

# IMPLEMENTACIÓN DE ESQUEMAS DE CONTROL NEURONAL BAJO CONTROL - LOGIX

Br. Pérez Rivas Lisbeth Carolina  
Profesor Guía: Prof. Francklin Rivas Echeverria  
Cotutor: Prof. Addison Ríos Bolivar

[www.bdigital.ula.ve](http://www.bdigital.ula.ve)

PROYECTO DE GRADO PRESENTADO ANTE LA ILUSTRE UNIVERSIDAD DE LOS ANDES  
COMO REQUISITO FINAL PARA OPTAR AL TÍTULO DE INGENIERO DE SISTEMAS

Mérida, Venezuela  
Septiembre 2005



Universidad de Los Andes © Derechos Reservados 2005

*Dedicado a todas aquellas personas  
que hicieron posible el terminar este  
largo camino*

[www.bdigital.ula.ve](http://www.bdigital.ula.ve)

# Índice

Índice de Figuras	vii
Agradecimientos	x
Resumen	xi
<b>1 Introducción</b>	<b>1</b>
<b>2 Autómatas Programables</b>	<b>3</b>
2.1 Introducción	3
2.2 Historia	4
2.3 Aplicaciones	5
2.4 Estructura	6
2.4.1 Externa	6
2.4.2 Interna	7
2.5 Funcionamiento	8
2.6 Plataforma Logix	9
2.7 Control - Logix	10
2.7.1 Módulos de Entrada/Salida Analógicas	12
2.7.2 Módulos de Entrada/Salida Digitales	15
2.7.3 Módulo de interfase de Comunicación Ethernet	15
2.8 Lenguajes de Programación	16
2.8.1 Diagrama de Contactos	17
2.8.2 Funciones de Diagramas de Bloque	17
2.8.3 Texto Estructurado	18

2.8.4	Listado de instrucciones . . . . .	18
2.9	Software de Programación . . . . .	18
<b>3</b>	<b>Redes Neuronales</b>	<b>21</b>
3.1	Introducción . . . . .	21
3.2	Redes Neuronales Biológicas . . . . .	22
3.3	Redes Neuronales Artificiales . . . . .	23
3.3.1	Antecedentes históricos . . . . .	25
3.3.2	Modelo matemático . . . . .	27
3.4	Clasificación de los Esquemas Neuronales Artificiales . . . . .	30
3.4.1	Según su Topología . . . . .	31
3.4.2	Según el Patrón de Conexión . . . . .	32
3.4.3	Según la Naturaleza de las Señales de Entrada . . . . .	33
3.5	Proceso de Aprendizaje . . . . .	33
3.5.1	Aprendizaje Supervisado . . . . .	33
3.5.2	Aprendizaje No Supervisado . . . . .	35
3.6	Algoritmo de Entrenamiento de Retropropagación . . . . .	37
3.6.1	Regla Delta Generalizada . . . . .	37
3.7	Control Neuronal . . . . .	39
3.7.1	Esