia del sistema

Td = Tiempo derivativo

Ti = Tiempo integral

T = Periodo de muestreo.

Forma de calcularlo. Para trabajar con el control PID, existen diferentes algoritmos para resolverlo ellos son: Ziegler–Nichols, criterio de Error, Ganancia limite, donde el Ziegler–Nichols es uno de los métodos más confiables por facilidad de cálculos, y mejores resultados del controlador; es decir para que el sistema trabaje más eficientemente. Primordialmente se debe de aplicar solo a sistemas de primer orden con retardo (POR), si por alguna razón se utiliza ecuaciones de orden superior se debe de llevar esta a orden uno, la ecuación que la rige es:

$$Gp(s) = \frac{K t^{\theta' s}}{\tau s + 1} \quad \text{Ec. 13}$$

Donde:

K = Ganancia del sistema

$\theta'S$ = Retardo del sistema
 τS = Constante de tiempo del sistema

En la tabla están los ajustes de los parámetros:

Tabla 9. Parámetros del controlador PID por Ziegler-Nichols.

Control / Parámetro	k_c	τ_i	τ_d
P	$\tau/k\theta$	-	-
PI	$0.9\tau/k\theta$	3.33θ	-
PID	$1.2\tau/k\theta$	2θ	0.5θ

Los parámetros del controlador PID dan pie a la obtención de buenos resultados, de sistemas estables, de respuesta rápida, permitiendo la adecuada identificación de los sistemas en el momento de realizar los diseños de los controladores a nivel industrial.¹¹

2.4.2 Cálculo del controlador por Ziegler-Nichols. Con los conceptos de la del numeral 2.5.1 se tiene claro lo necesario para calcular en controlador por Ziegler-Nichols, por lo que se tiene la función del sistemas de primer orden con retardo (POR) para hacer los cálculos necesarios.

$$G(s) = \frac{4.7227e^{-0.459841s}}{0.67585s + 1}$$

$$T = 0.749033$$

$$\theta = \theta' + T/2$$

$$\theta = 0.459841 + 0.749033/2$$

$$\theta = 0.8343575$$

$$k_c = 1.2 * \tau/k\theta$$

$$k_c = (1.2 * 0.67585)/(4.7227 * 0.8343575)$$

$$k_c = 0.205820$$

$$\tau_i = 2\theta = 1.668715$$

$$\tau_d = 0.5\theta = 0.417178$$

$$q_0 = k_c \left[1 + \frac{T}{2\tau_i} + \frac{\tau_d}{T} \right] = 0.183324$$

¹¹ GARCÍA. OP.CIT. p.194-198, [online]

$$q_1 = -k_c \left[1 - \frac{T}{2\tau_i} + \frac{\tau_d}{T} \right] = -0.194447$$

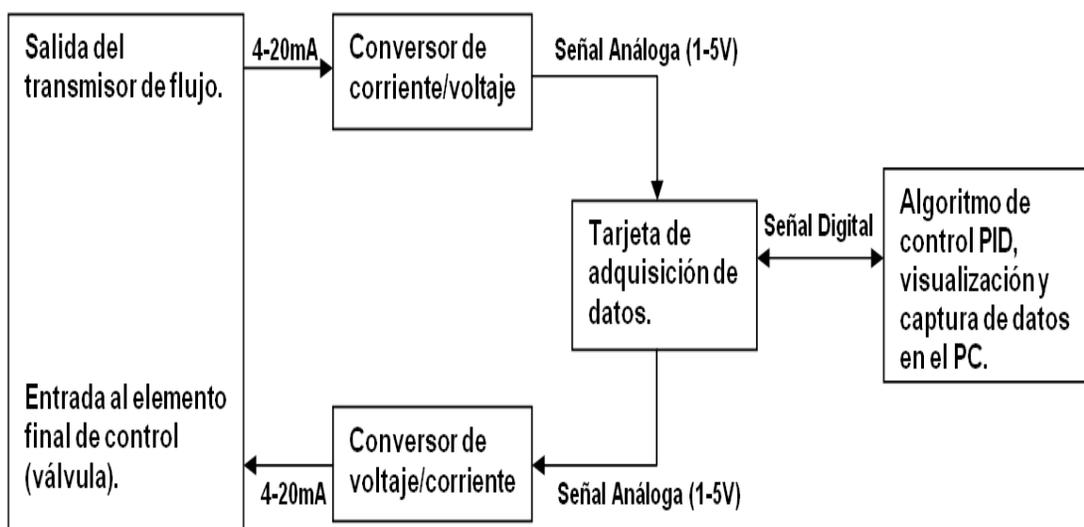
$$q_2 = \frac{k_c \tau_d}{T} = 0.057317$$

2.4.3 Implementación del controlador por Ziegler-Nichols. Después de tener calculada la función de transferencia del controlador, se procede a implementar el control en el proceso, para este es necesario contar con software LabView instalado en el computador, en este software se hace el desarrollo del algoritmo de control, visualización para el monitoreo de las variables y captura de los datos cada tiempo de muestreo para ir generando la base de datos, también, los datos capturados se procesan con algoritmo de control y con los cálculos realizados se da una salida para la tarjeta de adquisición que después la da al proceso, para dar esta salida, se tiene en cuenta el valor del set-point digitada por el usuario, el valor de la salida anterior, el estado de la variable que se está controlando, en este caso es el flujo de agua, con esto se calcula el error (set-point menos el valor de la variable controlada) con el cual se hace un registro de corrimiento para poder tener el valor actual del error y sus valores anteriores, necesarios para calcular la salida del controlador, repitiéndose este proceso una y otra vez.

Para la parte de interfaz de comunicación entre el computador y el modulo del flujo (proceso), es necesaria una tarjeta de adquisición de datos, que se encargaba de recibir los datos analógicos provenientes del módulo para convertirlos en datos digitales de manera que el computador los pueda procesar, para el caso contrario, en la comunicación del computador hacia el módulo, la tarjeta se encarga de convertir la señal de salida dada por el algoritmo de control que es digital a una señal analógica, pero como el módulo de flujo sólo trabaja con señales de corriente en un rango de 4-20mA y el microcontrolador de la tarjeta de adquisición trabaja con señales en un rango de 1-5V para la entrada y la salida, entonces la señal de salida del microcontrolador se le adiciona un conversor de voltaje a corriente, que cuando la salida en voltaje sea 1V el conversor entregue 4mA y cuando la salida sea 5V el conversor entregue 20mA, para el caso de la entrada, se adiciona un conversor de corriente a voltaje, de manera que cuando el módulo este entregando 4mA, el conversor entregue 1V y si está entregando 20mA, el conversor entregue 5V.

Figura 10. Diagrama de bloques del proceso de control PID.

Módulo De Flujo



Como se explicó antes, es necesario elaborar un convertor de corriente a voltaje y otro de voltaje a corriente, para lo cual se utilizó el amplificador operacional LM358 con una polarización de ± 12 voltios, los cuales se diseñaron para que conviertan de 1-5v a 4-20mA y 4-20mA a 1-5v, como se muestra en las figuras 11 y 12.

Figura 11. Convertor de voltaje a corriente.

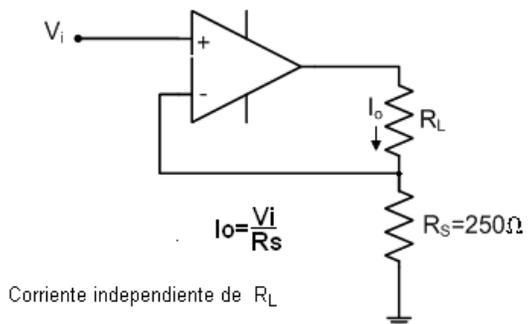
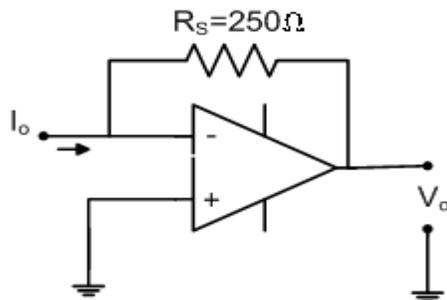


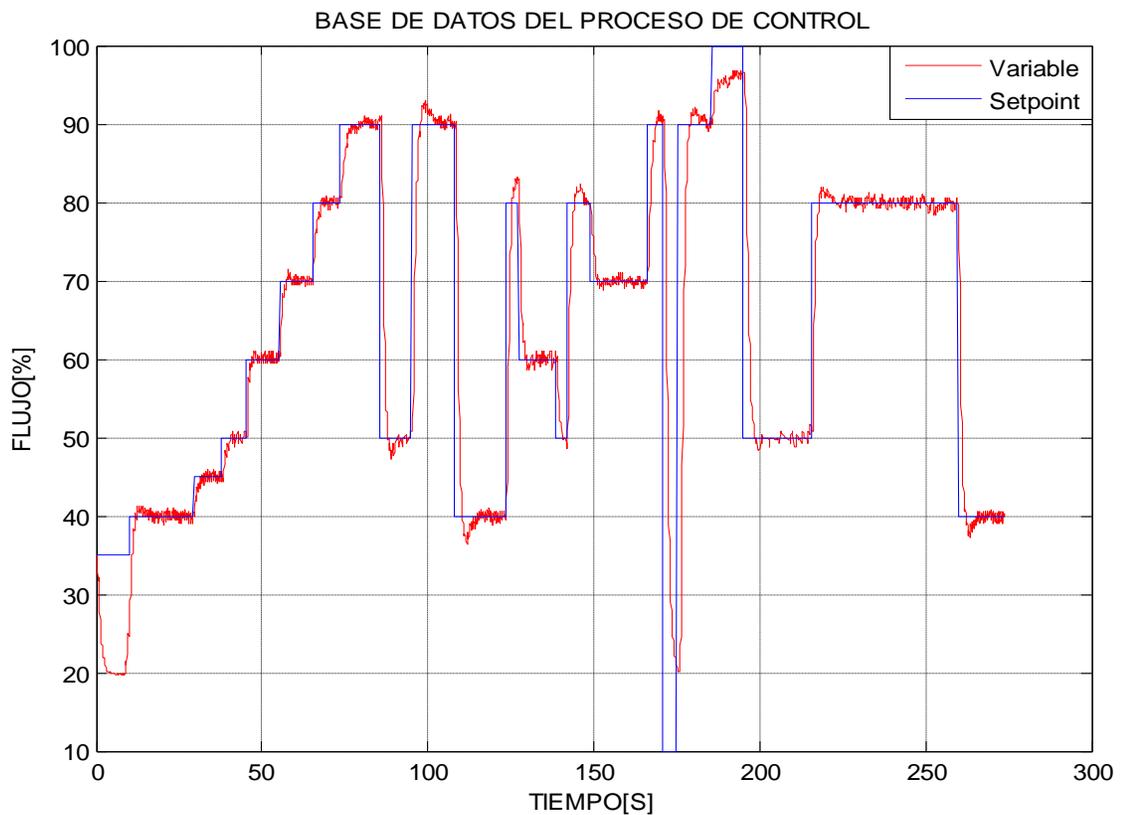
Figura 12. Convertor de corriente a voltaje.



2.5 GENERACIÓN DE LA BASE DE DATOS.

Al mismo tiempo que se tiene implementado el control PID en el proceso de flujo de agua, también se están guardando los datos cada tiempo de muestreo en un archivo plano, en este archivo queda reflejado el comportamiento de la variable controlada y cómo trata de alcanzar el set-point lo más rápido posible, y la acción necesaria del controlador en el elemento final de control (válvula) para que el error llegue a cero, que es el objetivo del control del proceso. Este archivo plano está comprendido por 4 columnas, en la cual primera columna es el tiempo de muestro, la segunda columna es el set-point, la tercera es la variable controlada y la cuarta columna es la acción del elemento final de control (la válvula), esta base de datos es necesaria para el entrenamiento de las redes neuronales artificiales, pero antes del entrenamiento a estos datos es necesario hacerles un procesamiento previo.

Figura 13. Base de datos del proceso de control.



2.6 ESTRUCTURACIÓN Y ADECUACIÓN DE LOS DATOS.

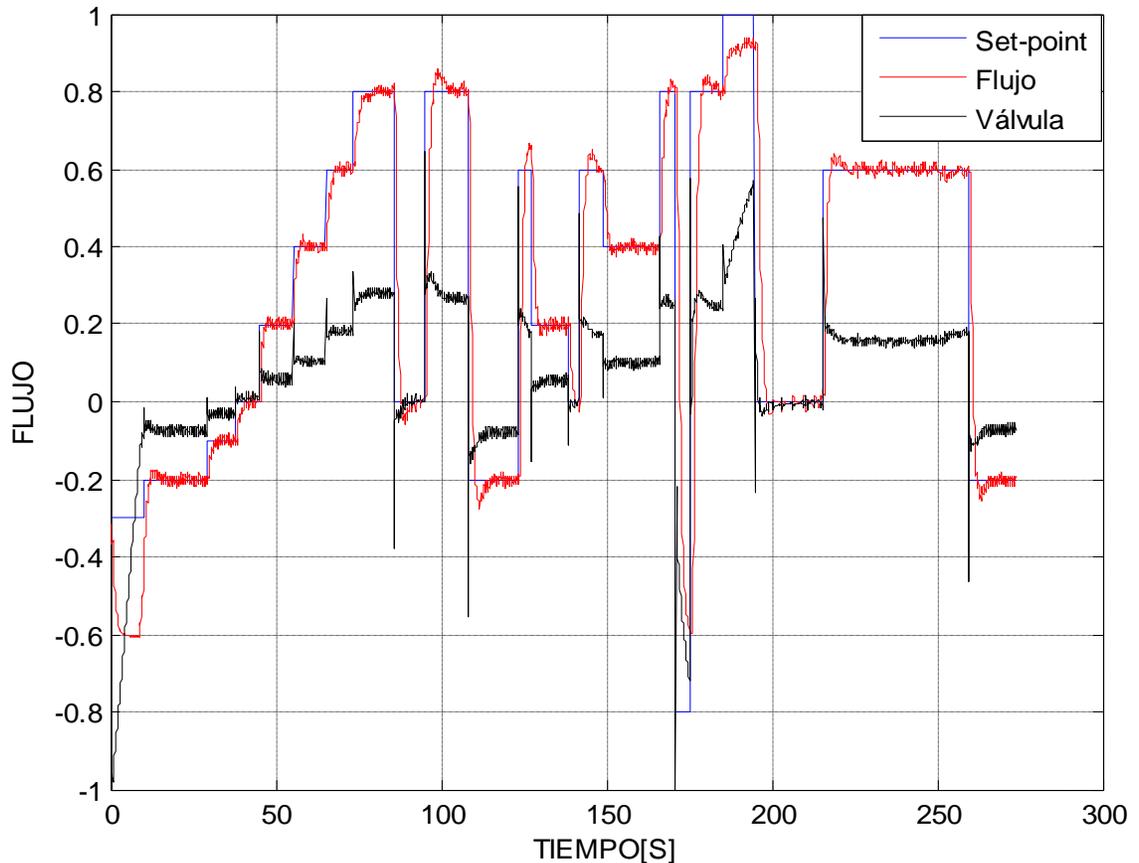
Esta estructuración y adecuación de los datos se lleva a cabo para un posterior entrenamiento de las redes neuronales artificiales, para esto es necesario que la información sea más completa y más variada, por lo cual se aplica derivada en diferencias en cada intervalo de muestreo a la variable controlada que es la tercera columna de la base de datos, de manera que se tenga más información para el entrenamiento de las RNA's (Redes Neuronales Artificiales).

$$y'(i) = \frac{y(i) - y(i - 1)}{t(i) - t(i - 1)}$$

Además, estos datos se encuentran en una escala muy amplia (entre 0 y 100), y como las redes neuronales entrenan y trabajan mejor en valores entre -1 y 1, por esto se hace un proceso de normalización con las herramientas de Matlab como lo es función *mapminmax* como lo muestra la figura donde se observa que el valor máximo es 1 y el mínimo es -1.

Figura 14. Base de datos escalada.

BASE DE DATOS NORMALIZADA ENTRE -1 Y 1



También, se utiliza otra función de Matlab para hallar la media y la varianza a las matrices de entrada y de salida de la red, la función utilizada para hacer estos cálculos es *mapstd*, con esta función se busca que los datos alejados de media que pueden ser errores sean suprimidos para evitar problemas en el entrenamiento ; además se utiliza la función *processpca* para hacer un análisis de componentes principales, para hacer este análisis es conveniente hacer una normalización de media y de varianza que es lo que se hace con función anterior.

3. EVALUACIÓN DEL COMPORTAMIENTO DE DIFERENTES MODELOS DE RED NEURONAL ARTIFICIAL PARA EL PROCESO DE CONTROL.

3.1 INTRODUCCIÓN A LA REDES NEURONALES.

El cerebro humano tiene varias características deseables en un sistema artificial, como lo es la robustez y tolerante a fallas, ya que diariamente mueren neuronas sin afectar su desempeño, es flexible, se ajusta a nuevos ambientes por aprendizaje, no hay que programarlo, puede manejar información difusa, con ruido o inconsistente, es altamente paralelo, es pequeño, compacto y consume poca energía.¹²

Además, el cerebro humano constituye una computadora muy notable, es capaz de interpretar información imprecisa suministrada por los sentidos a un ritmo increíblemente veloz. Logra discernir un susurro en una sala ruidosa, un rostro en un callejón mal iluminado y leer entre líneas un discurso; lo más impresionante de todo, es que el cerebro aprende sin instrucciones explícitas de ninguna clase, a crear las representaciones internas que hacen posibles estas habilidades.

Basados en la eficiencia de los procesos llevados a cabo por el cerebro, e inspirados en su funcionamiento, varios investigadores han desarrollado desde hace más de 30 años la teoría de las Redes Neuronales Artificiales (RNA), las cuales emulan las redes neuronales biológicas, y que se han utilizado para aprender estrategias de solución basadas en ejemplos de comportamiento típico de patrones; estos sistemas no requieren que la tarea a ejecutar se programe, ellos generalizan y aprenden de la experiencia.

La teoría de las RNA ha brindado una alternativa a la computación clásica, para aquellos problemas, en los cuales los métodos con computadores tradicionales no han entregado resultados muy convincentes, o poco convenientes. Las aplicaciones más exitosas de las RNA son:

- Procesamiento de imágenes y de voz.
- Reconocimiento de patrones.
- Planeamiento.
- Interfaces adaptivas para sistemas Hombre/máquina.
- Predicción.
- Control y optimización.
- Filtrado de señales.¹³

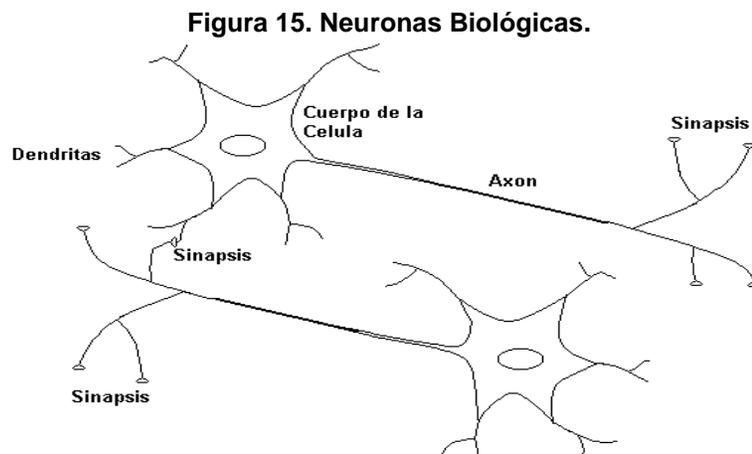
¹² DELGADO, Alberto. Inteligencia Artificial y Minirobots. Segunda edición. Bogotá: Ecoe ediciones, 1998. p. 1.

¹³ ACOSTA BUITRAGO, María Isabel; ZULUAGA MUÑOZ, Camilo Alfonso. Tutorial sobre redes neuronales aplicadas en ingeniería eléctrica y su implementación en un sitio web. Trabajo de grado Ingeniero electricista. Pereira: Universidad Tecnológica de Pereira. Facultad de Ingeniería Eléctrica, 2000, [online]

Los computadores tradicionales se basan en la arquitectura de Von Neumann que opera ejecutando serie (una tras otra) las instrucciones que componen u algoritmo que se codifica en forma de programa el cual se encuentra almacenado en memoria. En contraste una neurona no es secuencial. La red neuronal no ejecuta instrucciones, esta responde en paralelo a las entradas que se le presentan. El resultado no lo almacena en una posición de memoria, este consiste en un estado de la red cuando se logra un equilibrio. El conocimiento en una red neuronal no se almacena en instrucciones, el poder de la red está en su topología y en los valores de las conexiones (pesos) entre las neuronas.

Las redes neuronales no ejecutan programas, las redes reaccionan, aprenden y se auto-organizan. Para entrenar una red neuronal se le presenta de manera repetitiva y sistemática una serie de entradas tipo con sus respectivas salidas.¹⁴

3.1.1 Neurona Biológica. El cerebro consta de un gran número (aproximadamente 10^{11}) de elementos altamente interconectados (aproximadamente 10^4 conexiones por elemento), llamados neuronas. Estas neuronas tienen tres componentes principales, las dendritas, el cuerpo de la célula o soma, y el axón como se muestra en la figura 15. Las dendritas, son estructuras que parten del soma con ramificaciones, se especializan en la recepción de las señales de otras células nerviosas por medio de conexiones sinápticas. El cuerpo de la célula o soma contiene el núcleo y es el encargado de las actividades metabólicas de toda la neurona, el soma puede recibir información de otras neuronas a través de la sinápsis en su superficie.



El axón permite enviar impulsos a otras células nerviosas. En algunas neuronas los impulsos se inician en la unión del axón y el soma y luego se transmite a lo largo del axón a otras células nerviosas. Cuando el axón está cerca de sus células

¹⁴ DELGADO. OP.CIT. Inteligencia Artificial y Minirobots. p.1

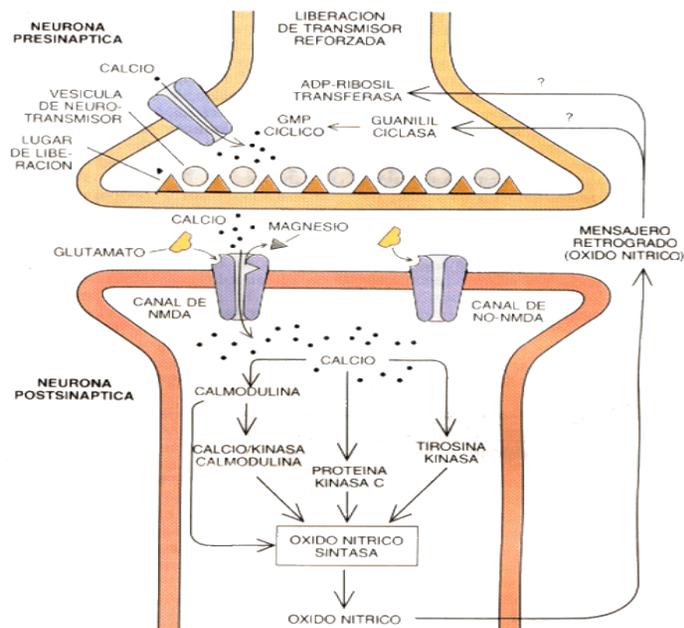
destino se divide en muchas ramificaciones que forman sinápsis con el soma o axón de otras células. La sinápsis es una conexión entre dos células nerviosa. Las sinápsis pueden ser excitativas o inhibitorias según el neurotransmisor que se libere, cada neurona recibe de 10.000 a 100.000 sinapsis y su axón realiza una cantidad similar de sinapsis.

Las sinapsis se clasifican según su posición en la superficie de la neurona receptora en tres tipos: axo-somática, axo-dendrítica, axo-axónica. Los fenómenos que ocurren en la sinapsis son de naturaleza química, pero tienen efectos eléctricos laterales que se pueden medir.¹⁵

En la figura 16 se visualiza el proceso químico de una sinapsis y los diferentes elementos que hacen parte del proceso tanto en la neurona presináptica, como en la postsináptica.

Las RNA no alcanzan la complejidad del cerebro, sin embargo hay dos aspectos similares entre redes biológicas y artificiales, primero los bloques de construcción de ambas redes son sencillos elementos computacionales (aunque las RNA son mucho más simples que las biológicas) altamente interconectados; segundo, las conexiones entre neuronas determinan la función de la red.¹⁶

Figura 16. Proceso químico de una sinapsis.

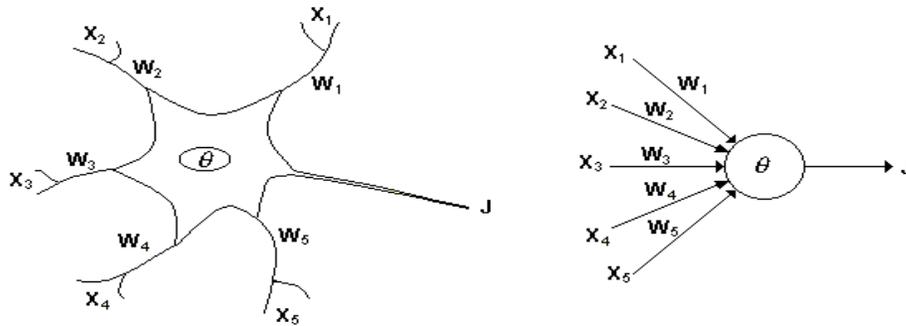


¹⁵ DELGADO. OP.CIT. p.3-4.

¹⁶ ACOSTA; ZULUAGA. OP.CIT. p.15-16.

3.1.2 Neurona Artificial. El modelo de una neurona artificial es una imitación del proceso de una neurona biológica, existen varias formas de nombrar una neurona artificial, es conocida como nodo, neuronodo, celda, unidad o elemento de procesamiento (PE); En la figura 17 se observa un PE en forma general y su similitud con una neurona biológica.

Figura 17. Similitud entre neurona biológica y artificial.



- Las entradas X_i representan las señales que provienen de otras neuronas y que son capturadas por las dendritas.
- Los pesos W_i son la intensidad de la sinápsis que conecta dos neuronas; tanto X_i como W_i son valores reales.
- θ es la función umbral que la neurona debe superar para activarse; este proceso ocurre biológicamente en el cuerpo de la célula.

Las señales de entrada a una neurona artificial X_1, X_2, \dots, X_n son variables continuas en lugar de pulsos discretos, como se presentan en una neurona biológica. Cada señal de entrada pasa a través de una ganancia o peso, llamado peso sináptico o fortaleza de la conexión cuya función es análoga a la de la función sináptica de la neurona biológica. Los pesos pueden ser positivos (excitatorios), o negativos (inhibitorios), el nodo sumatorio acumula todas las señales de entradas multiplicadas por los pesos o ponderadas y las pasa a la salida a través de una función umbral o función de transferencia. La entrada neta a cada unidad puede escribirse de la siguiente manera en la ecuación.¹⁷

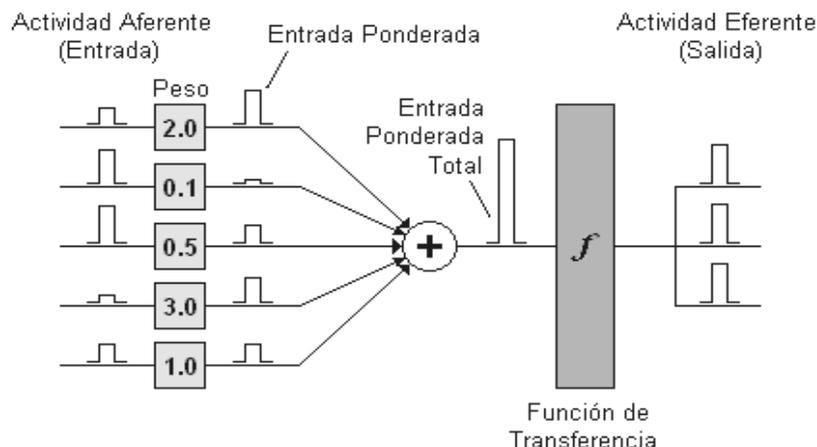
$$neta_i = \sum_{i=1}^n W_i X_i = \vec{X} \vec{W}$$

Ec. 14

Para entender de una manera más clara el proceso se ilustra con la figura 18, en donde se puede observar el trayecto de las señales de entrada hasta llegar a la salida.

¹⁷ ACOSTA; ZULUAGA. OP.CIT. p.16-17.

Figura 18. Proceso de una red neuronal artificial.



Una vez que se ha calculado la activación del nodo, el valor de salida equivale a

$$x_i = f_i(neta_i)$$

Ec. 15

Donde f_i representa la función de activación para esa unidad, que corresponde a la función escogida para transformar la entrada $neta_i$ en el valor de salida x_i y que depende de las características específicas de cada red.

3.1.3 Función de activación. La función de activación o función de transferencia determina la salida total de la red, la cual puede ser una lineal o no lineal de n , y que se escoge según los requerimientos de problema que la neurona tenga que resolver; aunque las RNA se inspiren en modelos biológicos no existe ninguna limitación para realizar modificaciones en las funciones de salida, así que se encontrarán modelos artificiales que nada tienen que ver con las características del sistema biológico.¹⁸

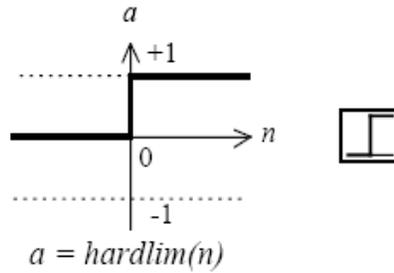
3.1.3.1 Limitador fuerte (Hardlim): La figura 19, muestra como esta función de transferencia acerca la salida de la red a cero, si el argumento de la función es menor que cero y la lleva a uno si este argumento es mayor que uno. Esta función crea neuronas que clasifican las entradas en dos categorías diferentes, característica que le permite ser empleada en la red tipo Perceptrón.

$$a = \begin{cases} 1 & \text{si } n \geq 0 \\ 0 & \text{si } n < 0 \end{cases}$$

Ec. 16

¹⁸ ACOSTA; ZULUAGA. OP.CIT. p.20-24.

Figura 19. Función de transferencia Hardlim.

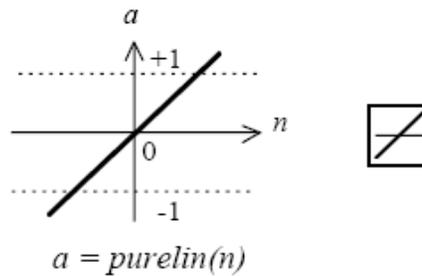


3.1.3.2 Función de transferencia lineal (purelin): La salida de una función de transferencia lineal es igual a su entrada.

$$a = n$$

Ec. 17

Figura 20. Función de transferencia lineal.



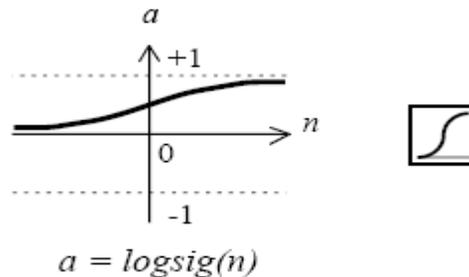
3.1.3.3 Función de transferencia sigmoideal (logsig): Esta función toma los valores de entrada, los cuales pueden oscilar entre más y menos infinito, y restringe la salida a valores entre cero y uno, de acuerdo a la expresión.

$$a = \frac{1}{1+e^{-n}}$$

Ec. 18

Esta función es comúnmente usada en redes multicapa, como la Backpropagation, en parte porque la función logsig es diferenciable.¹⁹

Figura 21. Función de transferencia sigmoideal.



¹⁹ DEMUTH, Howard; BEALE, Mark; HAGAN, Martin. Neural Network Toolbox 5. USA: The MathWorks, Inc. p. 45-46

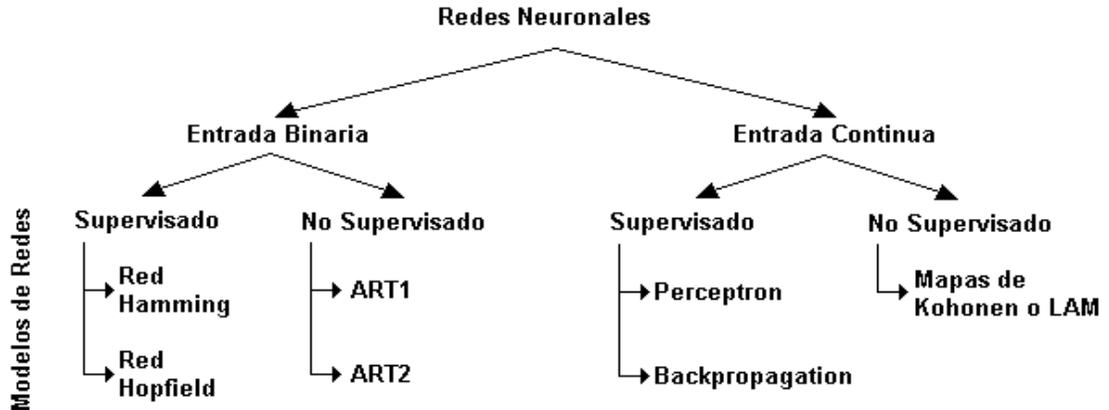
La tabla 10 hace una relación de las principales funciones de transferencia empleadas en el entrenamiento de redes neuronales.

Tabla 10. Funciones de Transferencia.

Nombre	Relación Entrada /Salida	Icono	Función
Limitador Fuerte	$a = 0 \quad n < 0$ $a = 1 \quad n \geq 0$		hardlim
Limitador Fuerte Simétrico	$a = -1 \quad n < 0$ $a = +1 \quad n \geq 0$		hardlims
Lineal Positiva	$a = 0 \quad n < 0$ $a = n \quad 0 \leq n$		poslin
Lineal	$a = n$		purelin
Lineal Saturado	$a = 0 \quad n < 0$ $a = n \quad 0 \leq n \leq 1$ $a = 1 \quad n > 1$		satlin
Lineal Saturado Simétrico	$a = -1 \quad n < -1$ $a = n \quad -1 \leq n \leq 1$ $a = +1 \quad n > 1$		satlins
Sigmoidal Logarítmico	$a = \frac{1}{1 + e^{-n}}$		logsig
Tangente Sigmoidal Hiperbólica	$a = \frac{e^n - e^{-n}}{e^n + e^{-n}}$		tansig
Competitiva	$a = 1$ Neurona con n max $a = 0$ El resto de neuronas		compet

3.1.4 Clasificación de las redes neuronales. En general las redes neuronales se pueden clasificar de diversas maneras, según su topología, forma de aprendizaje (supervisado o no supervisado), tipos de funciones de activación, valores de entradas (binarias o continuas), como lo muestra la figura 22.

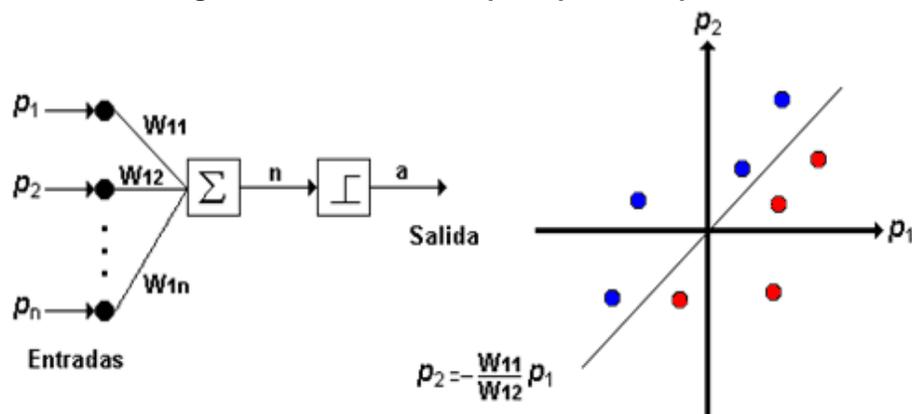
Figura 22. Clasificación de las redes neuronales.



3.2 ALGUNOS MODELOS DE REDES NEURONALES ARTIFICIALES.

3.2.1 Perceptrón. El perceptrón simple o perceptrón es el tipo de red neuronal más sencillo que se puede implementar. Este tipo de red sólo sirve para clasificar los elementos pertenecientes a dos clases linealmente separables, el perceptrón para obtener la salida realiza una suma ponderada de las entradas, resta el umbral θ y pasa el resultado por la función de activación²⁰. El perceptrón utiliza funciones de activación, hardlim con salidas 1, 0 o hardlims con salidas 1, -1; todo depende de la salida esperada para la red; sin embargo es preferible usar la función hardlims, ya que el tener un cero multiplicando algunas de los valores resultantes del producto de las entradas por el vector de pesos, ocasiona que estos no se actualicen y que el aprendizaje sea más lento.²¹

Figura 23. Estructura del perceptrón simple.



²⁰ RODRÍGUEZ PONCE, Héctor Uriel. Perceptrón Multicapa para Reconocimiento de Objetos sobre Planos. Puebla, México: Universidad de las Américas Puebla, 2004. p. 21, [online].

²¹ ACOSTA; ZULUAGA. OP.CIT. p.39.

3.2.1.1 Regla de aprendizaje. El Perceptrón es un tipo de red de aprendizaje supervisado, es decir necesita conocer los valores esperados para cada una de las entradas presentadas; su comportamiento está definido por pares de esta forma:

$$\{p_1, t_1\}, \{p_2, t_2\}, \dots, \{p_Q, t_Q\} \quad \text{Ec. 19}$$

La salida de la red tipo perceptrón está dada por la ecuación (18).

$$a = f(\sum_i w_i p_i) \quad \text{Ec. 20}$$

En el proceso de entrenamiento el Perceptrón se expone a un conjunto de patrones de entrada y los pesos de la red son ajustados de forma que al final de entrenamiento se obtengan las salidas esperadas para cada unos de esos patrones de entrada.

El algoritmo de entrenamiento del Perceptrón puede resumirse en los siguientes pasos:

1. Se inicializa la matriz de pesos y el valor de la ganancia, por lo general se asignan valores aleatorios a cada uno de los pesos w_i y al valor b .
2. Se presenta el primer patrón a la red, junto con la salida esperada en forma de pares entrada/salida.
3. Se calcula la salida de la red por medio de.

$$a = f(w_1 p_1 + w_2 p_2 + b) \quad \text{Ec. 21}$$

Donde f puede ser hardlim o hardlims

4. Cuando la red no retorna la salida correcta, es necesario alterar el valor de los pesos, tratando de llevarlo hasta p y así aumentar las posibilidades de que la clasificación sea correcta, una posibilidad es adicionar p a w haciendo que el vector w apunte en la dirección de p , y de esta forma después de repetidas presentaciones de p a la red, w se aproximará asintóticamente a p ; este es el procedimiento adoptado para la regla de aprendizaje del Perceptrón.²²

El proceso de aprendizaje del Perceptrón puede definirse en tres reglas, las cuales cubren la totalidad de combinaciones de salidas y sus correspondientes valores

²² ACOSTA; ZULUAGA. OP.CIT. p. 45.

esperados. Estas reglas utilizando la función de transferencia *hardlim*, se expresan como sigue:

$$e = t - a \quad \text{Ec. 22}$$

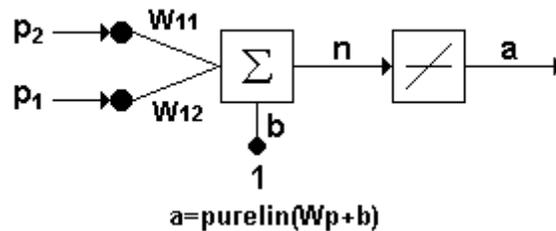
$$w^{nuevo} = w^{anterior} + e \cdot p = w^{anterior} + (t - a)p \quad \text{Ec. 23}$$

Y extendiendo la ley a las ganancias

$$b^{nueva} = b^{anterior} + e \quad \text{Ec. 24}$$

3.2.2 Adaline. Otro modelo de red neuronal artificial es el adaline, introducida por Widrow en 1959, cuyo nombre proviene de ADaptive Linear Neuron. Este modelo utiliza una red muy similar a la del perceptrón, pero de respuesta lineal (figura 24), cuyas entradas pueden ser continuas. Por otra lado, diferencia del nodo asociador lineal, el adaline incorpora un parámetro de adicional denominado bias, que se puede traducir como umbral, aunque debe tenerse en cuenta que no se trata de un umbral de dispara como en el perceptrón, sino de un parámetro que proporciona un grado de libertad adicional.

Figura 24. Estructura del adaline.



Sin embargo, la diferencia más importante con el perceptrón reside en la regla de aprendizaje que implementa. En el adaline se utiliza la regla de Widrow-Hoff, también conocida como regla LMS (least Mean Squares, mínimos cuadrados), que conduce a actualizaciones de tipo continuo, siendo la actualización del error proporcional al error que la neurona comete.²³

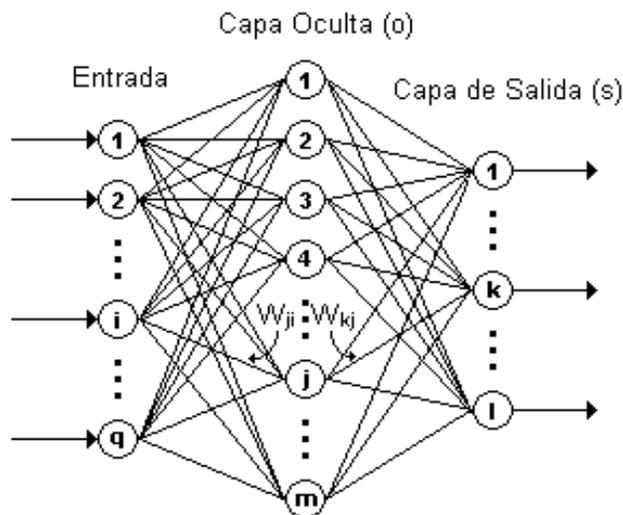
La red Adaline presenta la misma limitación del Perceptrón en cuanto al tipo de problemas que pueden resolver, ambas redes pueden solo resolver problemas linealmente separables, sin embargo el algoritmo LMS es más potente que la regla de aprendizaje del Perceptrón ya que minimiza el error medio cuadrático, la regla

²³ MARTÍN DEL BRIO, Bonifacio; SANZ MOLINA, Alfredo. Redes neuronales y sistemas difusos. España: Alfa omega, 2001.p 55-56.

servió de inspiración para el desarrollo de otros algoritmos, este es el gran aporte de esta red.²⁴

3.2.3 Perceptrón multicapa. El perceptrón multicapa, MLP por sus siglas en inglés, se deriva del concepto de perceptrón simple, con la diferencia que en lugar de una sola neurona, tiene capas de neuronas intermedias que van ocultas, lo que quiere decir que no pertenecen ni a la entrada ni a la salida de la red. Este tipo de redes, al estar compuestas por varias capas, tiene la característica de ser no lineal, lo que quiere decir que es capaz de clasificar entradas que pertenecen a dos o más clases que no son linealmente separables.²⁵ En la figura 25 se puede observar la estructura típica de una red multicapa.

Figura 25. Estructura del perceptrón multicapa.



Donde:

q : Equivale al número de componentes el vector de entrada.

m : Número de neuronas de la capa oculta.

l : Número de neuronas de la capa de salida.

Esta arquitectura suele entrenarse mediante el algoritmo denominado retropropagación del error o BP, o bien haciendo uso de alguna de sus variantes o derivados, motivo por el cual en muchas ocasiones el conjunto de arquitectura

²⁴ ACOSTA; ZULUAGA. OP.CIT. p.63.

²⁵ IBID. p.88-89.

MLP + aprendizaje BP suele denominarse red de retropropagación o simplemente BP.²⁶

3.2.3.1 Algoritmo de retropropagación del error o BP. El BP (Back-Propagation) tiene el mismo objetivo que proceso de aprendizaje utilizado en el perceptrón simple, usar la diferencia entre las salidas deseadas y las salidas actuales de la capa de salida de la red para cambiar los pesos (iniciados en valores muy pequeños) con el fin de reducir el mínimo esta diferencia (error). Esto se logra mediante una serie de iteraciones donde se modifica cada peso de derecha a izquierda (en sentido inverso a la propagación de información de la red) hasta modificarse los pesos de la capa de entrada e iniciando nuevamente el proceso de propagación de la información de entrada, esto hasta que la diferencia entre la salida deseada y la obtenida en cada neurona de salida sea mínima.²⁷

El Back-Propagation hace uso de la ecuaciones (23), (24) conocida como regla delta, (25), (26) (descenso por gradiente), y la (27) (función de activación) en el proceso del ajuste de los pesos que conectan a la salida de la neurona i a la entrada de neurona j en la iteración n .

$$w_{ji}(n + 1) = w_{ji}(n) + \Delta w_{ji}(n) \quad \text{Ec. 25}$$

$$\Delta w_{ji}(n) = \eta \delta_j(n) \cdot y_i(n) \quad \text{Ec. 26}$$

$$\delta_j(n) = \begin{cases} [t_j(n) - y_j(n)] \cdot y_j'(n) & \text{para la capa de salida} \\ y_j'(n) \cdot \sum_k \delta_k(n) \cdot w_{kj}(n) & \text{para las capas ocultas} \end{cases} \quad \text{Ec. 27}$$

$$\delta_k(n) = [t_j(n) - y_j(n)] \cdot y_j'(n) \quad \text{Ec. 28}$$

$$y_j(n) = \frac{1}{1 + e^{-v_j(n)}} \quad \text{Ec. 29}$$

Donde k es el subíndice específico de la capa de salida y se utiliza j como subíndice para las demás capas, $v_j(n)$ es la salida inmediata de la neurona j en la posición n de los datos, $[t_j(n)]$ es la salida deseada, $y_j(n)$ es la salida obtenida de la función de activación de la neurona, $y_j'(n)$ es la derivada de función de activación de cada neurona con respecto a $v_j(n)$, y η es el parámetro de aprendizaje entre 0 y 1.

²⁶ MARTÍN DEL BRÍO; SANZ MOLINA. OP.CIT. p.63.

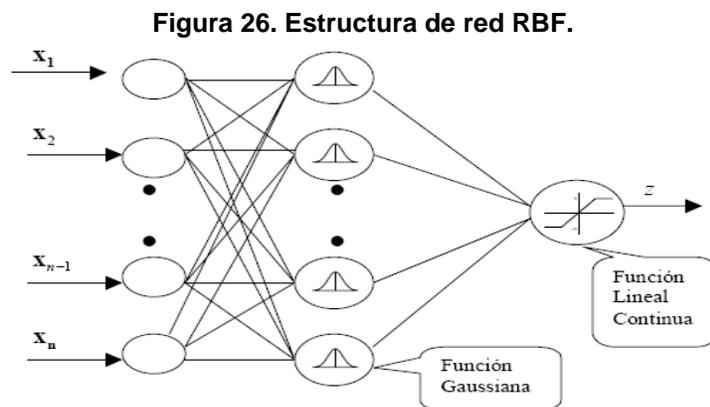
²⁷ RODRÍGUEZ PONCE, Héctor Uriel. Perceptrón Multicapa para Reconocimiento de Objetos sobre Planos. p. 27-28.

En el Back-Propagation, $\delta_j(n)$ es el gradiente local del valor instantáneo de la suma de los errores cuadráticos obtenidos en las neuronas de salida.²⁸ Este gradiente es el que orienta el entrenamiento para que el error sea mínimo para todo en conjunto de datos. La demostración del gradiente es algo extensa por lo que no se desarrolla, además, no es uno de los del trabajo.

3.2.4 Funciones de base radial (RBF). El modelo de funciones de base radial o RBF (Radial Basis Functions) a menudo se estudia junto con el MLP por ser una red unidireccional para aproximación funcional, pero que puede considerarse de tipo híbrido por que incorpora aprendizaje supervisado y no supervisado. Como pasa en el caso del MLP, la RBF permiten modelar con relativa facilidad sistemas no lineales arbitrarios, con la particularidad de que el tiempo que se necesita para su entrenamiento suele ser mucho más reducido que en el caso de BP clásico.

La arquitectura de una red RBF cuenta con tres capas de neuronas, de entradas, ocultas y capa de salida. Las neuronas de entrada, como suele ser habitual, simplemente envían la información del exterior hacia las neuronas de la capa oculta, la neuronas de la capa oculta completamente interconectada entre todos sus nodos con la capa de entrada y activada a través de la función radial (gaussiana) como se muestra en la figura 26, y las neuronas de la capa de salida son lineales, esencialmente calculan la suma ponderada de las salidas que proporciona la capa oculta.²⁹

El entrenamiento, a diferencia de la red usando backpropagation, es solamente hacia delante. De este modo, la salida z de una red RBF, en general, está influenciada por una transformación no lineal originada en la capa oculta a través de la función radial y una lineal en la capa de salida a través de la función lineal continua.



²⁸ RODRÍGUEZ PONCE, Héctor Uriel. Perceptrón Multicapa para Reconocimiento de Objetos sobre Planos. p. 28.

²⁹ MARTÍN DEL BRIO, Bonifacio; SANZ MOLINA, Alfredo. Redes neuronales y sistemas difusos. p. 152.

En el RBF Los nodos ocultos contienen una función base radial, la cual tiene como parámetros a centro y ancho. En el caso del centro, para cada función radial involucrada en la capa oculta. Regularmente, definen un vector de la misma dimensión del vector de entrada y hay normalmente un centro diferente por cada nodo de la capa oculta. Por otro lado, el ancho es el término empleado para identificar a la amplitud de la campana de gauss originada por la función radial. Es decir, la desviación estándar de la función radial. Algunos autores consideran a este ancho como un valor constante para cada una de las funciones radiales consideradas en la capa oculta y de este modo, así contribuiría a simplificar los pasos de construcción del modelo de entrenamiento de la red.

El primer cálculo efectuado en la capa oculta es hallar en un nodo de la capa oculta la distancia radial (distancia euclidiana) d entre el vector de entrada x , con n observaciones, a ese nodo en particular y el centro de gravedad c de ese mismo nodo. Es decir:

$$d = \|x - c\| = \sqrt{(x_1 + c_1)^2 + (x_2 + 2)^2 + \dots + (x_n + c_n)^2} \quad \text{Ec. 30}$$

Este valor d es un componente de la entrada para activar la función radial $G(\bullet)$. Este valor establece la principal diferencia con las redes MLP, entre ellas la backpropagation, quienes incluyen el producto interno en sus capas ocultas de las entradas por sus respectivos pesos. En cuanto a la función radial $G(\bullet)$, siendo una de las más comunes e^{-r^2} , siendo r el contenido evaluado en cada nodo de la capa oculta. En este caso particular, el contenido evaluado en cada nodo es la distancia euclidiana d . De ahí la expresión, entonces sería e^{-d^2} . Una de las derivaciones del modelo RBF es emplear el ancho (desviación estándar) para activar la función $G(\bullet)$. En este caso se estaría trabajando con algo como $e^{-d^2/a}$, donde a es el ancho para ese nodo oculto. En la capa oculta, en la medida que los valores de entrada se parezcan más a un centro su distancia tenderá a cero y de este modo la función gaussiana se dispararía a las vecindades de uno. Por otro lado, en la medida que los valores de entrada no se parezcan a su centro la distancia será mayor y la función radial parecería tender a cero. Este proceso es una clasificación no lineal de las entradas.

En la capa de salida del modelo RBF, los valores obtenidos en las salidas de la capa oculta serían transformados por la función lineal que permite aproximar los

valores z a los valores deseados, mediante la combinación lineal que se sucede en esta capa entre sus pesos y el resultado de aplicar la función radial³⁰. Es decir:

$$z = \sum w_i G(\bullet) \quad \text{Ec. 31}$$

3.3 ENTRENAMIENTO DE LOS MODELOS DE REDES NEURONALES ARTIFICIALES.

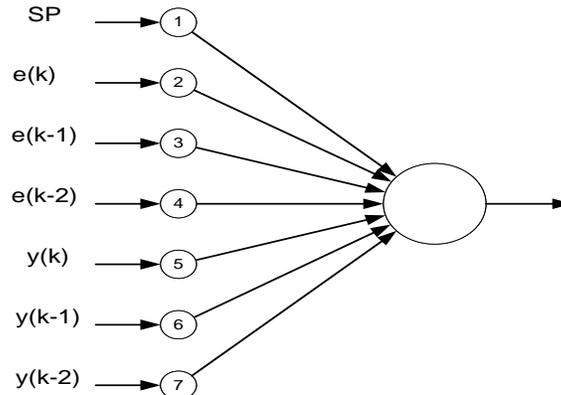
En esta etapa del proceso se tiene en cuenta las características del problema que se desea solucionar, como el número de variable de entrada, en número de variables de salida, tipo de función de activación según los requerimientos, en modelo de red neuronal artificial más adecuado, con que tanto volumen de información cuenta la base de datos para realizar el proceso de entrenamiento, validación y test de la red.

Para escoger cual es la red neuronal que mejores condiciones tiene para el proceso de control de flujo de agua, iniciaremos con una red tipo perceptrón simple, aunque de ente mano se puede concluir que no es apta para este tipo de proceso ya que el perceptrón solamente permite discriminar entre dos regiones linealmente separable, lo que no ocurre en este caso, el perceptrón se puede usar en controles de tipo on-off. Este entrenamiento de esta red solo se lleva a cabo de manera didáctica para apreciar su comportamiento. El entrenamiento de todas las redes se desarrollara con las herramientas disponibles en Matlab como lo es el Neural Network Toolbox.

3.3.1 Entrenamiento del preceptrón simple. Para el entrenamiento del perceptrón se cuenta los requerimientos del problema como son el número de entrada y salida, que para este problema son siete entrada, una salida, con representado por la figura 27 y la función de activación de tipo Tangente Sigmoidal Hiperbólica (tansig) ya que los valores de la salida deseada varían entre -1 y 1 por lo que las otras funciones como hardlim y el hardlims no son de utilidad, estas características como es la función de activación, numero de entrad, entre otras, hacen parte de los parámetros a ingresar al momento de crear la red en Matlab con función newp, que es la utilidad para este tipo de red.

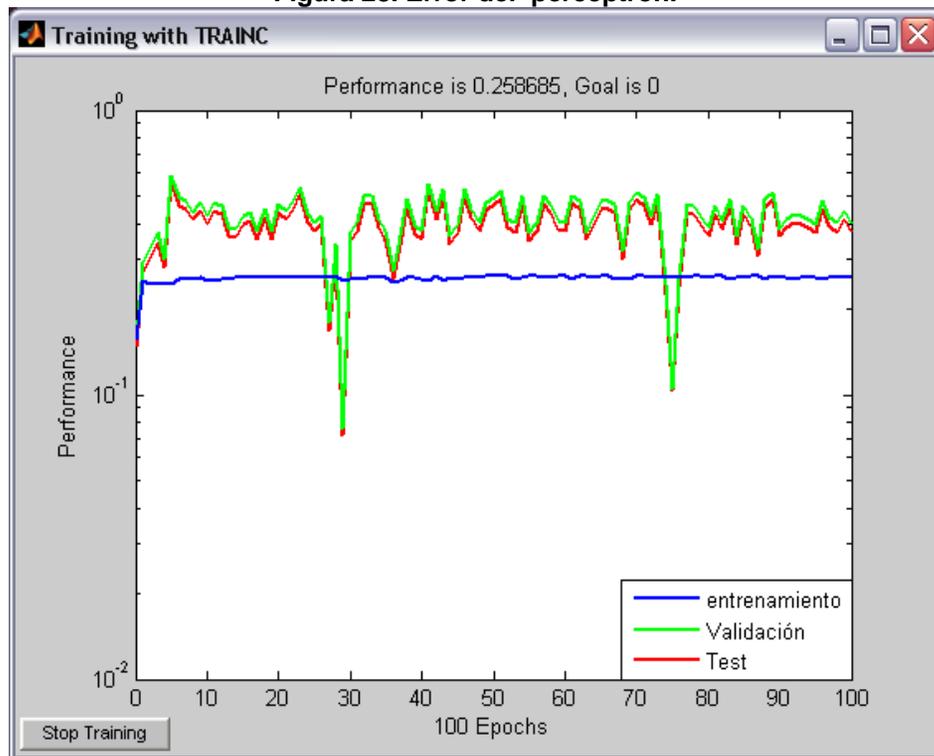
³⁰ COLMENARES LACRUZ, Gerardo. Redes neuronales. Mérida, Venezuela: Universidad de Los Andes, Instituto de Investigaciones Económicas y Sociales, [online].

Figura 27. Topología de la red perceptrón.



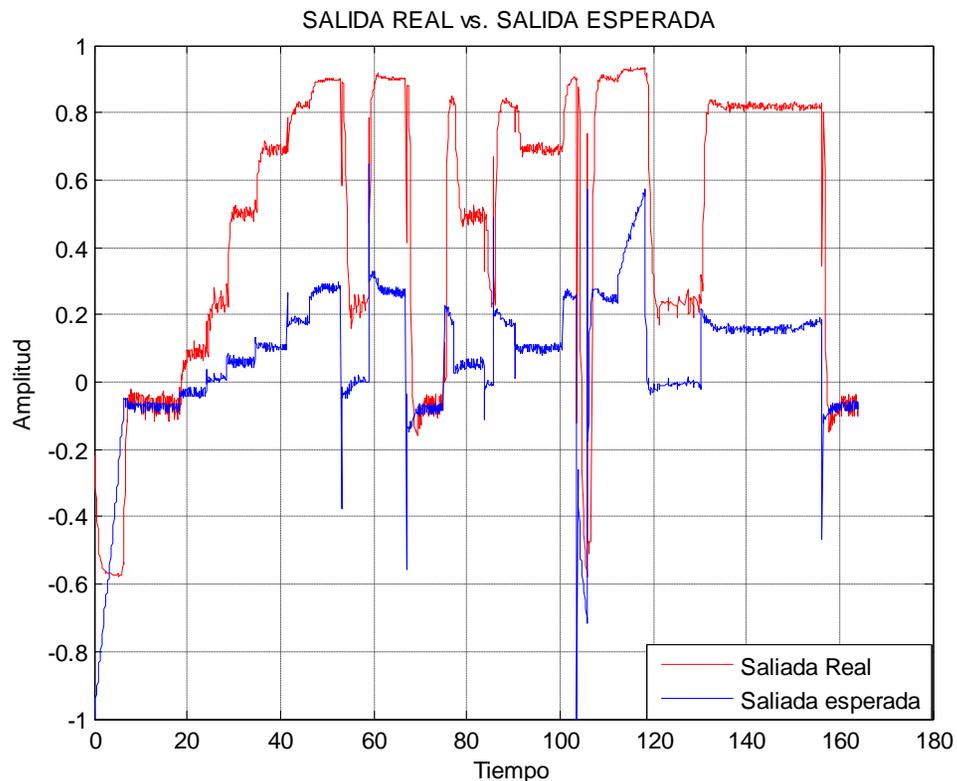
Para el caso de la función de la red tipo perceptrón en Matlab, se le carga el conjunto de datos de entrenamiento, validación y test, se dan los parámetros de la red, esto se puede entender más claramente en el ANEXO D donde se muestra el programa para el entrenamiento de las distintas redes. Con el conjunto de datos de entrada y salida, el perceptrón arrojó el resultado del error de los datos de entrenamiento, error de validación y el error de test, donde el máximo error logro un valor de 0.258685, mayor que error máximo permitido para la red que se había determinado como cero, este error lo alcanzó durante 100 épocas de entrenamiento, como se muestra en la figura 28.

Figura 28. Error del perceptrón.



Además, de los error también se puede simular la salida de la red ante la entrada del conjunto de datos de entrenamiento, esto para observar si la salida de la red se aproxima a los valores deseados. En la figura 29 se aprecia claramente lo que se decía con anterioridad, con respecto a las limitaciones del perceptrón simple, ya que el objetivo de la red no es linealmente separable como lo es la salida de una red de este tipo, también de la figura 29 se puede decir que La salida de la red es muy similar al SP (set-point), que fue uno de los parámetros de entrenamiento lo que es un caso muy particular ya que aproximó su salida a una variable de entrada.

Figura 29. Simulación del perceptrón vs. Salida esperada.



3.3.2 Entrenamiento del perceptrón multicapa (MLP). En este caso a diferencia del perceptrón simple que tiene una sola neurona por lo que no hay ningún problema; a diferencia de este tipo de modelo con el que se tiene una complicación para escoger las características de la red, ya que no hay ningún parámetro o condición que determine cuantas capas ocultas debe tener, o cuantas neuronas en cada capa, a diferencia de la capa de salida que ya se sabe de antemano cuantas entradas son, según el planteamiento del problema al igual que las funciones de activación de cada capa. Por lo anterior, se probaran varias topologías para determinar cuál de toda ellas es la que tiene mejores resultados.

Al igual que en proceso del entrenamiento del perceptrón, en el MLP se procede a cargar los datos de entrenamiento, validación y test, se dan los parámetros de la red como capas ocultas, número de neuronas en cada capa oculta, funciones de activación de cada capa, también, se inicializa la red, se ingresan parámetros como el número de iteraciones, máximo error permitido, entre otros parámetros.

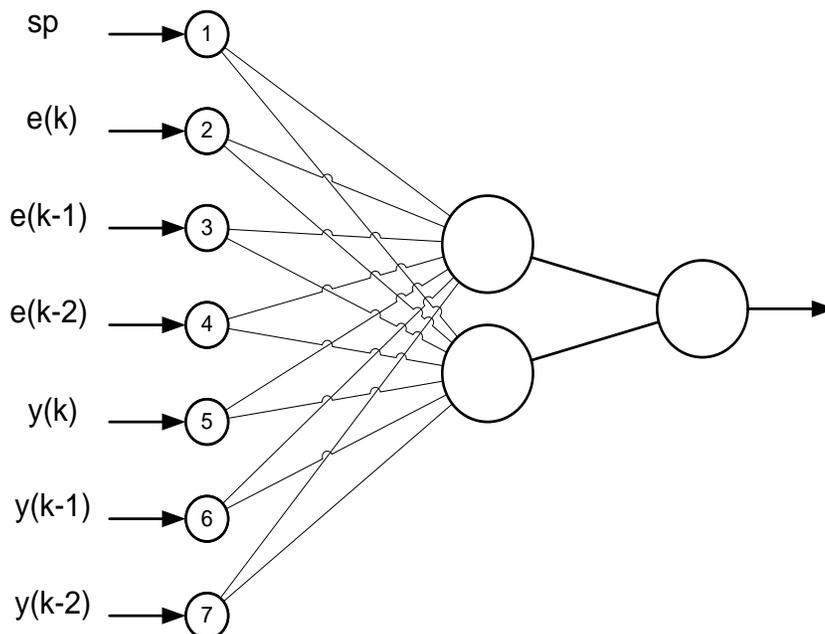
En esta topología se uso funciones de activación del tipo tansig en todas las capas, debido a que la salida deseada varía en números positivos y negativos (-1,1) por lo que fue la función de activación más indicada.

Para el primer caso se entrena un MLP con las siguientes características:

Tabla 11. Valores de la topología MLP 1

Valores de la topología MLP 1	
No. De nodos en la capa de entrada	7
No. De capas ocultas	1
No. De nodos en la capa oculta	2
No. De nodos en la capa de salida	1
Error máximo permitido	0
No. Máximo de iteraciones	100

Figura 30. Topología MLP 1.



Después de un entrenamiento de 100 épocas del MLP descrito en el ANEXO D (Programa de entrenamiento de redes neuronales), se obtuvieron los siguientes resultados con respecto al error máximo que se había determinado como cero en

los parámetros de la red pero que llegó a un valor de 0.0041374, a pesar de que no llegó a un error de cero, en este caso el error es inferior que al del entrenamiento del perceptrón que logro un error de 0.258685 lo que se mostro reflejado en el desempeño de la red al momento de hacer la simulación e la misma. Para facilitar el entendimiento del desempeño de la topología MLP 1 se mostrara los resultados de manera grafica en la figura 31 (error) y 32 (simulación de la red).

Figura 31. Error del MLP 1.

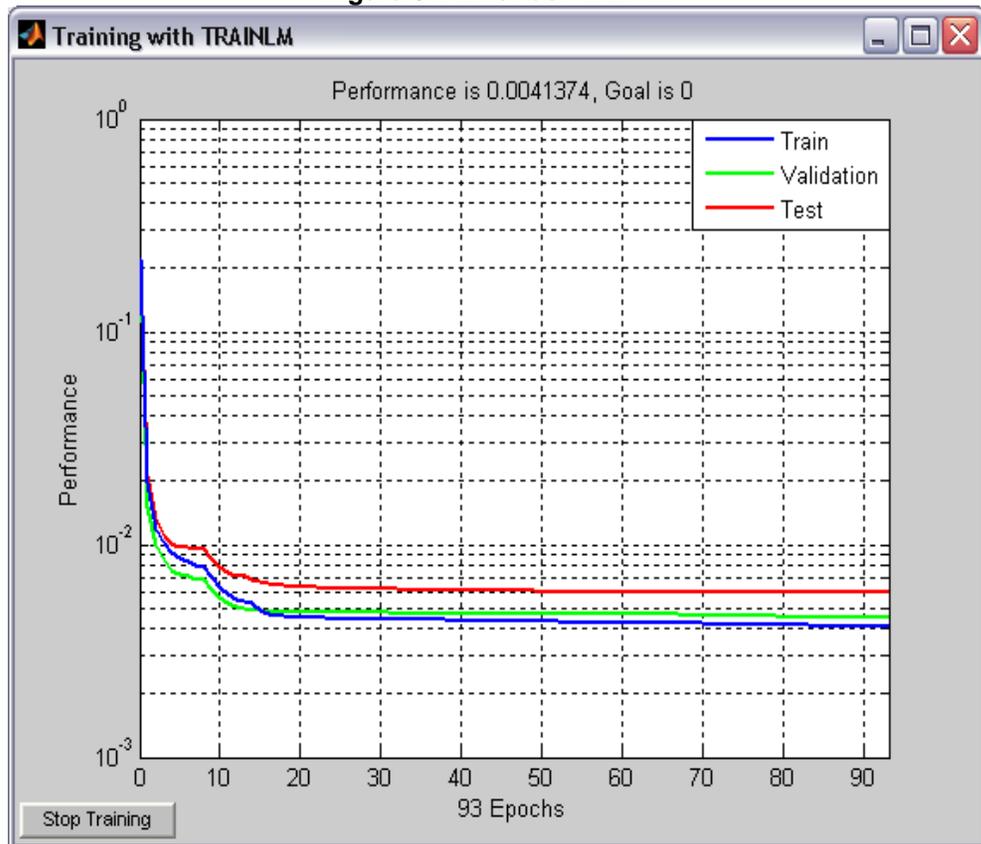
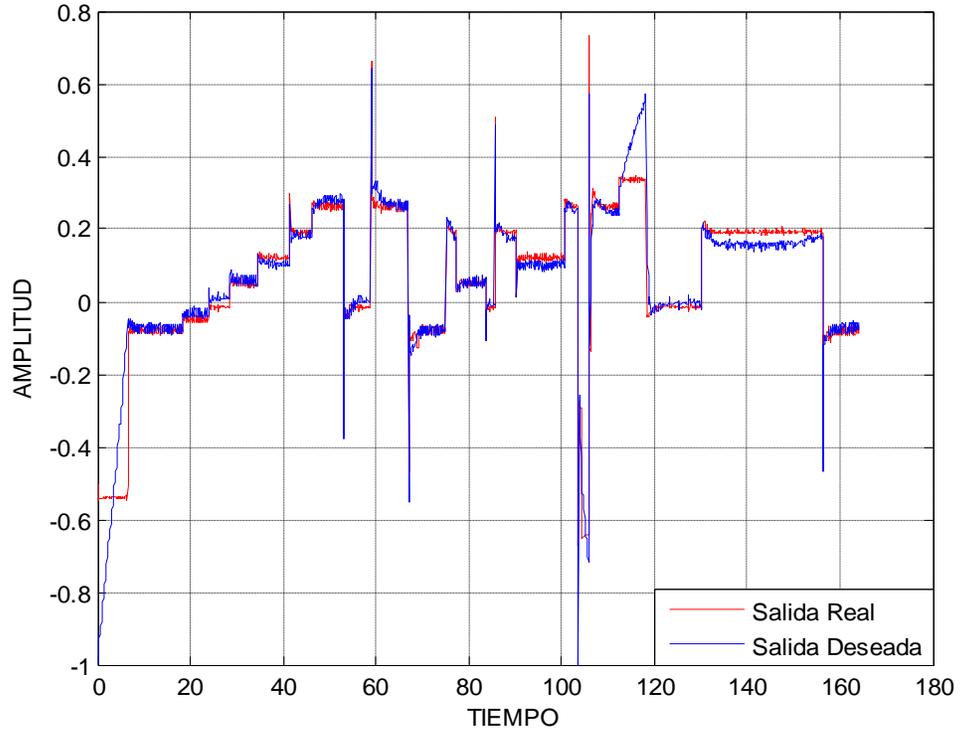


Figura 32. Simulación del MLP 1.

SALIDA REAL vs. SALIDA DESEADA



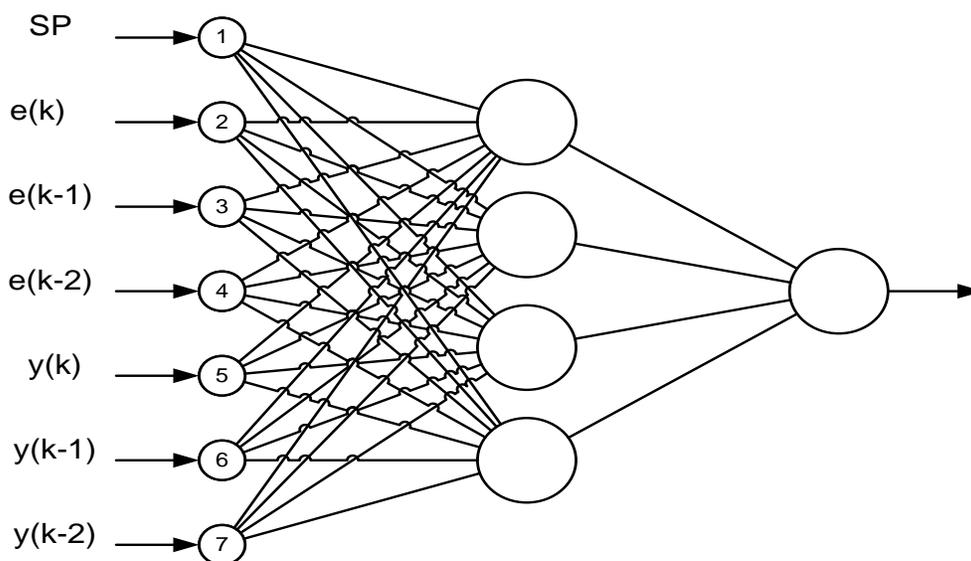
En la figura 32 se puede observar que hay intervalos en donde la red se comporta casi igual que la salida deseada, pero también hay otros intervalos donde la red tiene grandes diferencias con los valores deseados, por lo que se procede a iniciar un entrenamiento con más neuronas en la capa oculta para ver si mejora los resultados de esta red.

Para un segundo caso se entrena un MLP con las siguientes características:

Tabla 12. Valores de la topología MLP 2.

Valores de la topología MLP 2	
No. De nodos en la capa de entrada	7
No. De capas ocultas	1
No. De nodos en la capa oculta	4
No. De nodos en la capa de salida	1
Error máximo permitido	0
No. Máximo de iteraciones	100

Figura 33. Topología MLP 2.



En este caso es esta topología de red el error es de 0.00314246, lo que quiere decir que con respecto a la red anterior esta es más eficiente ya que su error es menor.

Figura 34. Error del MLP 2.

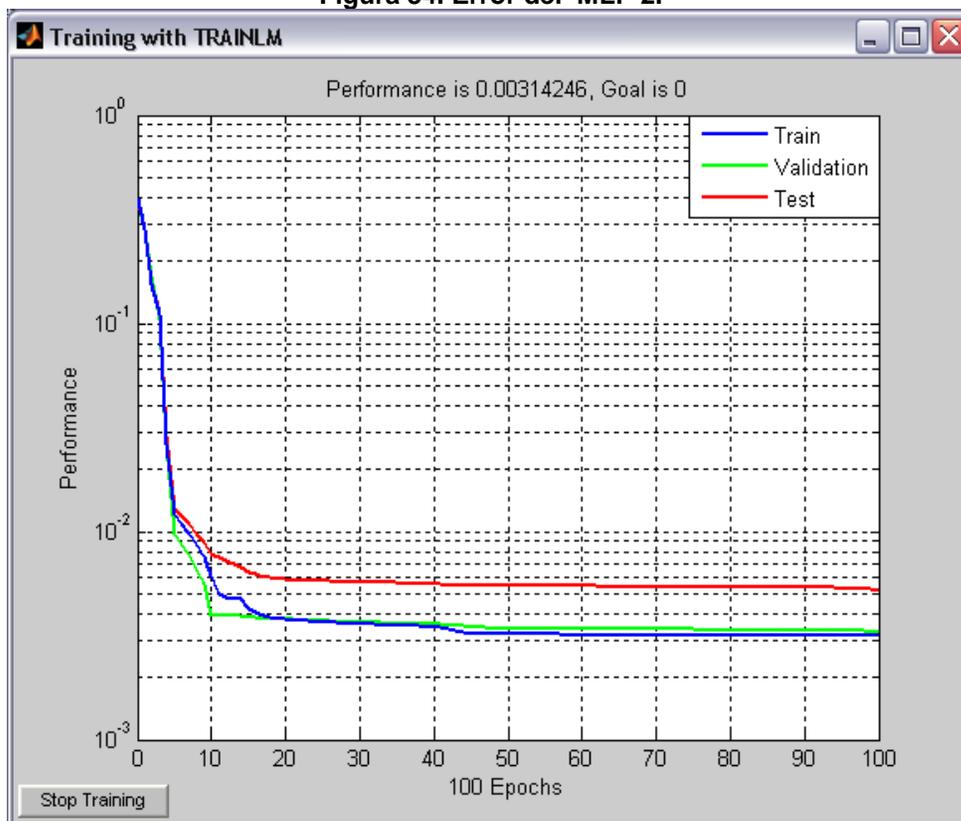
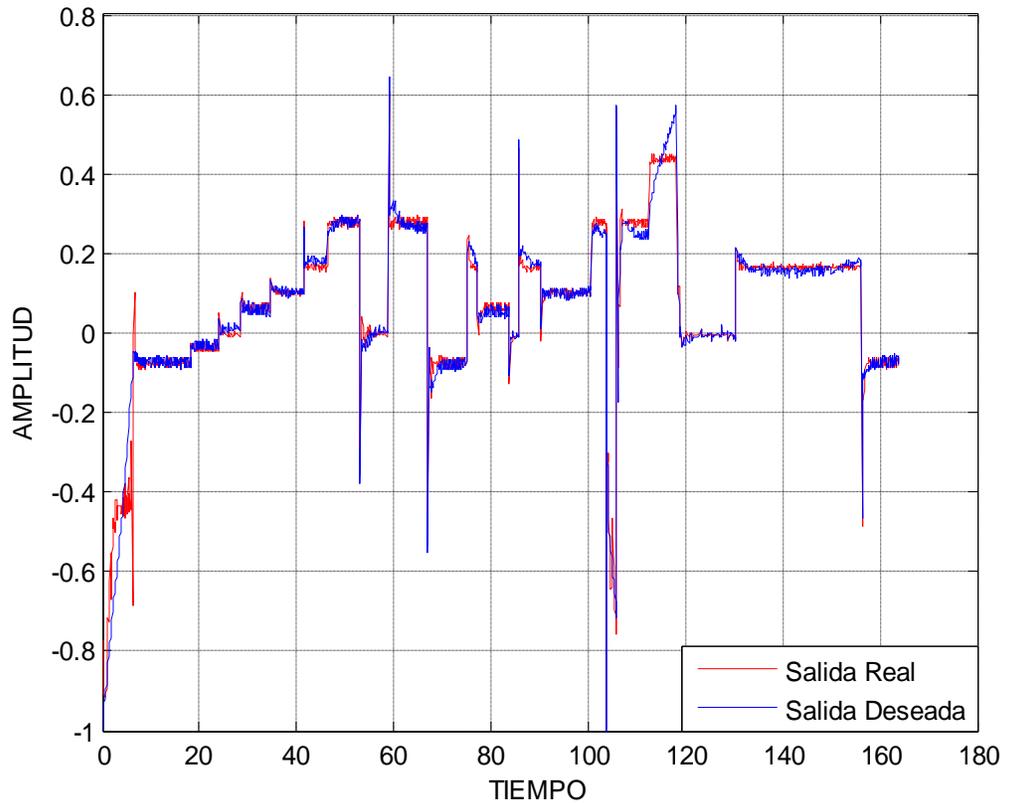


Figura 35. Simulación del MLP 2.
SALIDA REAL vs. SALIDA DESEADA

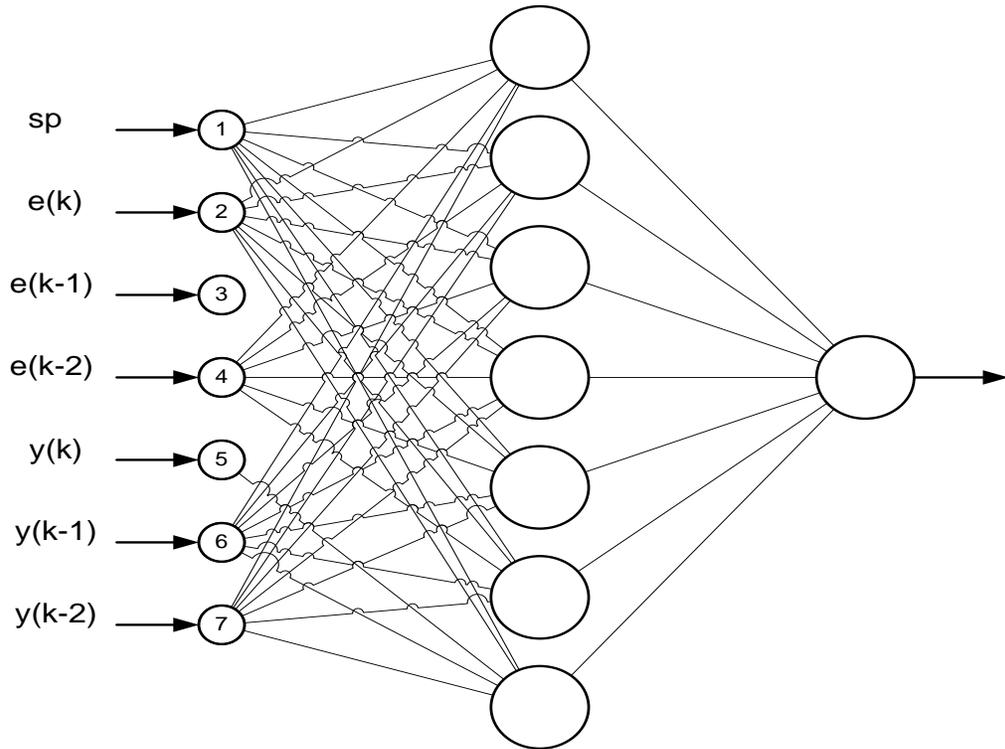


En el tercer caso se entrena un MLP con las siguientes características:

Tabla 13. Valores de la topología MLP 3.

Valores de la topología MLP 3	
No. De nodos en la capa de entrada	7
No. De capas ocultas	1
No. De nodos en la capa oculta	7
No. De nodos en la capa de salida	1
Error máximo permitido	0
No. Máximo de iteraciones	100

Figura 36. Topología MLP 3.



Para este caso se opto por un número de neuronas en la capa oculta igual al número de entradas, con esta topología se lograron buenos resultados, e incluso el tiempo de entrenamiento fue inferior a las 100 épocas colocadas como número máximo de iteraciones, el número de iteraciones fue de 52 épocas ya que el error de validación se aproximaron en gran parte al de entrenamiento por esto el Matlab termina el proceso de entrenamiento automáticamente, logrando un error de 0.00160218, con este error las redes MLP anteriores quedan prácticamente descartadas aunque su rendimiento fue, además a las otras redes se les venía guardado en archivos .txt los valores de los peso sinápticos y los valores de los bias para la posterior implementación, pero sus resultados no ameritaban mostrarlos, en este caso se mostraran los valores correspondientes a esto pesos

W1=net.IW {1,1}

Tabla 14. Pesos sinápticos de las entrada.

	1	2	3	4	5	5	7
N1	2,5537	5,7022	0,0930	1,7012	-0,8530	0,0317	-0,3860
N2	-1,5354	-0,4244	-0,3797	0,8702	0,1420	1,0178	0,0007
N3	5,7949	1,2265	0,6415	0,2721	3,0874	0,7334	2,3111
N4	4,7738	0,4534	0,6464	0,1519	1,7428	0,4587	1,4957
N5	-2,5645	1,0754	-4,3830	-4,0529	-1,3334	-0,3115	1,3899

N6	0,7694	-0,8115	-0,3316	0,1733	0,7048	-2,0578	0,3968
N7	2,6992	-0,2433	1,4429	-0,0280	1,5795	2,2384	1,9362

W2=net.LW {2,1}

Tabla 15. Pesos sinápticos de capa oculta.

	1	2	3	4	5	6	7
N1	3,8248	-1,6295	-1,8394	4,6270	4,2976	0,4861	-2,7329

b1=net.b {1}

b2=net.b {2}

Tabla 16. Bías de capa oculta y capa de salida.

	1
N1	4,5806
N2	2,2652
N3	1,7274
N4	1,9219
N5	-4,5317
N6	0,0044
N7	3,1783

	1
N1	2.0773

Figura 37. Error del MLP 3.

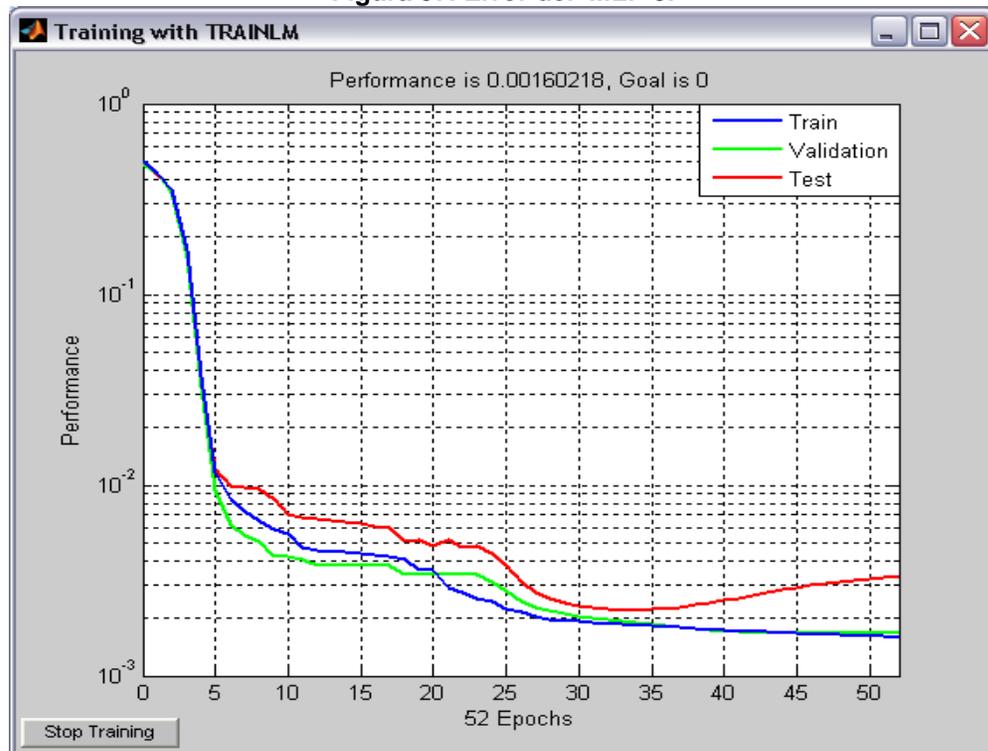
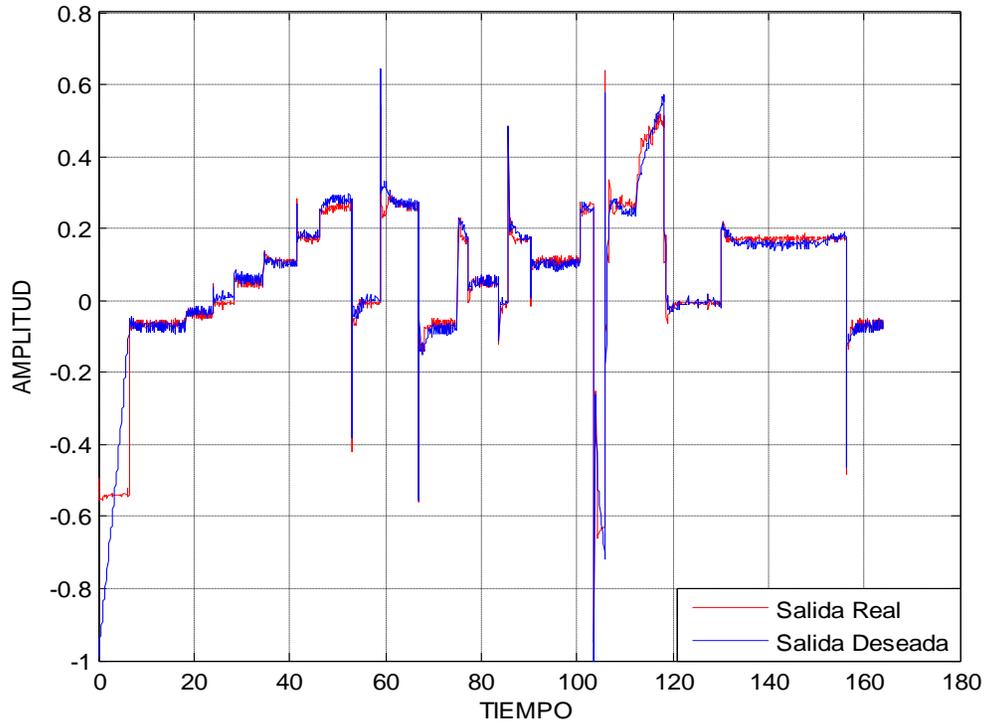


Figura 38. Simulación del MLP 3.

SALIDA REAL vs. SALIDA DESEADA

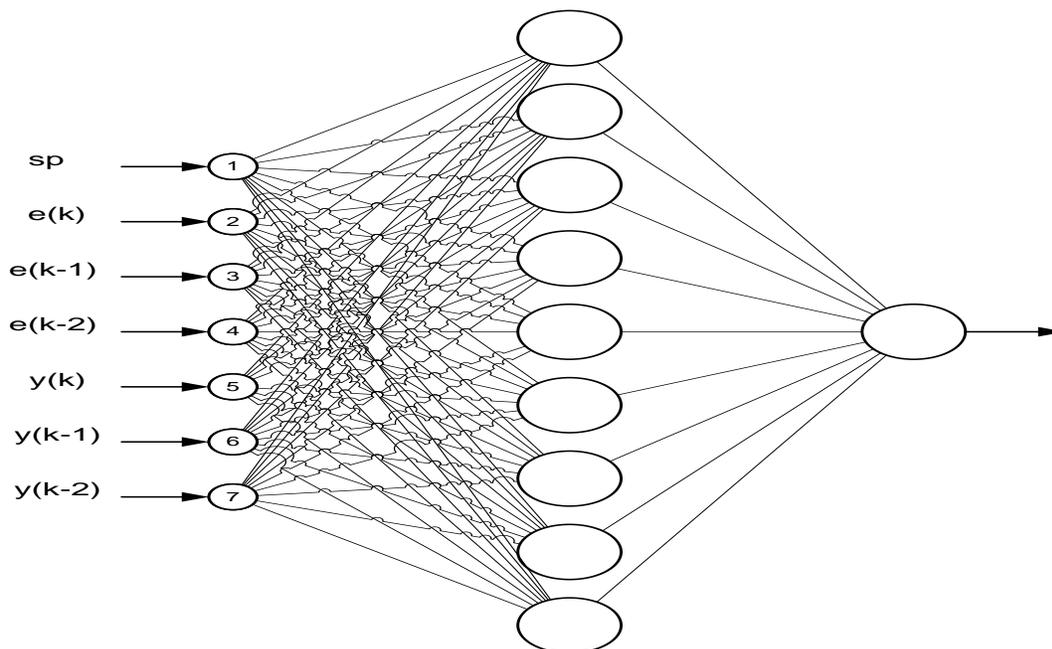


Para dar por terminada el entrenamiento de las redes MLP, se decidió entrenar un último modelo con las siguientes características:

Tabla 17. Valores de la topología MLP 4.

Valores de la topología MLP 4	
No. De nodos en la capa de entrada	7
No. De capas ocultas	1
No. De nodos en la capa oculta	9
No. De nodos en la capa de salida	1
Error máximo permitido	0
No. Máximo de iteraciones	100

Figura 39. Topología MLP 4.



La tabla 19 muestra los pesos sinápticos de la entrada de la red

Tabla 18. Pesos sinápticos de las entrada.

	1	2	3	4	5	6	7
N1	-1,8386	-3,6256	-1,7085	-2,1379	-1,1739	-2,1077	-2,5431
N2	-3,4976	-2,4015	-0,4557	1,4727	1,4145	-0,3319	-3,2823
N3	5,8620	5,8219	5,9919	2,6421	-0,6989	7,7862	4,9284
N4	-0,5154	0,2040	0,0033	0,1386	-0,1283	0,7197	-0,0524
N5	-0,0513	0,7389	0,7140	0,5038	-0,4653	-1,3801	1,1865
N6	-1,6971	-1,2748	-1,5562	3,9140	-1,5814	8,7461	-0,1597
N7	-1,3648	-0,0580	-4,5750	-4,4865	1,0713	1,7255	0,1217
N8	-9,5631	-1,0755	-4,2840	-5,5982	-5,4768	-0,1211	0,6023
N9	0,5095	-3,9370	-3,1090	-0,9637	1,7065	-1,0480	-1,3514

La tabla 20 muestra los pesos sinápticos de las neuronas de la capa oculta.

Tabla 19. Pesos sinápticos de capa oculta.

	1	2	3	4	5	6	7
N1	-0,571	-0,027	4,992	-1,564	2,661	-2,300	-0,768

La tabla 20 muestra los bias de la neuronas de la capa oculta y de la capas de salida.

Tabla 20. Bias de capa oculta y capa de salida.

	1		1
N1	-3,9837	N1	-4.217
N2	3,4710		
N3	7,1774		
N4	0,2476		
N5	-4,6903		
N6	-4,0483		
N7	10,6191		
N8	-7,4860		
N9	-9,5469		

Figura 40. Error del MLP 4.

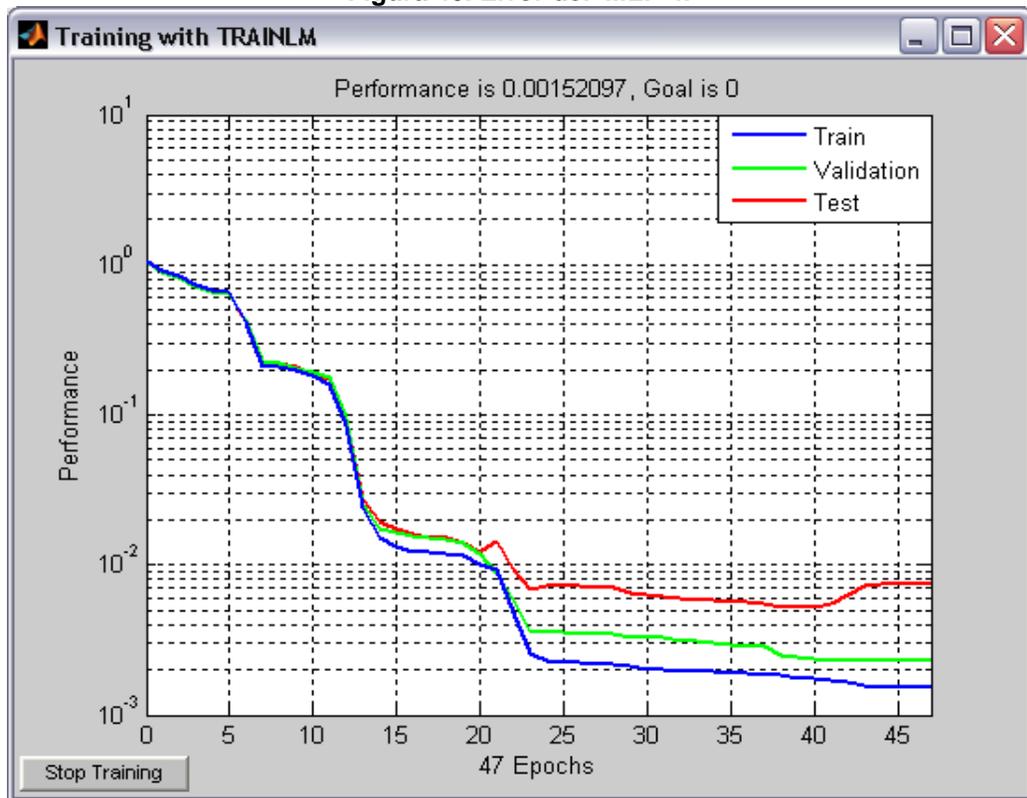
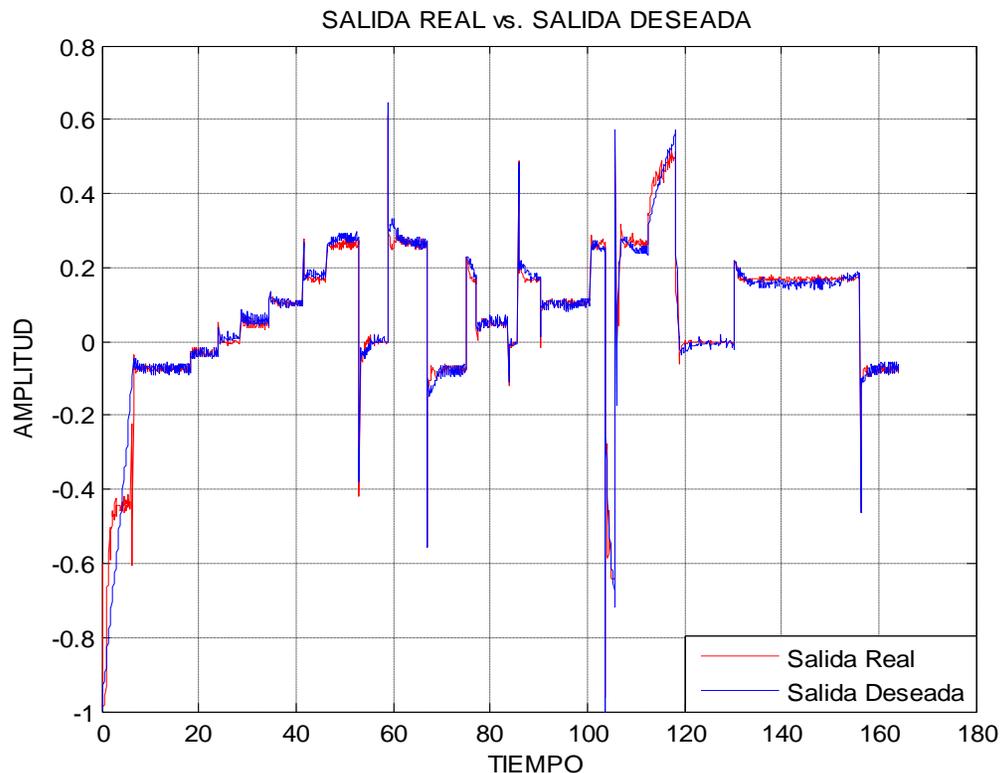


Figura 41. Simulación del MLP 4.



3.3.3 Entrenamiento del RBF. Para este tipo de red el entrenamiento es mucho más sencillo y rápido, en la creación de la red y entrenamiento también se hace uso de las funciones del Neural Network Toolbox, que tiene una función para redes de funciones de base radial que se llama `newrb`, tiene como los siguientes parámetros y estructura.

`newrb(P,T,goal,spread,MN,DF)`

Donde:

P: matriz $R \times Q$ de Q vectores de entrada.

T: matriz $S \times Q$ de Q vectores objeto.

goal: error cuadrático medio (default = 0)

spread: propagación de la función de base radial (default = 1.0)

MN: máximo número de neuronas (default es Q)

DF: en proceso de entrenamiento se muestra cada vez que ocurra este número de épocas (default = 25)

Algo que pasa con la función `newrb` con el parámetro MN que es el número máximo de neuronas, por defecto también es el número máximo de épocas, o sea

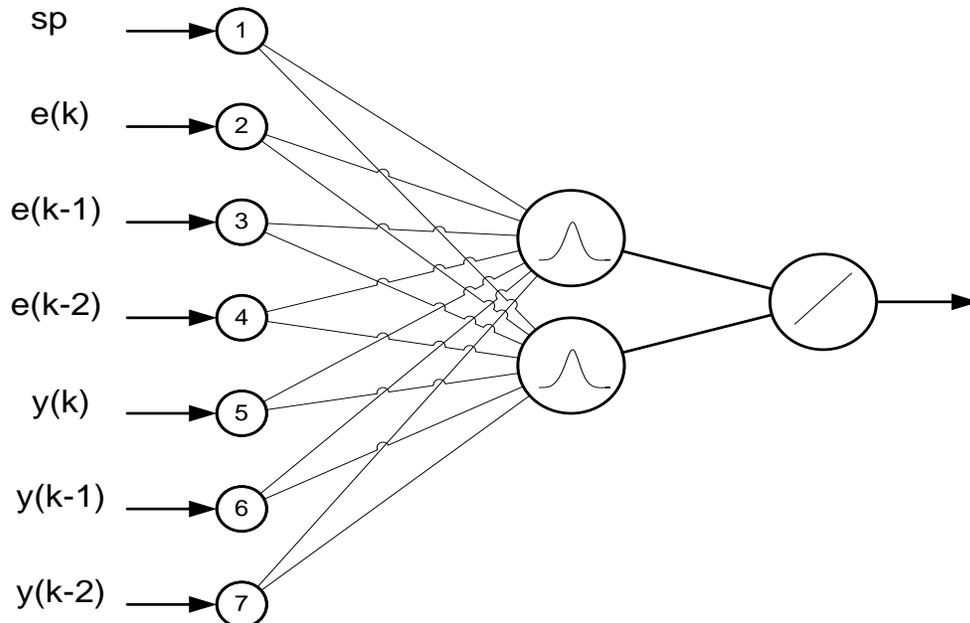
que si se tienen cuatro neuronas en la capa oculta el proceso de entrenamiento dura 4 épocas.

Para iniciar con el proceso de entrenamiento de la red utilizaremos una res con las siguientes características:

Tabla 21. Valores de la topología RBF 1.
Valores de la topología RBF 1

No. De nodos en la capa de entrada	7
No. De capas ocultas	1
No. De nodos en la capa oculta	2
No. De nodos en la capa de salida	1
Error máximo permitido	0
No. Máximo de iteraciones	2

Figura 42. Topología RBF 1.



Los pesos de la red son los siguientes

$$W1 = \text{net.IW}\{1,1\};$$

Tabla 22. Pesos sinápticos de entrada.

	1	2	3	4	5	6	7
N1	1,000	0,930	0,938	0,938	0,035	0,031	0,031
N2	-0,800	-0,578	-0,578	-0,576	-0,111	-0,111	-0,112

W2=net.LW{2,1}

Tabla 23. Peso sinápticos de capa oculta.

	1	2
N1	0,1887	-0,4365

b1=net.b{1}

b2=net.b{2}

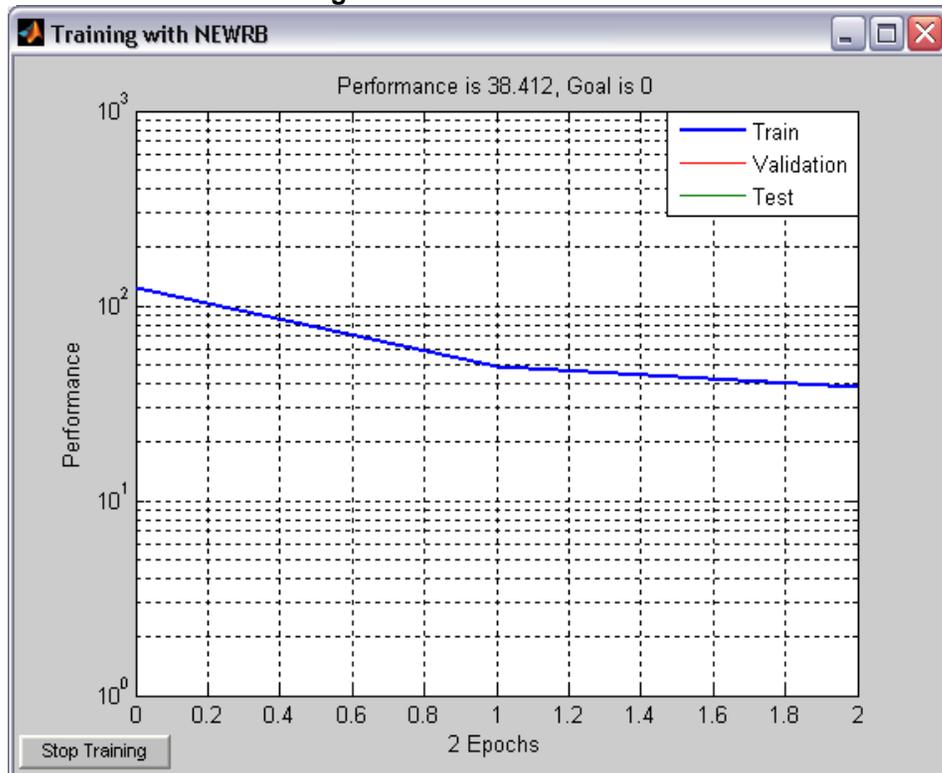
Tabla 24. Bias del capa oculta y capa de salida.

	1
N1	0.8326
N2	0.8326

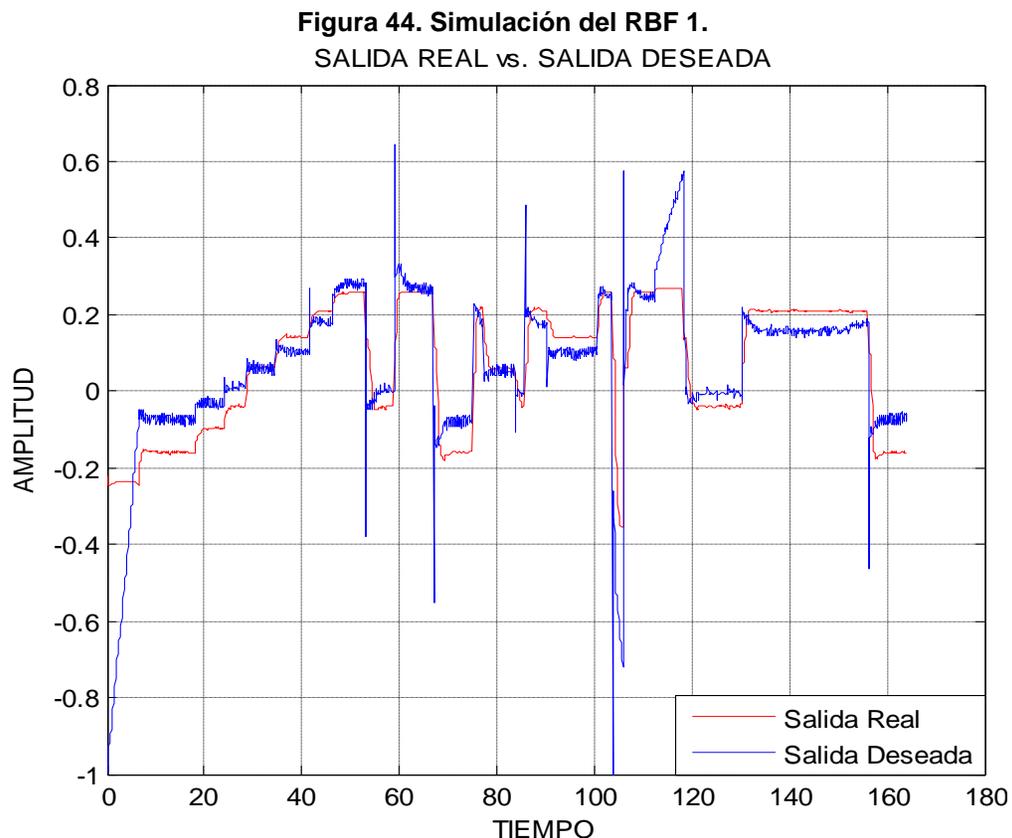
	1
N1	0.0822

Grafica del erro a lo largo de las dos épocas de entrenamiento (figura 43)

Figura 43. Error del RBF 1.



Grafica de la salida de la red ante el conjunto de datos de entrada (figura 44)

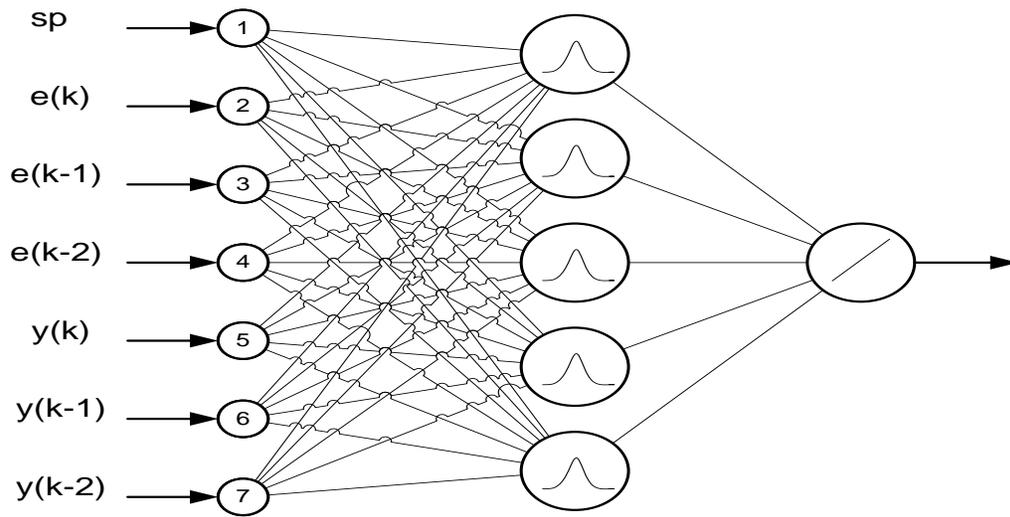


La segunda topología de red RBF tiene las siguientes características:

Tabla 25. Valores de la topología RBF 2.

Valores de la topología RBF 2	
No. De nodos en la capa de entrada	7
No. De capas ocultas	1
No. De nodos en la capa oculta	5
No. De nodos en la capa de salida	1
Error máximo permitido	0
No. Máximo de iteraciones	5

Figura 45. Topología RBF 2.



Valor de pesos sináptico de la red

$W1 = \text{net.IW}\{1,1\};$

Tabla 26. Pesos sinápticos de las entrada.

	1	2	3	4	5	5	7
N1	1,000	0,934	0,930	0,938	0,033	0,035	0,031
N2	-0,800	-0,578	-0,578	-0,576	-0,111	-0,111	-0,112
N3	-0,300	-0,426	-0,406	-0,383	0,063	0,053	0,041
N4	-0,200	0,805	0,787	0,797	-0,502	0,006	0,002
N5	-0,800	-0,557	-0,551	-0,545	-0,122	-0,125	-0,128

Tabla 27. Pesos sinápticos de la capa oculta.

	1	2	3	4	5
N1	0,465	-6,790	0,517	-0,536	5,502

$b1 = \text{net.b}\{1\}$

$b2 = \text{net.b}\{2\}$

Tabla 28. Bias del capa oculta y capa de salida.

	1
N1	0,833
N2	0,833
N3	0,833
N4	0,833
N5	0,833

	1
N1	0.0772

Error de la red RBF 2 (figura 46)

Figura 46. Error del RBF 2.

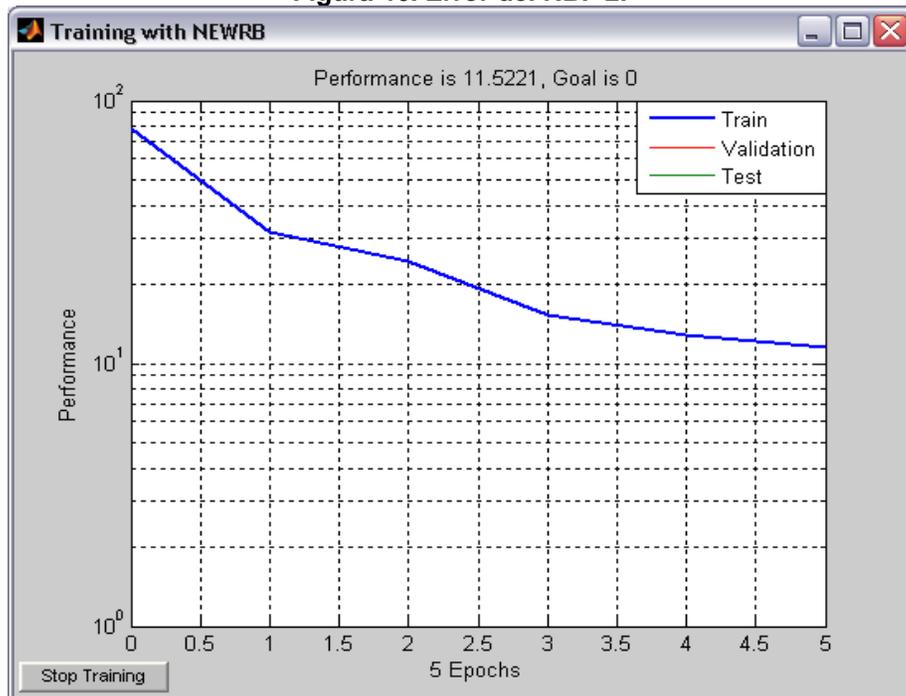
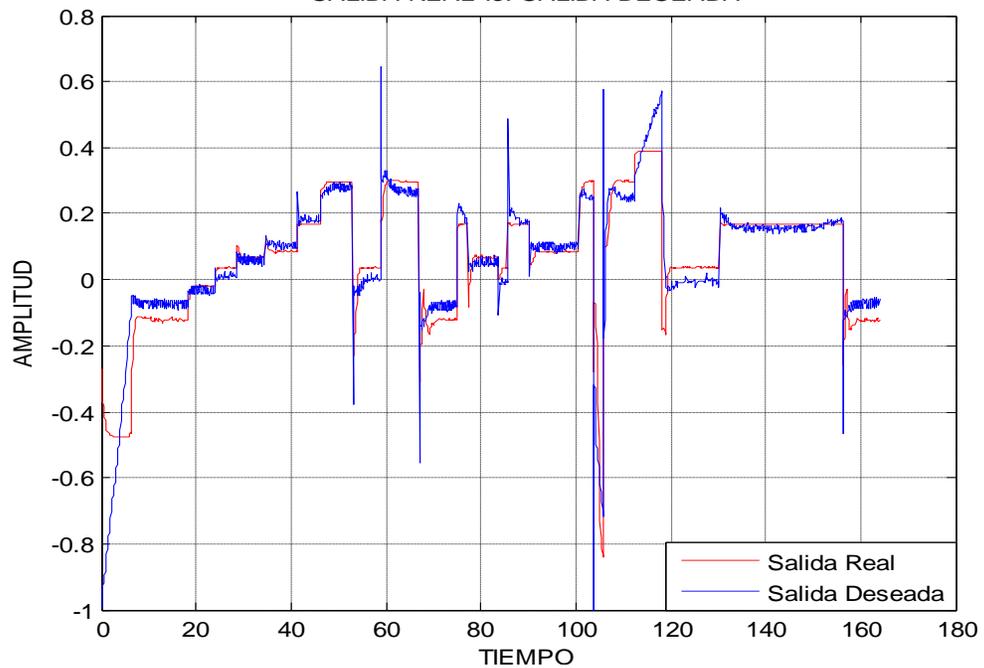


Figura 47. Simulación del RBF 2.

SALIDA REAL vs. SALIDA DESEADA

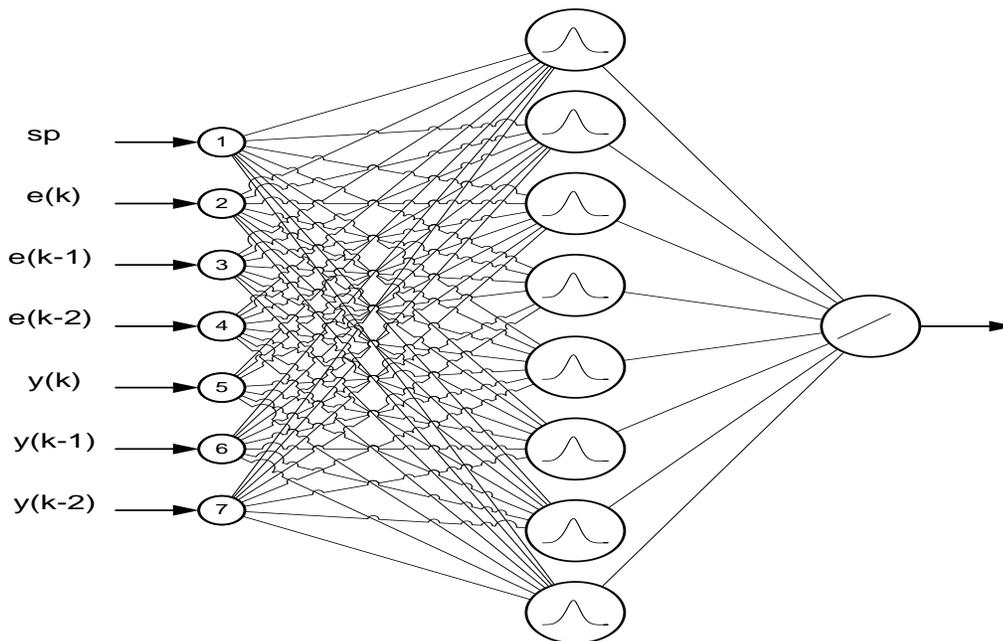


Para la tercera red RBF las características son las siguientes:

Tabla 29. Valores de la topología RBF 3.

Valores de la topología RBF 3	
No. De nodos en la capa de entrada	7
No. De capas ocultas	1
No. De nodos en la capa oculta	8
No. De nodos en la capa de salida	1
Error máximo permitido	0
No. Máximo de iteraciones	8

Figura 48. Topología RBF 3.



Pesos sinápticos de la red RBF 3

Tabla 30. Pesos sinápticos de las entradas.

	1	2	3	4	5	5	7
N1	1,000	0,934	0,930	0,938	0,033	0,035	0,031
N2	-0,800	-0,578	-0,578	-0,576	-0,111	-0,111	-0,112
N3	-0,300	-0,426	-0,406	-0,383	0,063	0,053	0,041
N4	-0,200	0,805	0,787	0,797	-0,502	0,006	0,002
N5	-0,800	-0,557	-0,551	-0,545	-0,122	-0,125	-0,128
N6	0,000	0,680	0,725	0,779	-0,340	-0,362	-0,390
N7	-0,800	-0,061	0,008	0,090	-0,370	-0,404	-0,445

N8 | 0,800 -0,588 -0,578 -0,578 0,694 -0,111 -0,111

La tabla 31 muestra los pesos sinápticos de las neuronas de la capa oculta.

Tabla 31. Pesos sinápticos de capa oculta.

	1	2	3	4	5	6	7	8
N1	0,398	-17,255	0,373	-1,536	16,382	1,091	-0,775	0,345

La tabla 32 muestra los bias de la neuronas de la capa oculta y de la capas de salida.

Tabla 32. Bias de capa oculta y capa de salida.

	1
N1	0,833
N2	0,833
N3	0,833
N4	0,833
N5	0,833
N6	0,833
N7	0,833
N8	0,833

	1
N1	0.0520

Figura 49. Error del RBF 3.

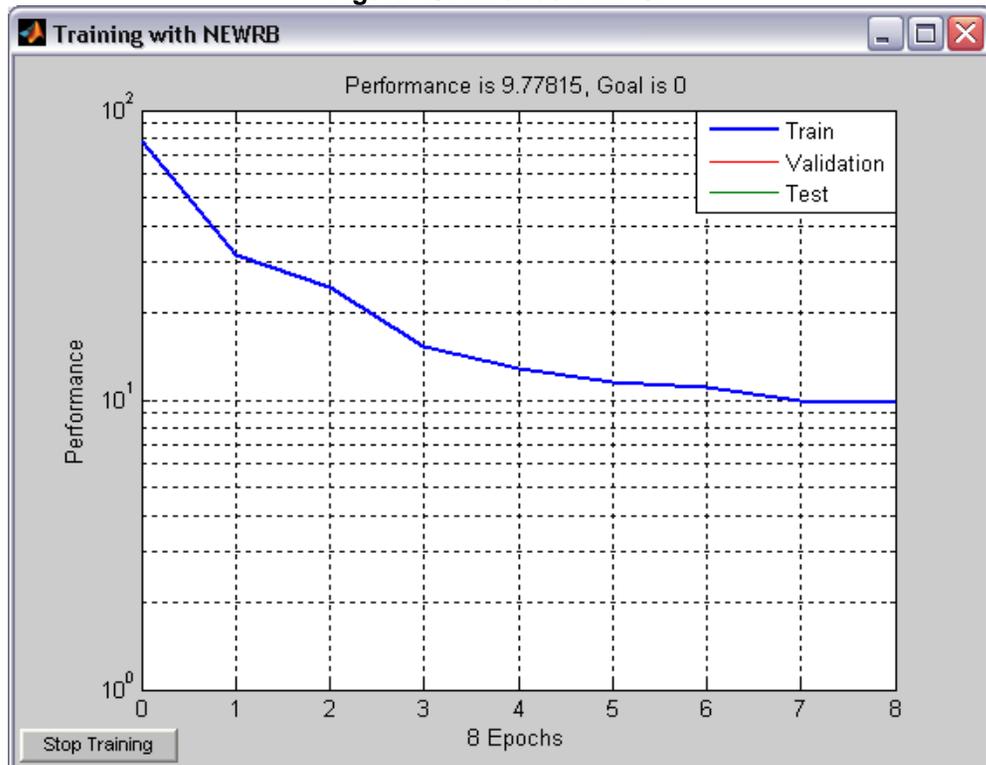
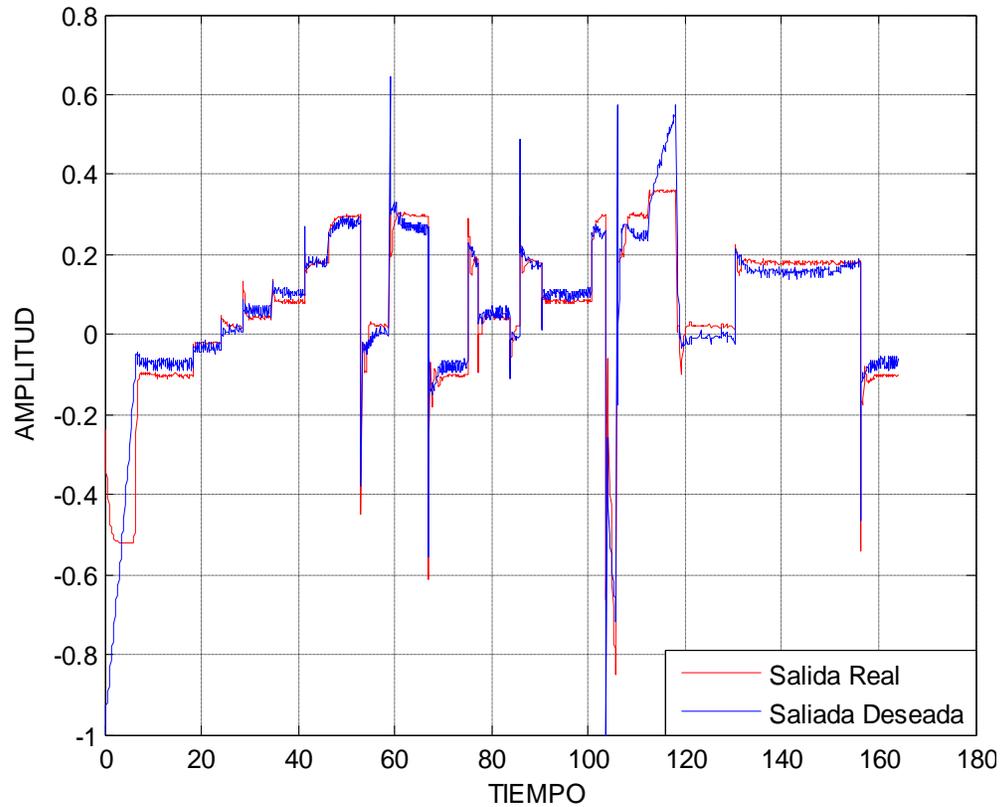


Figura 50. Simulación del RBF 3.
SALIDA REAL vs. SALIDA DESEADA



Habiendo terminado la etapa de entrenamiento de las redes, se procede a definir cuál fue la topología de red neuronal que mejor comportamiento con relación a las demás, teniendo en cuenta toda la información recopilada de cada uno de los modelos.

Teniendo en cuenta la interpretación de la información los modelos que mejor comportamiento tuvieron con respecto a los demás fueron el MLP 4, el RBF 2 y RBF 3, siendo el más destacado de estos tres el MLP 4, por lo que es el modelo al que se la hará el desarrollo para controlara el sistemas de flujo de agua, con los valores de los pesos y los bías que se habían guardado con anterioridad para el posterior desarrollo del controlador.

4. DESARROLLO DEL MODELO DE CONTROL BASADO EN REDES NEURONALES ARTIFICIALES.

Después de tener definida la topología y modelo de RNA (Red Neuronal Artificial), se procede con el desarrollo del programa para el control, para el cual se utiliza LabView; que es una herramienta de lenguaje gráfico, con la cual se puede hacer el procesamiento de los datos, graficarlos y hacer uso de sus interfaces de comunicación como lo es la interfaz del puerto serie, que es la que se utiliza en este caso para comunicar el proceso a través de la tarjeta de adquisición de datos con el PC y viceversa.

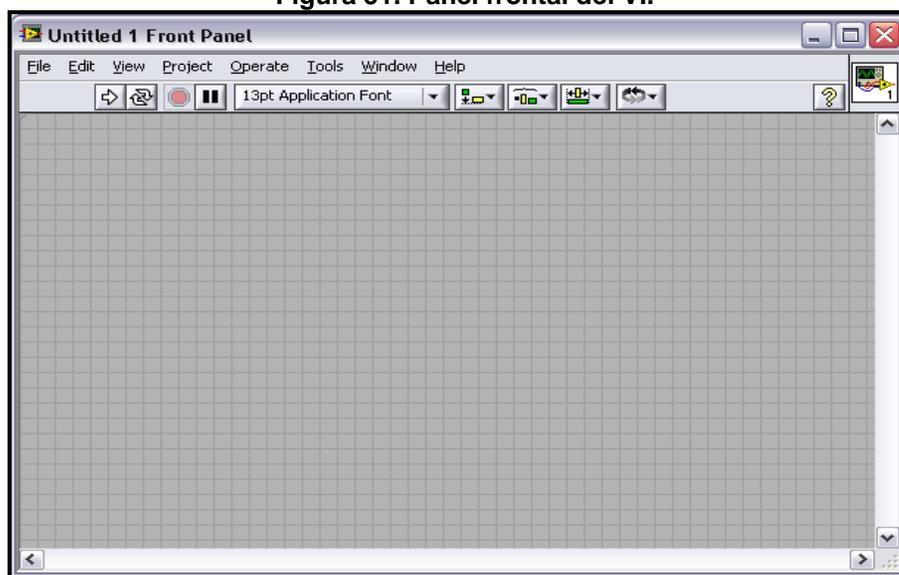
4.1 DESARROLLO DEL MODELO DE RNA'S EN LABVIEW.

LabView, como se mencionaba anteriormente es un software de lenguaje grafico que permite la adquisición y transmisión de datos por cualquier de sus interfaces de comunicación como lo son el puerto serie, puerto paralelo, USB, TCP/IP, UDP, entre otros, además, permite la visualización y el desarrollo del algoritmos de control, con las herramientas que están a disposición del programador.

Los programas desarrollado en LabView son denominados VI's (Instrumentos Virtuales), estos VI's están contienen tres componentes principales, que son el panel frontal, el diagrama de bloques y el panel icono/conector.

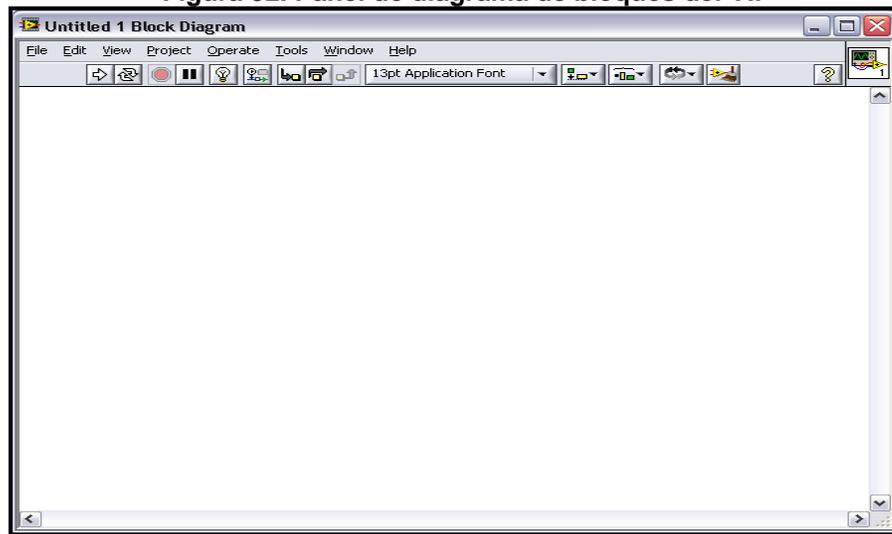
El panel frontal es la interfaz del usuario para el VI, donde el programador construye controles e indicadores, los cuales son terminales interactivas de entrada y de salida del VI, respectivamente. La figura 51 muestra el panel frontal de un VI en LabView.

Figura 51. Panel frontal del VI.



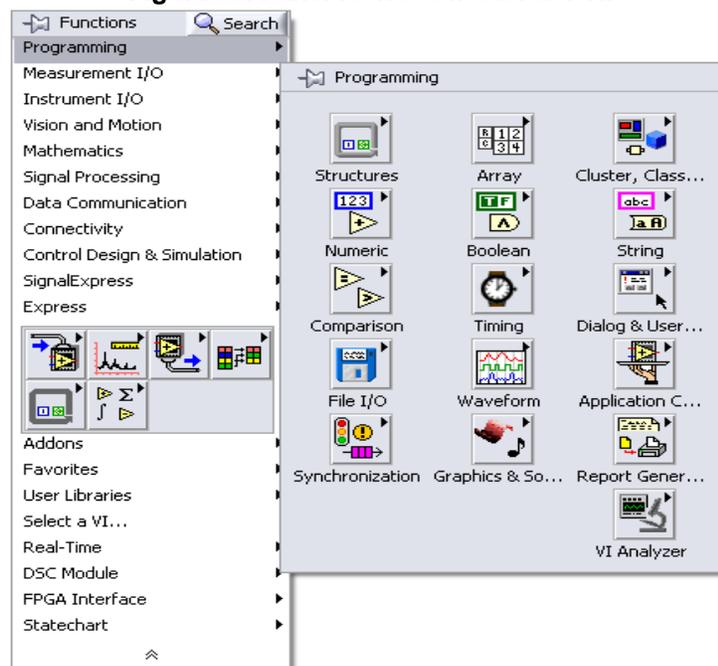
El panel de diagrama de bloques es donde se adiciona código agregando representaciones graficas de funciones para controlar los objetos del panel frontal. La figura 52 muestra el panel de diagramas de bloques.

Figura 52. Panel de diagrama de bloques del VI.



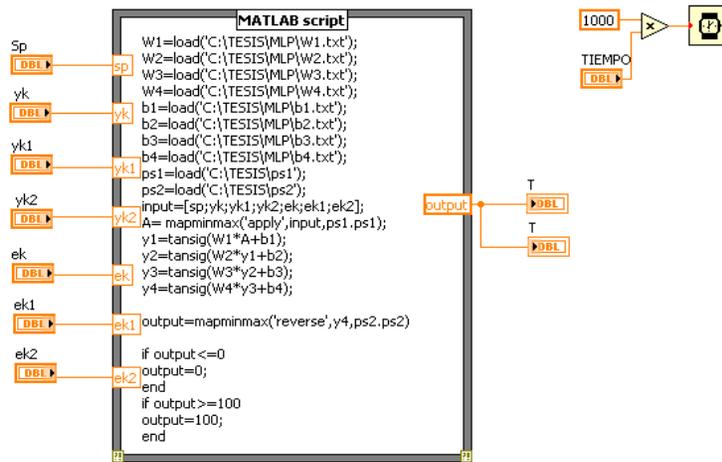
El panel icono/conector es donde encuentran toda la funciones para crear un programa en LabView, simulando funciones de lenguajes de programación basados en texto, pero en este caso no son líneas de código sino diagramas de bloques, como lo muestra la figura 53.

Figura 53. Panel icono/conector del VI.



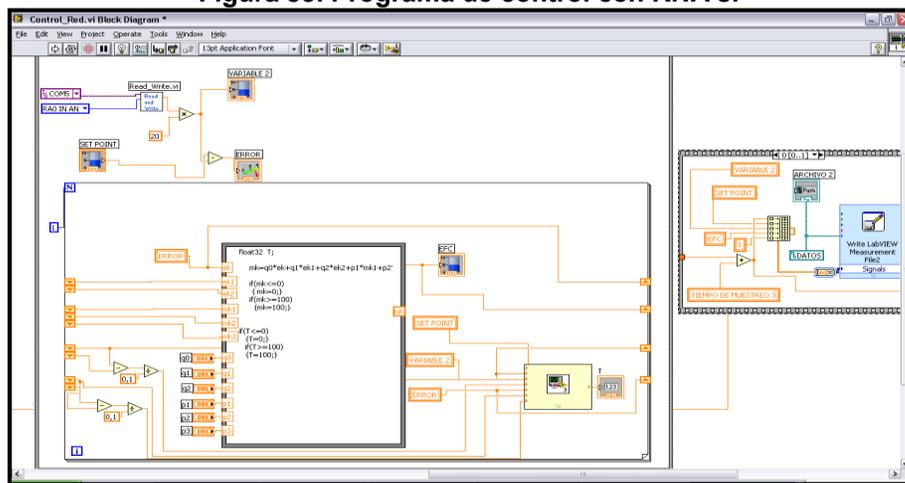
Para hacer el desarrollo de la RNA en LabView fue necesario utilizar una función de comunicación entre este software y Matlab, ya que la red neuronal fue creada y entrenada en Matlab; este bloque permite utilizar funciones propias de Matlab, permite cargar datos, tener variables de entradas provenientes de LabView y variables de salidas hacia LabView, como se muestra en la figura 54, donde los iconos de la izquierda son variables de entrada y los de la derecha son variables de salida del Matlab script.

Figura 54. Función de Matlab en LabView.



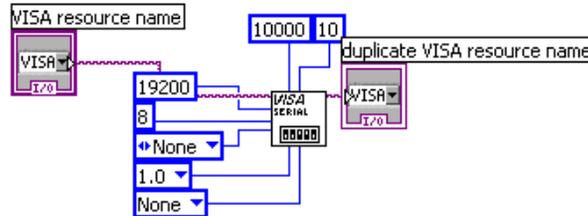
La figura 55, muestra el desarrollo el programa para el control basado en RNA para la implementación en el proceso de flujo de agua, en este programa se encuentra la comunicación entre LabView y el proceso, el procesamiento de los datos para hacer los registros de desplazamiento para obtener los valores anteriores de las variables que son las entradas de la red para que esta los procese y de la salida.

Figura 55. Programa de control con RNA's.



Para la comunicación con el proceso se debe configurar el puerto por donde se van a recibir y a transmitir los datos, en este caso el puerto de comunicación es el serie con el protocolo RS232, que tiene la siguiente configuración como lo muestra la figura 56, donde la velocidad de transmisión y recepción es 19200 Baudios, a 8 bits y sin bit de paridad.

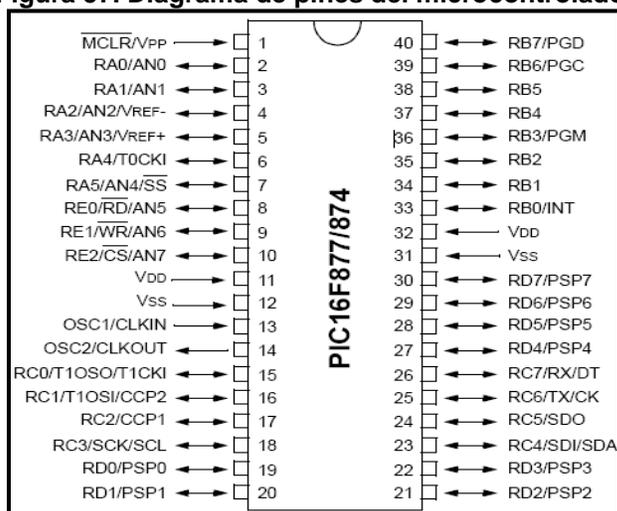
Figura 56. Configuración del puerto serial RS232.



Para la configuración del puesto serial se tuvo que tener en cuenta la configuración de la tarjeta de adquisición de datos, ya que el módulo de comunicación serial tanto de la tarjeta como de la interfaz tienen que estar a la misma velocidad de transmisión y recepción de los datos, al igual que el número de bits que también tiene que ser el mismo.

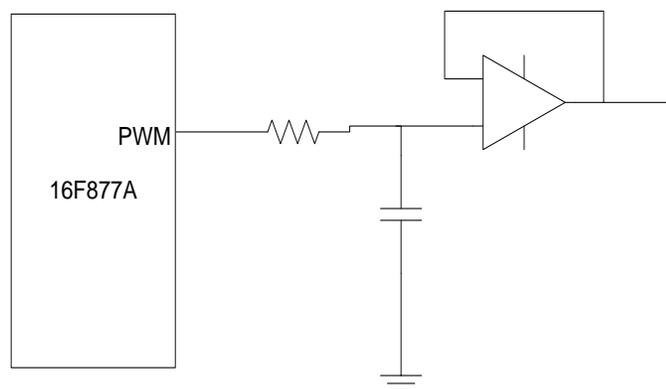
Para la tarjeta antes mencionada se utilizó el microcontrolador PIC16F877A de 40 pines como lo muestra la figura 57, con comunicación serial RS232, conversión analógica digital, salidas por PWM, entre otras características, que se utilizó como unidad de procesamiento, donde se lleva a cabo la conversión de la señal analógica proveniente del conversor corriente/voltaje a señal digital, esto se hace con una resolución de 10 bits en el PIC, para ser transmitida a la computadora por una interfaz de comunicación RS232, después de la información ser recibida es procesada en la red neuronal en el programa y luego entrega su salida a la tarjeta de adquisición de datos para efectuar una acción sobre el módulo de flujo, esta información se transmite por la misma interfaz de comunicación RS232, que es recibida por el microcontrolador, que la convierte de digital a analógica por medio de un filtro, ya que el elemento final del módulo (la válvula) no puede recibir el tipo de señal que entrega el microcontrolador, que es una señal del tipo PWM.

Figura 57. Diagrama de pines del microcontrolador.



Para el diseño del filtro antes mencionado, se tuvo que tener en cuenta la frecuencia de salida del PWM, esto para que coincidiera la conversión, ya que si el PWM estaba al 50% de ciclo de dureza la salida en voltaje del filtro tenía que ser 2.5 voltios y al 100% la salida tenía que ser 5 voltios, además del filtro RC se le agregó un seguidor o Buffer con un amplificador operacional para hacer el desacople entre el filtro y la válvula para protección de los circuitos.

Figura 58. Circuito del filtro RC.



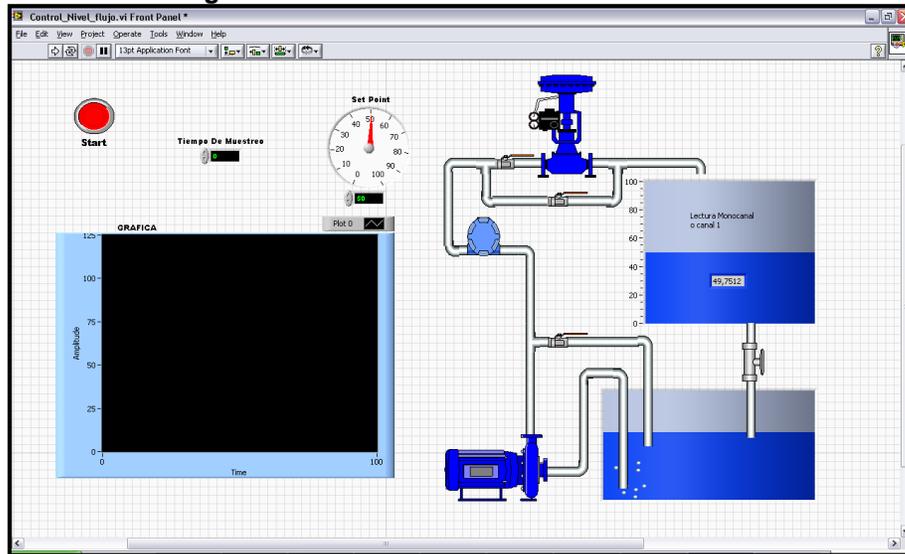
El filtro se calcula con la siguiente ecuación:

$$RC = \frac{1}{2\pi f} \quad \text{Ec. 32}$$

Se diseñó con una resistencia de 1k y la frecuencia de 10khz, dando como resultado un condensador de 0.01uf.

La figura 59 muestra el panel frontal del programa de control basado en RNA's, en este se podrá visualizar la variable del proceso, la apertura del elemento final de control, y el valor de referencia deseado, entre otras cosas.

Figura 59. Panel frontal de control con RNA's.

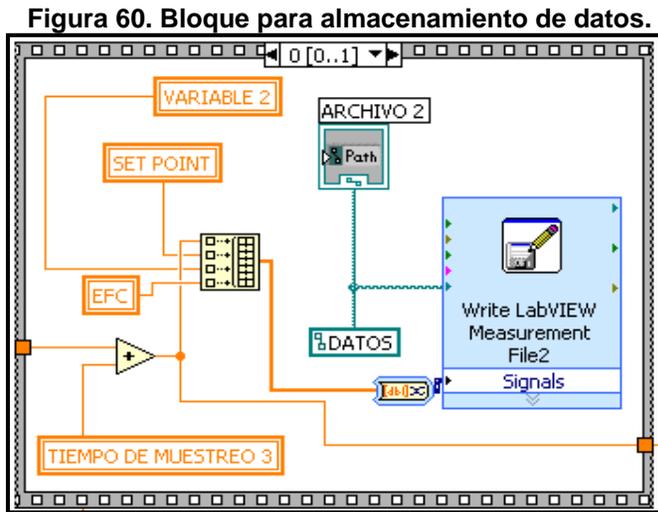


Para el proceso de implementación en el proceso se procede a hacer la conexiones necesarias entre el modulo y el PC, para esto se conecta el salida del transmisor de flujo a al convertidor de corriente a voltaje que estará en una de la entradas de la tarjeta de adquisición de datos, también, se conectara a la entrada del elemento final de control (Válvula) el convertidor de voltaje a corriente que estará en una de las salidas análogas de la tarjeta, la tarjeta estará comunicada a el PC a través del puerto serial.

Después de hacer todas las conexiones necesarias se pone a correr el programa, que inicialmente mide el estado de la variable a controlar, la compara con la referencia (set-point), calcula el error y da la salida necesaria para compensar el erro, de manera que este llegue a cero.

5. VALIDACIÓN DEL COMPORTAMIENTO DEL SISTEMAS DE CONTROL EN EL MÓDULO DE FLUJO-NIVEL.

En esta etapa se genera un archivo plano de los datos del proceso de control basado en RNA's ya implementado para determinar el comportamiento de la red, como lo muestra la figura 60, donde se guarda el tiempo de muestreo, el set point, el valor de la variable y la acción del elemento final de control, con estos datos se puede saber cuál es el error para compararlo con el control convencional.



Para hacer el proceso de validación se realizarán pruebas al sistema de control basado en RNA para el módulo de flujo de agua y se analizará el comportamiento de dicho control, el resultado de las pruebas se comparará con el control convencional, para esto como se decía antes, se genera un archivo con los datos de importancia del control convencional y el control basado en RNA's para su posterior comparación de resultados.

Para iniciar se proceda a aplicar pruebas al proceso con el controlador convencional PID por Ziegler-Nichols desarrollado e implementado en la primera etapa del proyecto para ver que tan eficiente es este tipo de control. Las pruebas a las que se someterá el controlador PID serán de la misma clase a la que se someterán al controlador basado en redes neuronales; serán una serie de escalones y cambios del set point para ver como el controlador trata de llegar a la dicha referencia, como lo muestran las figuras 61 (pruebas del proceso de control convencional) y la figura 62 (pruebas al proceso de control basado en RNA's).

Figura 61. Pruebas al control convencional.

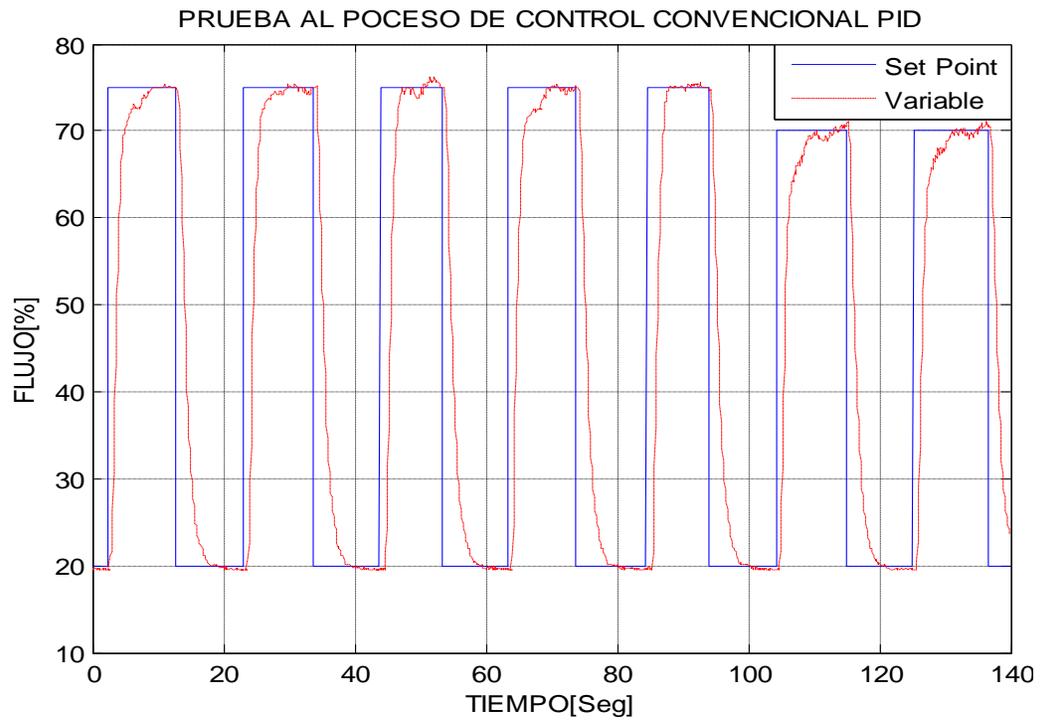
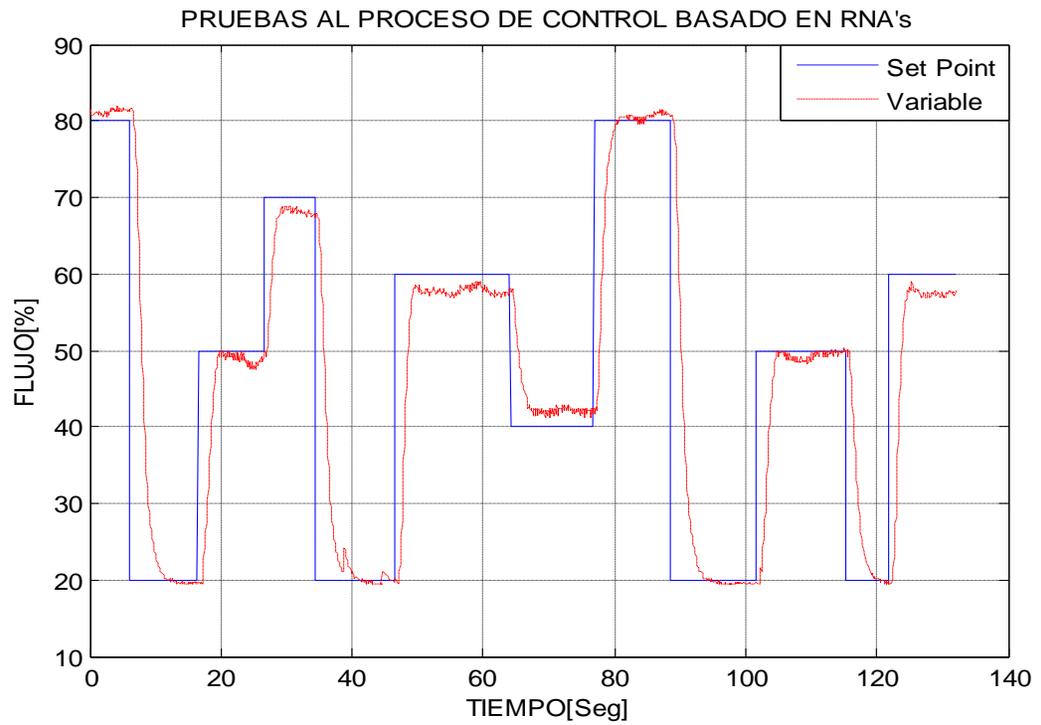


Figura 62. Pruebas al control basado en RNA's



De la información recolectada se puede determinar que el controlador con menor error a los largo de la prueba fue el convencional, manteniendo la variable controlada lo más cercana posible a al el set point para que el error fuera mínimo.

En el CD, se anexan los archivos de la base de datos de identificación y la del proceso de control.

6. CONCLUSIONES

- En el momento de implementar el algoritmo de control convencional PID (Proporcional Integral Derivativo) en el modulo de flujo de agua, es necesario recopilar un volumen de información suficiente dentro del rango de funcionamiento del proceso para tener un entrenamiento optimo de la red neuronal ya que en ocasiones al no tener a información adecuada en ciertos rangos de funcionamiento la red puede presenta problemas y no se obtendrá la salida deseada, y la red podría trabajar bien en los rangos donde si se recolecto la información necesaria.
- La estructuración y adecuación juega un papel importante en el entrenamiento de los diferentes modelos de redes para este tipo de aplicaciones, ya que estos tienen mejor respuesta con valores escalados entre 0 y 1 o entre -1 y 1, mejorando la velocidad de aprendizaje y minimizando el error más rápido.
- El modelo de red y la topología del mismo son de gran importancia en el desarrollo del problema al que se le necesite dar solución, ya que cada modelo de red tiene un grupo de aplicaciones para las cuales son utilizada debido a que tienen un mejor desempeño que otros modelos, debido a esto la buena escogencia del modelo y su topología implica la solución o la no solución de un problema.
- Cuando se le quiere dar solución a un problema con redes neuronales de debe tener en cuenta las características del problema, como lo son las variables de entrada de la red y la variable (es) de salida, por que cuando más claro se tiene esto mejor y más rápido se le puede dar solución, para que no se pierda tiempo en soluciones que no son de ayudan en nada al desarrollo de problema.
- Al momento de hacer la implementación es necesario determinar cuál es el software con mejor condiciones y prestaciones a la hora de hacer el desarrollo del programa, si el software tiene herramientas para este tipo de aplicaciones, como es su velocidad de procesamiento, también se puede

optar por el lenguaje programación en el que se tiene más experiencia programando.

7. RECOMENDACIONES

- Es conveniente recoger más información para las zonas en donde el control basado en redes neuronales no da respuestas acordes con lo que se espera sea la salida.
- El momento de hacer la recolección de la información para el entrenamiento de las redes neuronales artificiales es necesario recoger un gran volumen de información del proceso tratado en este caso el control de flujo de agua, para poder reducir las posibles desinformaciones en el entrenamiento, afectando el desempeño de la red.
- Al necesitar volúmenes de información tan grandes para el entrenamiento de las RNA's es necesario contar además del software, contar con computador adecuado, aunque estos requerimientos está muy ligado al modelo de red a la que se esté entrenando.
- Es indispensable tener los instrumentos y equipos de medición correctamente calibrados para lograr las mediciones con la exactitud necesaria y obtener la respuesta adecuada.

BIBLIOGRAFÍA

KUO, Benjamin C. Sistemas de control automático. Séptima edición. México: Prentice Hall Hispanoamericana, S.A., 1996. p.2. ISBN: 968-880-723-0.

GARCÍA JAIMES, Luís Eduardo. Control Digital: Teoría y práctica. Segunda edición. Medellín: Politécnico Colombiano Jaime Isaza Cadavid. 2009. p.2, [online], disponible en:

< <http://controlpoli.eexweb.com/downloads/books/contb.html> >

ALBA BORDONS, Carlos. Control Predictivo: metodología, tecnología y nuevas perspectivas. España: Universidad de Sevilla. I Curso de Especialización en Automática, 2000, [online], disponible en:

<http://www.esi2.us.es/~bordons/apuntes_MPC.pdf>

ACOSTA, Jesús; FERNANDEZ, Joseph; BECERRA, Lenin. Aplicación de las Redes Neuronales para la Identificación de un Sistema no lineal. España: Comité Español de Automática, [online], disponible en:

<http://www.cea-ifac.es/actividades/jornadas/XXII/documentos/D_02_CI.pdf>

VÁSQUEZ M, Rafael Esteban; VÁSQUEZ C, Juan Miguel. Implementación de un control digital en un PC para un proceso de flujo. Medellín: Universidad Pontificia Bolivariana. Grupo de Investigación en Automática y Diseño A+D, 2004, [online], disponible en:

< http://amasd.upb.edu.co/publicaciones/vasquezr_vasquezj_ACA2004.PDF >

CHAVEZ DURÁN, Marcelo; HURTADO MORATÓ, Reynaldo; JUANIQUINA FLORES, Josefina. Red neuronal para el control de un servomecanismo. Bolivia: Universidad Mayor De San Andrés, [online], disponible en:

<http://pgi.umsa.bo/enlaces/investigacion/pdf/DIPAUDSIS1_93.pdf>

OCHOA RAMÍREZ, Angélica María. Diseño de un sistema de control con algoritmo predictivo para un proceso de control de flujo de agua. Trabajo de grado Ingeniero electrónico. Envigado: Institución Universitaria de Envigado. Facultad de Ingenierías, 2008.

DELGADO, Alberto. Inteligencia Artificial y Minirobots. Segunda edición. Bogotá: Ecoe ediciones, 1998. p. 1.

ACOSTA BUITRAGO, María Isabel; ZULUAGA MUÑOZ, Camilo Alfonso. Tutorial sobre redes neuronales aplicadas en ingeniería eléctrica y su implementación en un sitio web. Trabajo de grado Ingeniero electricista. Pereira: Universidad

Tecnológica de Pereira. Facultad de Ingeniería Eléctrica, 2000, [on line], disponible en:
< <http://ohm.utp.edu.co:16080/neuronales/>>

DEMUTH, Howard; BEALE, Mark; HAGAN, Martin. Neural Network Toolbox 5. USA: The MathWorks, Inc. P. 45-46.

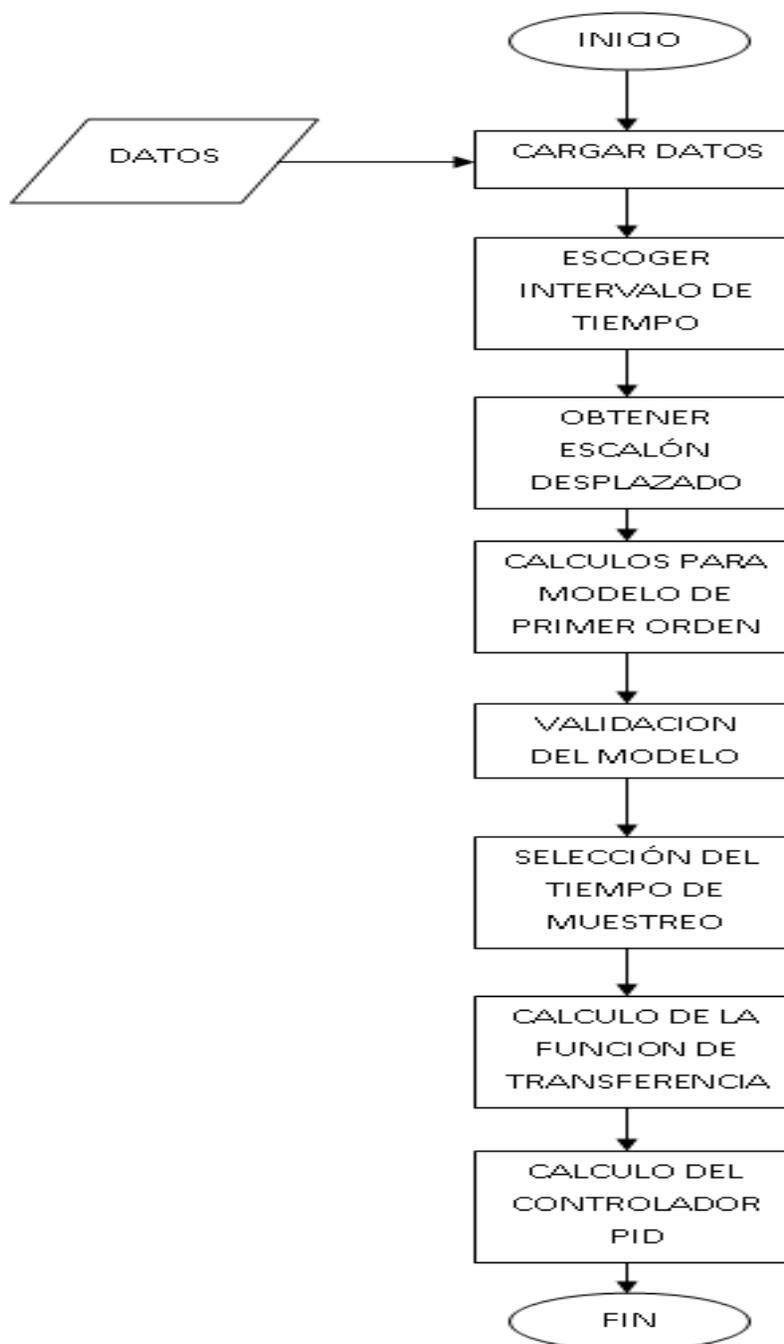
RODRÍGUEZ PONCE, Héctor Uriel. Perceptrón Multicapa para Reconocimiento de Objetos sobre Planos. Puebla, México: Universidad de las Américas Puebla, 2004. p 21, [online], disponible en:
<http://catarina.udlap.mx/u_dl_a/tales/documentos/lem/rodriguez_p_hu/>

MARTÍN DEL BRIO, Bonifacio; SANZ MOLINA, Alfredo. Redes neuronales y sistemas difusos. España: Alfa omega, 2001.p 55-56. ISBN: 970-15-0733-9.

COLMENARES LACRUZ, Gerardo. Redes neuronales. Mérida, Venezuela: Universidad de Los Andes, Instituto de Investigaciones Económicas y Sociales, [online], disponible en:
< <http://webdelprofesor.ula.ve/economia/gcolmen/postgrado1.html>>

ANEXOS

ANEXO A. Programa de identificación y diagrama de flujo.



```

%PROGRAMA DE IDENTIFICACIÓN Y CÁLCULO DEL CONTROLADOR
clear all
clc

%%%%%%%%CARGAR ARCHIVO %%%%%%%%%

Datos=load('D:\Mis Documentos\LabVIEW Data\ident');
t=Datos(:,1);      %me saca unicamente la columna 1 con el tiempo transcurrido;
u=Datos(:,2);      %me saca unicamente la columna 2 de datos y la llama u;
y=Datos(:,3);      %me saca unicamente la columna3 de datos y la llama y;
plot(t,u,t,y)
title('Base de Datos')
grid

%%%%%%%%ESCOGER INTERVALO DE TIEMPO %%%%%%%%%

disp('oprima una tecla')
pause
tini=input('Entre el tiempo inicial tini= ');
tfin=input('Entre el tiempo final tfin= ');
i1=1;
t1=t(i1);
while t1<tini
i1=i1+1;
t1=t(i1);
end
i2=i1;
t2=t(i1);

while t2<tfin
i2=i2+1;
t2=t(i2);
end
t1=t(i1:i2);
u1=u(i1:i2);
y1=y(i1:i2);
i3=1;
ui=u1(i3);
i1=1;
u2=u1(i1);
u3=u1(i1+1);
while u2==u3
i1=i1+1;

```

```

u2=u1(i1);
u3=u1(i1+1);
end

%%%%%%%%OBTENER ESCALON DESPLAZADO %%%%%%%%%

esc=u3-u2;
kff=length(t1);
t2=t1(i1:kff);
u2=u1(i1:kff);
y2=y1(i1:kff);
u3=y1(i1:kff);
t3=t2-t2(1);
u3=u2-u2(1);
y3=y2-y2(1);
figure(2)
plot(t3,u3,t3,y3)
title('Escalon desplazado')
grid
disp('Oprima una tecla')
pause

%%%%%%%%SELECCIONAR ORDEN DEL SISTEMA %%%%%%%%%

p=menu('Seleccione la base de datos','Primer orden','Salir');
if p==1;
clc
tf=length(t3);
tff=floor(tf-(tf/25));
yf=y3(tff:tf);
ymax=mean(yf); %mean: calcula un valor promedio
kpor=ymax/esc;
y283=0.283*ymax;
y632=0.632*ymax;
i2=1;
yp=y3(i2);
while yp<y283
i2=i2+1;
yp=y3(i2);
end
yyi=[y3(i2-1) y3(i2+1)];
tti=[t3(i2-1) t3(i2+1)];

```

```

t283=interp1(yyi,tti,y283,'spline') %interpoleme (aproximar) el valor
correspondiente a y283 % epline : es una expresion que aproxima las funciones
a una expresion cubica
i4=1;
yp1=y3(i4);
while yp1<y632
i4=i4+1;
yp1=y3(i4);
end
yyf=[y3(i4-1) y3(i4+1)];
tff=[t3(i4-1) t3(i4+1)];
t632=interp1(yyf,tff,y632,'spline')
theta=(3*t283-t632)/2;
if theta<=0
theta=0;
np=1;
dp=1;
tao1=t632;

%%%%%%%%VALIDACION DEL MODELO %%%%%%%%%

else
[np,dp]=pade(theta,3);
tao1=t632-theta;
end
npor=kpor;
dpor=[tao1,1];
[ns,ds]=series(np,dp,npor,dpor);
disp('Oprima una tecla')
pause
clc
disp('El sistema Por es: ')
disp(' ')
disp(' ')
printsys(npor,dpor)
disp(' ')%sprintf
sprintf('El retardo es theta %f \n\n', theta)
disp(' ')
disp('Oprima una tecla')
pause
clc
figure(3)
plot(t3,y3,'g')

```

```

hold      %guarda la grafica
step(esc*ns,ds)
title('Validación del Modelo')
grid
disp('Oprima una tecla')
pause
%selección del periodo de muestreo por ancho de banda
[nw,dw]=cloop(npor,dpor); %cloop Hallar la función de transferencia en lazo
cerrado
[mag,fase,w]=bode(nw,dw); %en un vector me coloca la magnitud, la fase para
cada frecuencia que seleccione. Hace la grafica de la respuesta en frecuencia de
los filtros
m=20*log10(mag);          %logaritmo decimal de la magnitud. Paso a db el valor
de la magnitud
m1=m(1,1);              %m1= a la posición 1 del vector m. Escoger el valor de la
magnitud a frecuencia 0
m2=m1-3;                %se le resta los 3 db
wc=interp1(m,w,m2,'spline'); %me interpola entre todos los valores de m.m2 para
obtener w(Wc)
wmin=8*wc;
wmax=12*wc;
Tmin=2*pi/wmax;
Tmax=2*pi/wmin;

%%%%%%SELECCION DEL TIEMPO DE MUESTREO%%%%%%%%

sprintf('El rango del periodo de muestreo es: %f < Ts < %f, \n\n', Tmin, Tmax)
%función activa de c que me permite escribir números en formatos especiales. %f
los números que se escriben se escriben en punto flotante.
disp(' ')              %escribe en la pantalla lo que se ponga en parentesis.
disp('Oprima una tecla')
disp(' ')              %dejar espacio
pause                  %pausa
disp(' ')
T=input('Entre el periodo de muestro deseado T= '); %seleccionamos el periodo
de muestreo dentro del rango hallado anteriormente el cual es el rango sugerido
[a,b,c,d]=tf2ss(npor,dpor); %la función de transf. dada por el
numerador y el denominador me la escriba a espacios de estado
[ad,bd,cd,dd]=c2dt(a,b,c,T,theta); %c2d: Pasar de continua a discreta
pero busca la transf. z sola. c2dm pasa de continuo a discreto con un metodo "zoh"
etc. c2dt: busca la transformada z para sistemas con retardo "continuo a discreto
con tiempo muerto" es la universal sirve para las anteriores
%comienzo a discretizar

```

```

[nd1,dd1]=ss2tf(ad,bd,cd,dd);           %ss2tf: paso de espacio de estado a
funcion de transferencia

km=length(nd1);                         %me dice cuantos terminos hay en el
numerador
for K=1:km;                              %para k=1 hata km
if abs(nd1(K))<10^(-8)                   %
    nd1(K)=0;
else
    nd1(K)=nd1(K);
end
end

%%%%%%%%%FUNCION DE TRANSFERENCIA DISCRETA %%%%%%%%%%

disp('La funcion de transferencia de pulso es: ')
disp(' ')
printsys(nd1,dd1,'Z')
pause
kr=roots(nd1);                           %Encuentra todas la raices del numerador
kr1=length(kr);                           %Kr es el vector con dotas las raices.
luego dice cuantas instrucciones fueron

while kr1>=1;

    while (abs(kr(kr1)))>=1;               %mientras el periodo de muestreo es
mayor a 1 haga
        T=T+T/20;                         %modifica el periodo de muestreo y hace
los calculos de nuevo
        [a,b,c,d]=tf2ss(npord,dpord);
        [ad,bd,cd,dd]=c2dt(a,b,c,T,theta);
        [nd1,dd1]=ss2tf(ad,bd,cd,dd);
        kr=roots(nd1);                     %busca las nuevas raices
        kr1=length(kr);
    end
    kr1=kr1-1;                             %contador quew recorre todas las raices
    en el vector
end

disp(' ')
fprintf('El periodo de muestreo definitivo es T=%f \n', T)
disp(' ')
disp('La funcion de transferencia de pulso es: ')

```

```

disp(' ')
printsys(nd1,dd1,'Z')
disp(' ')
%diseño del controlador
for j=1:100,

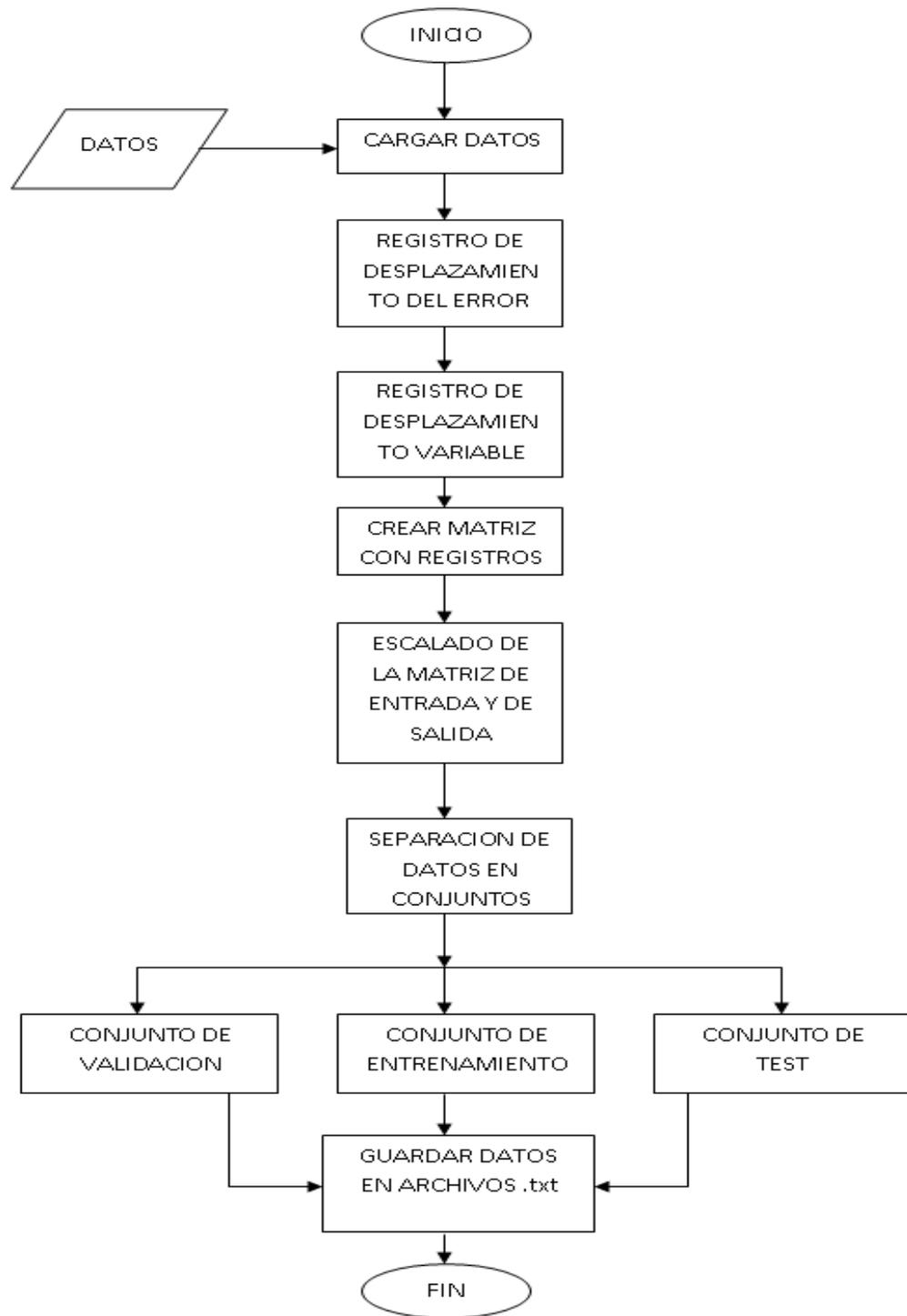
%%%%%%%%%CALCULO DEL CONTROLADOR PID %%%%%%%%%%

c=menu('Seleccione el tipo de Controlador','Proporcional-Integral-Derivativo
(PID)','Salir');
if c==1;
clc
for i=1:100;
k=menu('Selecciones el controlador', 'PID por Ziegler-Nichols','Salir');
if k==1;
%controlador PID por Ziegler-Nichols
disp('PID por Ziegler-Nichols')
thetat=theta+(T/2);
Kc=(1.2*tao1)/(kpor*thetat);
Kc=0.5*Kc;
Ti=2*thetat;
Td=0.5*thetat;
q0=Kc*(1+(T/(2*Ti))+(Td/(T)));
q1=-Kc*(1-(T/(2*Ti))+(2*Td/(T)));
q2=(Kc*Td)/T;
nc=[q0 q1 q2];
dc=[1 -1 0];
disp('El controlador PID por Ziegler-Nichols es: ')
printsys(nc,dc,'Z') %escribe la funcion de transferencia en z
disp(' ')
fprintf('El valor de q0 es:%f \n\n', q0)
disp(' ')
fprintf('El valor de q1 es:%f \n\n', q1)
disp(' ')
fprintf('El valor de q2 es:%f \n\n', q2)
disp(' ')
disp('Oprima una tecla')
sim CPI
pause
clc
elseif k==2
%salir
break

```

```
end
end %end for
elseif c==2
    %salir
break
end
end %for
disp('Oprima una tecla para cerrar y borrar todo')
pause
clc
break;
end
```

ANEXO B. Programa de estructuración y adecuación de datos.



```

%PROGRAMA DE ESTRUCTURACIÓN Y ADECUACIÓN ED LOS DATOS
clear all
clc

%%%%%%%%CARGAR ARCHIVO CON DATOS%%%%%%%%

datos=load ('D:\Mis Documentos\MATLAB\Datos.dat');

sp1=datos(:,2);
yk1=datos(:,3);
error=sp1-yk1;
[j,k]=size(datos);

%%REGISTRO DE DESPLAZAMIENTO DEL ERROR%%

for i=3:j
ek(i-2,1)=error(i,1); %error actual
ek(i-2,2)=error(i-1,1); %error retardado un tiempo de muestro.
ek(i-2,3)=error(i-2,1); %error retardado dos tiempos de muestro.
end

%%REGISTRO DE DESPLAZAMIENTO DE LA VARIABLE%%

for i=3:j
t(i-1)=datos(i,1); %Tiempos de muestreo
yk(i-2,1)=datos(i,3);
yk(i-2,2)=datos(i-1,3);
yk(i-2,3)=datos(i-2,3);
sp(i-2,1)=datos(i,2); %referencia del controlador
uk(i-2,1)=datos(i,4); %salida del controlador para la válvula en (%)
end

%%AGRUPACIÓN DE LAS VARIABLES EN UNA MATRIZ%%

input(:,1)=sp;
input(:,2:4)=yk;
input(:,5:7)=ek;
output=uk';

```

```
%%AGREGANDO MAXIMOS Y MINIMOS PARA EL ESCALADO%%
```

```
output(1,length(output)+1)=100;  
adic=[0 0 0 0 -100 -100 -100;100 100 100 100 100 100 100];  
input(j-1:j,:)=adic;  
input=input';
```

```
[m1,n1]=size(input); %arroja la dimensión de la matriz
```

```
[P4,ps4]=mapminmax(input); %Escala de matriz de entrada  
[T4,ps5]=mapminmax(output); %Escala de vector de salida
```

```
%quitar mínimos y máximos agregados  
P=P4(1:m1,1:n1-2);  
T=T4(1:length(T4)-1);
```

```
%%Separación de los datos en conjunto de entrenamiento, validación y test  
% %60% de los datos para entrenamiento  
% %20% para validación y 20% para test
```

```
[trainV,val,test] = dividevec(P,T,0.20,0.20);
```

```
Entrada=trainV.P;  
objetivo=trainV.T;
```

```
dat_val=val.P;  
res_val=val.T;
```

```
dat_test=test.P;  
res_test=test.T;
```

```
%%Guardando conjunto de entrenamiento%%  
save('D:\Mis Documentos\MATLAB\train_P.txt','Entrada','-ASCII');  
save('D:\Mis Documentos\MATLAB\train_T.txt','objetivo','-ASCII');
```

```
%%Guardando conjunto de validación %%  
save('D:\Mis Documentos\MATLAB\val_P.txt','dat_val','-ASCII');  
save('D:\Mis Documentos\MATLAB\val_T.txt','res_val','-ASCII');
```

```
%Guardando conjunto de test%  
save('D:\Mis Documentos\MATLAB\test_P.txt','dat_test','-ASCII');  
save('D:\Mis Documentos\MATLAB\test_T.txt','res_test','-ASCII');
```

ANEXO C. Programa de entrenamiento de redes neuronales.

```
clear all
clc

%%Cargando conjunto de entrenamiento%%
P=load('C:\TESIS\train_P.txt');
T=load('C:\TESIS\train_T.txt');

%%Cargando datos de validación%%
val_P=load('C:\TESIS\val_P.txt');
val_T=load('C:\TESIS\val_T.txt');
val=struct('P',val_P,'T',val_T); %convirtiendo variables a estructura

%%Cargando datos de test%%
test_P=load('C:\TESIS\test_P.txt');
test_T=load('C:\TESIS\test_T.txt');
test=struct('P',test_P,'T',test_T); %convirtiendo variables a estructura

% Se aplicará el entrenamiento basado en Levenberg-Marquard , con factor de
% aprendizaje variable
net=newff(P,T,[20 17 8 1],{'tansig','tansig','tansig','tansig'},'trainlm', 'learnqdm',
'mse');
net = init(net); %inicialización de la red
Y=sim(net,P); %simulacion de la red sin entrenar
net.trainParam.epochs=300; %máximo número de iteraciones
net.trainParam.goal=0; %error maximo permitido
%net.trainParam.show=10; %muestra resulsitado cada 1o iteraciones
[net,tr]=train(net,P,T,[],[],val,test);
Y=sim(net,P);

W1=net.IW{1,1};
W2=net.LW{2,1};
W3=net.LW{3,2};
W4=net.LW{4,3};
b1=net.b{1};
b2=net.b{2};
b3=net.b{3};
b4=net.b{4};
```

```
%%Guardando los pesos y bias resultantes del entrenamiento
save('C:\TESIS\MLP\W1.txt','W1','-ASCII');
save('C:\TESIS\MLP\W2.txt','W2','-ASCII');
save('C:\TESIS\MLP\W3.txt','W3','-ASCII');
save('C:\TESIS\MLP\W4.txt','W4','-ASCII');
save('C:\TESIS\MLP\b1.txt','b1','-ASCII');
save('C:\TESIS\MLP\b2.txt','b2','-ASCII');
save('C:\TESIS\MLP\b3.txt','b3','-ASCII');
save('C:\TESIS\MLP\b4.txt','b4','-ASCII');
```

ANEXO D. Artículo científico.

IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL DE FLUJO DE AGUA BASADO EN REDES NEURONALES ARTIFICIALES.

ESTUDIANTE

Rubén Darío Uribe Quejada
Institución Universitaria de envigado
E-mail: uriberuben89@hotmail.com

ASESOR

Hugo de Jesús Mesa Yepes
E-mail: hjmesa@iue.edu.co

Resumen: En este trabajo presenta el diseño de un controlador para un sistema de flujo de agua basado en redes neuronales artificiales. Para el desarrollo de este trabajo fue necesaria la implementación de un controlador PID, para ello fue necesario obtener el modelo dinámico de la planta mediante la aplicación de una serie de escalones a la válvula de control, de este proceso se adquirieron los datos necesarios para la identificación del sistema que rige el comportamiento del proceso de flujo de agua; luego de realizar esto, se efectúan los cálculos matemáticos que dan como resultado la parámetros de la función de transferencia del controlador.

Con el controlador calculado, se procede a implementarlo en el modulo de flujo de agua, para hacer una nueva adquisición de datos, pero esta vez con el proceso controlado, después, la base de datos generada por el proceso, es sometida a una estructuración y adecuación para el posterior entrenamiento de las redes neuronales artificiales, ya que las redes tiene mejores respuestas en el entrenamiento con valores escalados entre ciertos rangos como -1 y 1; luego, se procese con el entrenamiento de las redes y se valora su desempeño, dependiendo de este desempeño se escoge la mejor red mejor red; esta mejor red es la que será implementada para controlara el sistema de flujo de agua, con la red neuronal ya implementada se validara que tan eficiente es comparándola con el controlador PID anteriormente implementado.

Palabras claves: Base de datos, controlador PID, entrenamiento, escalado, flujo, modelo dinámico, redes neuronales artificiales.

Abstract: This paper presents the design of a controller for a water flow system based on artificial neural networks. For the development of this work was necessary to implement a PID controller, it was necessary to obtain the dynamic model of the plant by applying a series of steps to the control valve of this process of acquiring the necessary data for the identification of the system governing the behavior of water flow process, then to accomplish this, the mathematical calculations are performed that result in the parameters of the transfer function of the controller.

With the controller calculated, we proceed to implement it in the water flow module, to make a new data acquisition, but this time with the controlled process, then the database generated by the process, is subjected to a structured and suitability for further training of artificial neural networks because the networks have better answers in training with scaled values between certain ranges as -1 and 1, then is processed with the training of the networks and their performance is evaluated, depending This performance was better network chooses the best network, this network is the best that will be implemented to control water flow system with the neural

network is implemented and how effective is validated by comparing with the PID controller previously implemented.

Keywords: Database, PID controller, training, scaling, flow, dynamic model, artificial neural networks.

1. INTRODUCCIÓN

En la actualidad se busca generar una solución para los problemas de control que son tan frecuentes que se solucionan en su gran medida por los métodos tradicionales de control, por esto se optó por buscar una solución fuera de lo convencional como lo son las redes neuronales artificiales que ofrecen una alternativa novedosa que simplifica y permite comparar su eficiencia de este nuevo enfoque respecto a las metodologías tradicionales, a través de conocimientos matemáticos de un nivel aceptable para comprender el fundamento y desarrollo del problema que se quiere resolver. Las redes neuronales artificiales, son una imitación del proceso de una neurona biológica, e imitan las características que hacen tan llamativas y prácticas, como lo son la tolerancia a fallas, procesamiento paralelo, son robustas, flexibles, consumen poca energía, lo que le da una ventaja sobre los computadores convencionales que se basan en la arquitectura de Von Neumann.

La capacidad del cerebro humano de pensar, recordar y resolver problemas ha inspirado a muchos científicos intentar o procurar modelar en el ordenador el funcionamiento del cerebro humano.

Los profesionales de diferentes campos como la ingeniería, filosofía, fisiología y psicología han unido sus esfuerzos debido al potencial que ofrece esta tecnología y están encontrando diferentes aplicaciones en sus respectivas profesiones. En este caso las aplicaciones se enfocaría en los problemas que tengan que ver con en campo del control de procesos, como lo son control de nivel, flujo, temperatura y presión que son procesos que se presentan con frecuencia en la industria lo que indica que hay posibilidades casi ilimitadas de aplicación de las RNA para el control.

En el primer capítulo, se trata de todo lo que tiene que ver con al origen del problema que se quiere tratar, el objetivo general de la investigación, objetivos específicos, la justificación del por qué es necesario el desarrollo del proyecto, que alcance tendrá y que metodología se aplicará.

En el segundo capítulo, se hace una documentación necesaria para el desarrollo del inicio del proyecto, como lo son los antecedentes referentes al control de flujo de agua y controles donde utilizan redes neuronales, además, se hace una identificación de las características del módulo de flujo de agua, la dinámica del proceso y el desarrollo de un controlador convencional PID.

En el tercer capítulo, se documenta sobre el modelo neuronal biológico y artificial, las funciones a activación de las RNA's, algunos modelos de RNA's, además, se hace el proceso de entrenamiento de las RNA's, para una posterior implementación, este proceso se lleva a cabo con el Toolbox de Matlab para redes neuronales.

En el cuarto capítulo, se implementa la RNA que mejor comportamiento tuvo en el proceso de entrenamiento, se desarrolla la red en un programa de interfaz grafica como lo es LabView y utilizando en este funciones de Matlab.

En el quinto capítulo, se hacen pruebas al controlador convencional y al controlador basado en RNA's para determinar cuál de los controladores es más eficiencia controlando el sistemas de flujo de agua.

2. IDENTIFICACIÓN DE LAS CARACTERÍSTICAS DEL MÓDULO DE FLUJO

Se realizará una descripción de las características del módulo del flujo y la dinámica del sistema.

2.1 Descripción del módulo

El módulo didáctico de flujo-nivel de agua que se encuentra presente en el laboratorio de Automatización y Robótica fue elaborado en la Institución Universitaria de Envigado, el diseño y la elaboración fue supervisada por el docente en las áreas de control digital y seminario III (control avanzado) Luis Eduardo García Jaimes Ms en automatización y control. El Módulo para monitoreo y control de flujo y nivel puede utilizarse para realizar prácticas de instrumentación industrial e implementar una gran variedad de estrategias de control análogo y control digital de procesos. Entra la gran variedad de características que tiene el módulo se encuentran las siguientes:

- Medición de nivel.
- Control on-off de nivel.
- Control análogo de nivel utilizando control P, PI y PID.
- Control digital de nivel utilizando control P, PI, PID y otros algoritmos.
- Medición de flujo.
- Control on-off de flujo.
- Control análogo de flujo utilizando control P, PI y PID.
- Control digital de flujo utilizando control P, PI, PID y otros algoritmos.
- Identificación de sistemas.
- Aplicación de técnicas de control predictivo basado en modelos.
- Aplicación de técnicas de control adaptativo.
- Implementación de sistemas de control borroso, neuronal y neuro-borroso.

El módulo para monitoreo y control de flujo y nivel es una réplica de un proceso de

producción real en una escala menor que consta de dos tanques: uno en la parte inferior que se utiliza como tanque de reserva de agua y otro tanque en la parte superior, sobre el cual se realiza control de nivel.

La figura 1 muestra el modulo flujo-nivel del laboratorio de Automatización y Robótica de la IUE.

Figura 1. Módulo flujo-nivel.



2.2 Modelación del sistema.

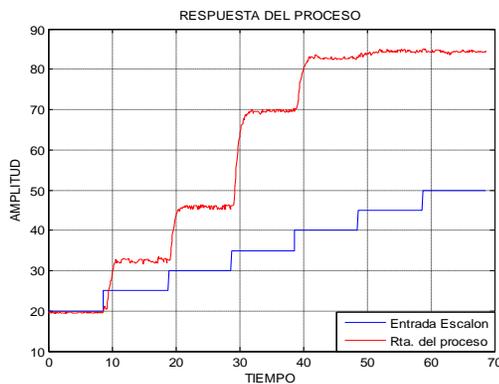
Con los datos obtenidos del módulo con el sensor de flujo y que se almacenaron en un archivo plano con extensión .dat, con este archivo que se procesará en software desarrollado en Matlab para calcular los parámetros del proceso, la dinámica del sistema y con base a esta se obtendrá la ecuación que representa el modelo del módulo de flujo (ecuación de la planta) para el desarrollo del algoritmo de control convencional que es el punto de referencia para la implementación del control basado en redes neuronales artificiales.

Con el software en Matlab se muestra paso por paso el desarrollo de los cálculos, de manera que se entienda el procedimiento y los resultados obtenidos. A continuación se mostrará el proceso llevado a cabo en el

software para la identificación del modelo matemático del sistema.

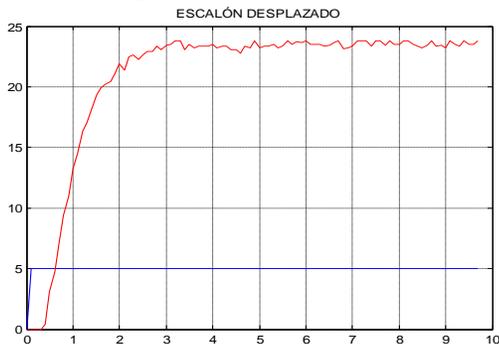
Al iniciar el procesamiento se muestra graficado los datos que se almacenaron anteriormente en la operación manual de la válvula con la señal de entrada tipo escalón y la respuesta del sistema que en este caso sería la variación del flujo con dicha entrada, como lo muestra la figura 2.

Figura 2. Respuesta del proceso.



Con se ve en la figura 2, hay varios escalones, pero para este caso se toma el escalón con mejor comportamiento frente a la entrada aplicada para obtener un mejor resultado, este escalón es el que se encuentra entre los tiempos de 20 (tiempo inicial) y 40 (tiempo final), después de que se escoge en el software el escalón deseado este se amplía y se desplaza de manera que el tiempo inicial que era de 20 se mueva hasta el origen (hasta cero) como lo muestra la figura 3.

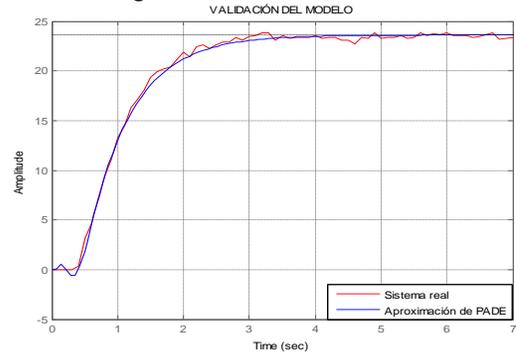
Figura 3. Escalón desplazado



Esta ampliación y este desplazamiento se hace para hallar los tiempos correspondientes al 28.3% y 63.2% para una aproximación a un sistema de primer orden con retardo (POR).

Los resultados obtenidos en Matlab para el 28.3%=0.7294 y para 63.2%=1.4081, además, en el software también se obtiene el modelo del sistema aproximado de primer orden con retardo que se observa en la figura 4 (validación del modelo) y su representación matemática en la ecuación.

Figura 4. Validación del modelo.



$$G(s) = \frac{4.7227e^{-0.459841s}}{0.67585s + 1}$$

Por otro lado, se obtiene un rango del periodo de muestreo $0.130943 < T_s < 0.196414$, para el cual se escogió 0.19 con lo que se tiene una función de transferencia de pulso de la siguiente manera:

$$HG(Z) = \frac{1.6441Z + 1.5195}{Z^2 - 0.33012Z}$$

2.3 Elementos del control convencional PID.

Definición. Un controlador es un dispositivo que compara el valor de la variable controlada con el valor deseado, establece la diferencia o error y la utiliza para realizar la acción de control, con el fin de reducir el error a 0 o a un valor mínimo aceptable.

Forma de calcularlo. Para trabajar con el control PID, existen diferentes algoritmos

para resolverlo ellos son: Ziegler–Nichols, criterio de Error, Ganancia limite, donde el Ziegler–Nichols es uno de los métodos más confiables por facilidad de cálculos, y mejores resultados del controlador; es decir para que el sistema trabaje más eficientemente. Primordialmente se debe de aplicar solo a sistemas de primer orden con retardo (POR), si por alguna razón se utiliza ecuaciones de orden superior se debe de llevar esta a orden uno, la ecuación que la rige es [LUI 09]:

$$Gp(s) = \frac{K\theta's}{\tau s + 1} \quad \text{Eq. (1)}$$

Donde:

K = Ganancia del sistema.

$\theta'S$ = Retardo del sistema.

τS = Constante de tiempo del sistema.

En la tabla están los ajustes de los parámetros:

Tabla 1. Parámetros del controlador PID por Ziegler-Nichols.

Control/ Parámetro	k_c	τ_i	τ_d
P	$\tau/k\theta$	-	-
PI	$0.9\tau/k\theta$	3.33θ	-
PID	$1.2\tau/k\theta$	2θ	0.5θ

2.4 Cálculo del controlador por Ziegler-Nichols.

Para el proceso de entrenamiento de las redes neuronales en necesario la implementación del controlador por Ziegler-Nichols, por lo que se tiene la función del sistemas de primer orden con retardo (POR) para hacer los cálculos necesarios.

$$G(s) = \frac{4.7227e^{-0.459841s}}{0.67585s + 1}$$

$$T = 0.749033$$

$$\theta = \theta' + T/2$$

$$\theta = 0.459841 + 0.749033/2$$

$$\theta = 0.8343575$$

$$k_c = 1.2 * \tau / k \theta$$

$$k_c = (1.2 * 0.67585) / (4.7227 * 0.8343575)$$

$$k_c = 0.205820$$

$$\tau_i = 2\theta = 1.668715$$

$$\tau_d = 0.5\theta = 0.417178$$

$$q_0 = k_c \left[1 + \frac{T}{2\tau_i} + \frac{\tau_d}{T} \right] = 0.183324$$

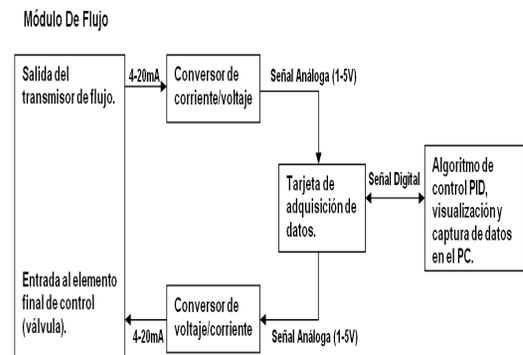
$$q_1 = -k_c \left[1 - \frac{T}{2\tau_i} + \frac{\tau_d}{T} \right] = -0.194447$$

$$q_2 = \frac{k_c \tau_d}{T} = 0.057317$$

2.5 Implementación del controlador por Ziegler-Nichols.

Después de de tener calculada la función de transferencia del controlador, se procede a implementar el control en el proceso, para este es necesario contar con software LabView instalado en el computador, en este software se hace el desarrollo del algoritmo de control, visualización para el monitoreo de las variables y captura de los datos cada tiempo de muestreo para ir generando la base de datos, que serán los datos para el entrenamiento de las RNA's y sigan comportamiento del controlador PID. La figura 5 muestra el diagrama de la implementación del controlador.

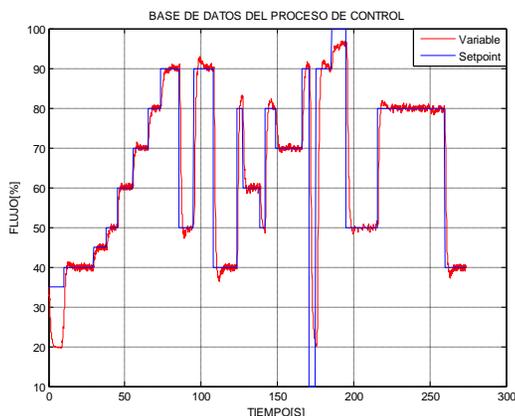
Figura 5. Diagrama de bloques del proceso de control PID.



2.6 Generación de la base de datos.

Al mismo tiempo que se tiene implementado el control PID en el proceso de flujo de agua, también se están guardando los datos cada tiempo de muestreo en un archivo plano, en este archivo queda reflejado el comportamiento de la variable controlada y cómo trata de alcanzar el set-point lo más rápido posible, y la acción necesaria del controlador en el elemento final de control (válvula) para que el error llegue a cero, que es el objetivo del control del proceso. Este archivo plano está comprendido por 4 columnas, en la cual primera columna es el tiempo de muestreo, la segunda columna es el set-point, la tercera es la variable controlada y la cuarta columna es la acción del elemento final de control (la válvula), esta base de datos es necesaria para el entrenamiento de las redes neuronales artificiales, pero antes del entrenamiento a estos datos es necesario hacerles un procesamiento previo. La figura 6 muestra la grafica de los datos recolectados.

Figura 6. Base de datos del proceso de control.



2.7 Estructuración y adecuación de los datos.

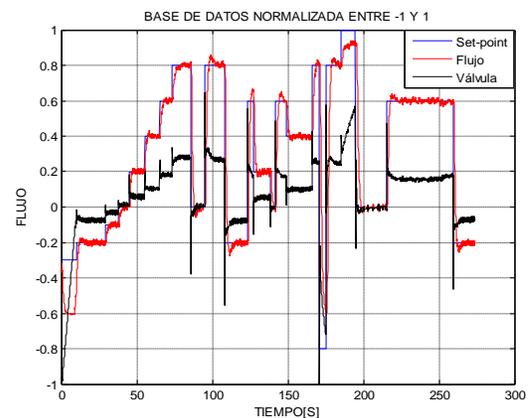
Esta estructuración y adecuación de los datos se lleva a cabo para un posterior entrenamiento de las redes neuronales artificiales, para esto es necesario que la información sea más completa y más variada, por lo cual se aplica derivada en

diferencias en cada intervalo de muestreo a la variable controlada que es la tercera columna de la base de datos, de manera que se tenga más información para el entrenamiento de las RNA's (Redes Neuronales Artificiales).

$$y'(i) = \frac{y(i) - y(i-1)}{t(i) - t(i-1)}$$

Además, estos datos se encuentran en una escala muy amplia (entre 0 y 100), y como las redes neuronales entrenan y trabajan mejor en valores entre -1 y 1, por esto se hace un proceso de normalización con las herramientas de Matlab como lo es función *mapminmax* como lo muestra la figura donde se observa que el valor máximo es 1 y el mínimo es -1. La figura 7 la base de datos escalada.

Figura 7. Base de datos escalada.



3. ENTRENAMIENTO DE DIFERENTES MODELOS DE RNA's.

En esta etapa del proceso se tiene en cuenta las características del problema que se desea solucionar, como el número de variable de entrada, en número de variables de salida, tipo de función de activación según los requerimientos, en modelo de red neuronal artificial más adecuado, con que tanto volumen de información cuenta la base de datos para realizar el proceso de

entrenamiento, validación y test de la red [HMM 07].

En esta etapa de entrenamiento se experimento con varios modelos y topologías de redes, de las cuales tres mostraron resultados significativos que seran la que se mostraran a continuación.

3.1 Entrenamiento del preceptrón multicapa (MLP).

En esta topología se uso funciones de activación del tipo tansig en todas las capas.

Tabla 2. Valores de la topología MLP 4.

Valores de la topología MLP 4	
No. De nodos en la capa de entrada	7
No. De capas ocultas	1
No. De nodos en la capa oculta	9
No. De nodos en la capa de salida	1
Error máximo permitido	0
No. Máximo de iteraciones	100

Figura 8. Topología MLP 4.

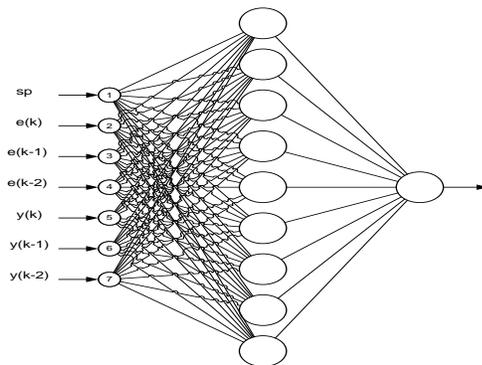


Figura 9. Error del MLP 4

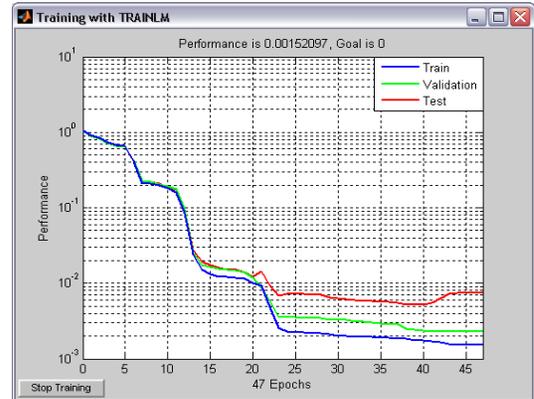
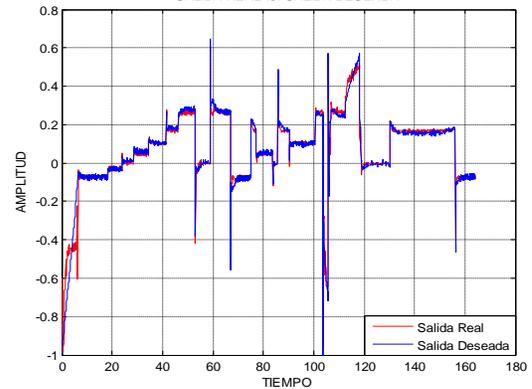


Figura 10. Simulación del MLP 4.
SALIDA REAL vs. SALIDA DESEADA



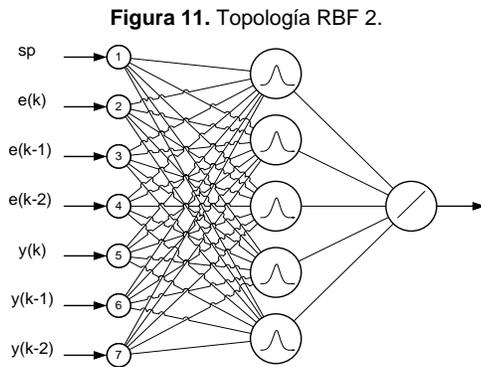
3.2 Entrenamiento del RBF.

Para este tipo de red el entrenamiento es mucho más sencillo y rápido, en la creación de la red y entrenamiento también se hace uso de la funciones del Neural Network Toolbox, que tiene una función para redes de funciones de base radial que se llama newrb, tiene como los siguientes parámetros y estructura.

Para iniciar con el proceso de entrenamiento de la red utilizaremos una res con las siguientes característica:

Tabla 3. Valores de la topología RBF 2.

Valores de la topología RBF 2	
No. De nodos en la capa de entrada	7
No. De capas ocultas	1
No. De nodos en la capa oculta	5
No. De nodos en la capa de salida	1
Error máximo permitido	0
No. Máximo de iteraciones	5



Valores de la topología RBF 3

No. De nodos en la capa de entrada	7
No. De capas ocultas	1
No. De nodos en la capa oculta	8
No. De nodos en la capa de salida	1
Error máximo permitido	0
No. Máximo de iteraciones	8

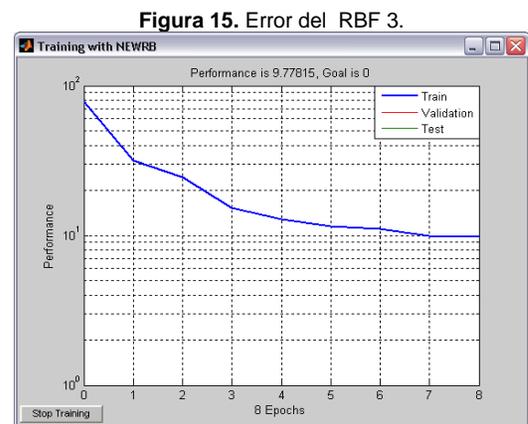
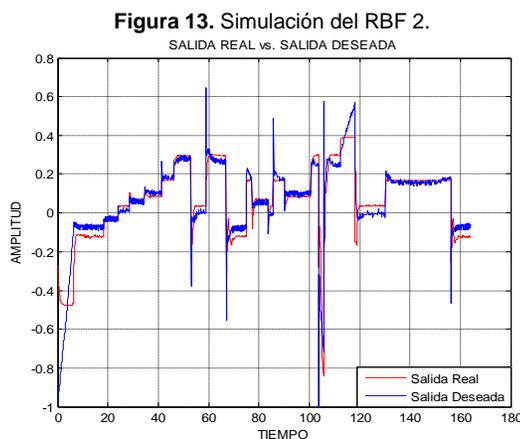
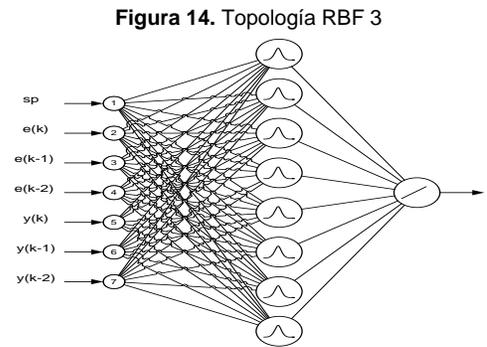
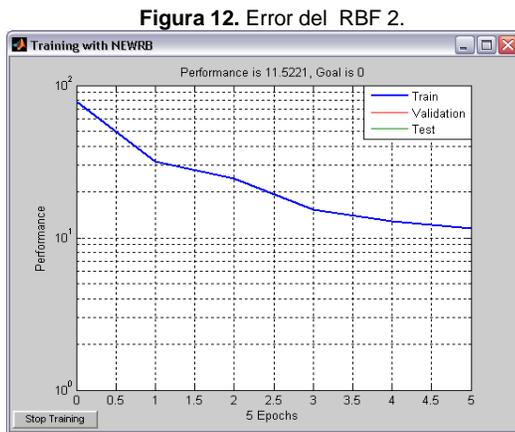
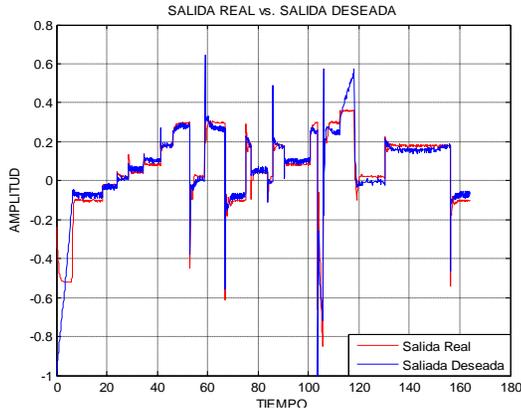


Figura 16. Simulación del RBF 3.

Esta es la última de las tres redes que mostro buenos resultados, con las siguientes características:

Tabla 4. Valores de la topología RBF 3



Los valores de los pesos de todas las capas se guardaron para poder hacer la implementación de estas mismas redes, también, se guardo los vales de los bías.

Teniendo en cuenta la interpretación de la información los modelos que mejor comportamiento tuvieron con respecto a los demás fueron el MLP 4, el RBF 2 y RBF 3, siendo el más destacado de estos tres el MLP 4, por lo que es el modelo al que se la hará el desarrollo para controlara el sistemas de flujo de agua, con los valores de los pesos y los bías que se habían guardado con anterioridad para el posterior desarrollo del controlador.

4. DESARROLLO EL CONTROLADOR BASADO EN RNA'S.

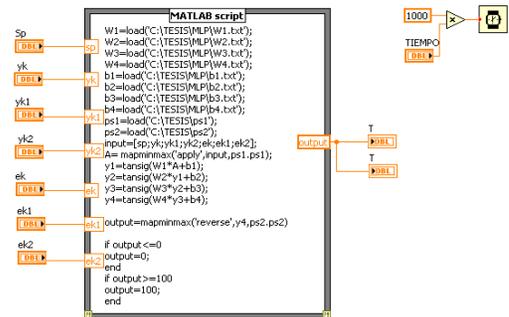
Después de tener definida la topología y modelo de RNA (Red Neuronal Artificial), se procede con el desarrollo del programa para el control, para el cual se utiliza LabView; que es una herramienta de lenguaje gráfico, con la cual se puede hacer el procesamiento de los datos, graficarlos y hacer uso de sus interfaces de comunicación como lo es la interfaz del puerto serie, que es la que se utiliza en este caso para comunicar el proceso a través de la tarjeta de adquisición de datos con el PC y viceversa.

4.1 Desarrollo del modelo de RNA's en LabView.

Para hace el desarrollo de la RNA en LabView fue necesario utilizar una función de

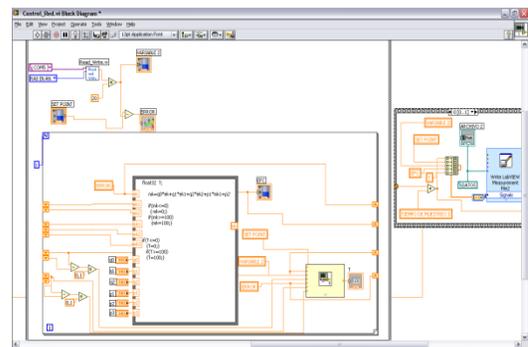
comunicación entre este software y Matlab, ya que la red neuronal fue creada y entrenada en Matlab; este bloque permite utilizar funciones propias de Matlab, permite cargar datos, tener variables de entradas provenientes de LabView y variables de saldas hacia LabView, como se muestra en la figura 17, donde los iconos de la izquierda son variables de entrada y los de la derecha son variables de salida del Matlab script [MAN 05].

Figura 17. Función de Matlab en LabView.



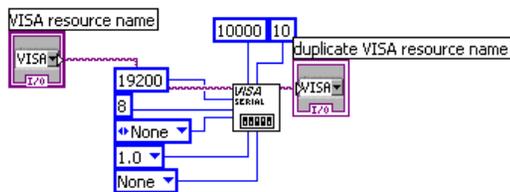
La figura 18, muestra el desarrollo el programa para el control basado en RNA para la implementación en el proceso de flujo de agua, en este programa se encuentra la comunicación entre LabView y el proceso, el procesamiento de los datos para hacer los registros de desplazamiento para obtener los valores anteriores de las variables que son las entradas de la red para que esta los procese y de la salida.

Figura 18. Programa de control con RNA's.



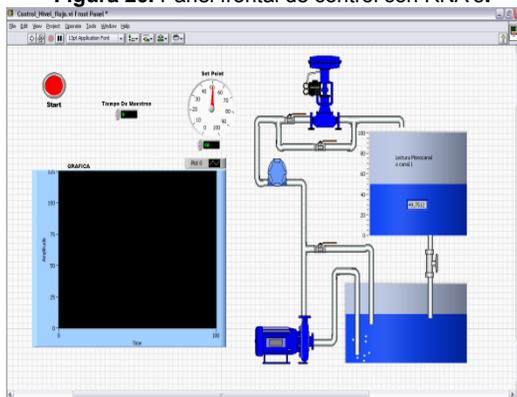
Para la comunicación con el proceso se debe configurar el puerto por donde se van a recibir y a transmitir los datos, en este caso el puerto de comunicación es el serie, que tiene la siguiente configuración como lo muestra la figura 53, donde la velocidad de transmisión y recepción es 19200 Baudios, a 8 bits y sin bit de paridad.

Figura 19. Configuración del puerto serial.



La figura 20 muestra el panel frontal del programa de control basado en RNA's, en este se podrá visualizar la variable del proceso, la apertura del elemento final de control, y el valor de referencia deseado, entre otras cosas.

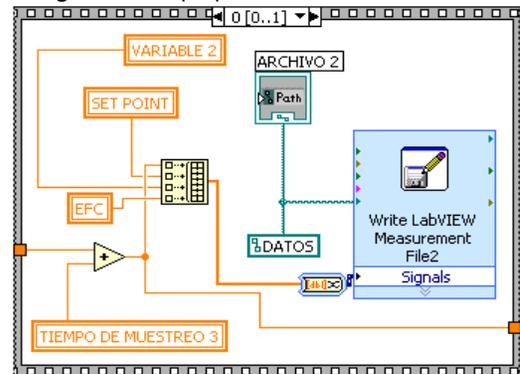
Figura 20. Panel frontal de control con RNA's.



5. VALIDACIÓN DEL COMPORTAMIENTO DEL SISTEMAS DE CONTROL DESARROLADO EN EL MÓDULO DE FLUJO-NIVEL.

En esta etapa se genera un archivo plano de los datos del proceso de control basado en RNA's ya implementado para determinar el comportamiento de la red, como lo muestra la figura 21, donde se guarda el tiempo de muestreo, el set point, el valor de la variable y la acción del elemento final de control, con estos datos se puede saber cuál es el error para compararlo con el control convencional.

Figura 21. Bloque para almacenamiento de datos.



Para iniciar se proceda a aplicar pruebas al proceso con el controlador convencional PID por Ziegler-Nichols desarrollado e implementado en la primera etapa del proyecto para ver que tan eficiente es este tipo de control. Las pruebas a las que se someterá el controlador PID serán de la misma clase a la que se someterán al controlador basado en redes neuronales; serán una serie de escalones y cambios del set point para ver como el controlador trata de llegar a la dicha referencia, como lo muestran las figuras 22 (pruebas del proceso de control convencional) y la figura 23 (pruebas al proceso de control basado en RNA's).

Figura 22. Pruebas al control convencional.

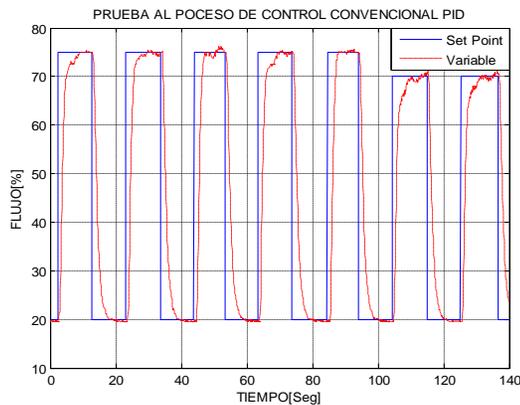
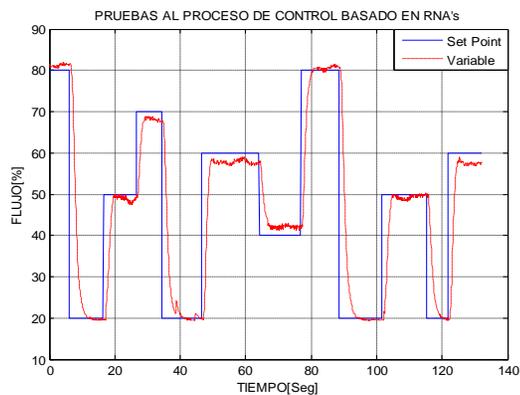


Figura 23. Pruebas al control basado en RNA's.



De la información recolectada se puede determinar que el controlador con menor error a los largo de la prueba fue el convencional, manteniendo la variable controlada lo más cercana posible a al el set point para que el error fuera mínimo.

6. CONCLUSIONES

➤ En el momento de implementar el algoritmo de control convencional PID (Proporcional Integral Derivativo) en el modulo de flujo de agua, es necesario recopilar un volumen de información suficiente dentro del rango de funcionamiento del proceso para tener un entrenamiento optimo de la red neuronal

ya que en ocasiones al no tener a información adecuada en ciertos rangos de funcionamiento la red puede presenta problemas y no se obtendrá la salida deseada, y la red podría trabajar bien en los rangos donde si se recolecto la información necesaria.

➤ La estructuración y adecuación juega un papel importante en el entrenamiento de los diferentes modelos de redes para este tipo de aplicaciones, ya que estos tienen mejor respuesta con valores escalados entre 0 y 1 o entre -1 y 1, mejorando la velocidad de aprendizaje y minimizando el error más rápido.

➤ El modelo de red y la topología del mismo son de gran importancia en el desarrollo del problema al que se le necesite dar solución, ya que cada modelo de red tiene un grupo de aplicaciones para las cuales son utilizada debido a que tienen un mejor desempeño que otros modelos, debido a esto la buena escogencia del modelo y su topología implica la solución o la no solución de un problema.

➤ Cuando se le quiere dar solución a un problema con redes neuronales de debe tener en cuenta las características del problema, como lo son las variables de entrada de la red y la variable (es) de salida, por que cuando más claro se tiene esto mejor y más rápido se le puede dar solución, para que no se pierda tiempo en soluciones que no son de ayudan en nada al desarrollo de problema.

➤ Al momento de hacer la implementación es necesario determinar cuál es el software con mejor condiciones y prestaciones a la hora de hacer el desarrollo del programa, si el software tiene herramientas para este tipo de aplicaciones, como es su velocidad de procesamiento, también se puede optar

por el lenguaje programación en el que se tiene más experiencia programando.

Envigado, miembro del grupo de investigación de Automatización y Robótica.

7. RECOMENDACIONES

- Es conveniente recoger más información para las zonas en donde el control basado en redes neuronales no da respuestas acordes con lo que se espera sea la salida.
- El momento de hacer la recolección de la información para el entrenamiento de las redes neuronales artificiales es necesario recoger un gran volumen de información del proceso tratado en este caso el control de flujo de agua, para poder reducir las posibles desinformaciones en el entrenamiento, afectando el desempeño de la red.
- Al necesitar volúmenes de información tan grandes para el entrenamiento de las RNA's es necesario contar además del software, contar con computador adecuado, aunque estos requerimientos está muy ligado al modelo de red a la que se esté entrenando.
- Es indispensable tener los instrumentos y equipos de medición correctamente calibrados para lograr las mediciones con la exactitud necesaria y obtener la respuesta adecuada.

REFERENCIAS.

[LUI 09] García Jaimes, Luís. Sistemas de control digital. Teoría y práctica. Politécnico Colombiano Jaime Isaza Cadavid. 2009.

[MAN 05] LAZARO, Manuel Antoni, RIO FERNANDEZ, Joaquín del. Labview 7.1, España: Thompson Editores Spain, 2005. ISBN: 84-9732-391-2

[HMM 07] DEMUTH, Howard; BEALE, Mark; HAGAN, Martin. Neural Network Toolbox 5. USA: The MathWorks, Inc.

C.V.: Rubén Darío Uribe Quejada, ingeniero electrónico de la Institución Universitaria de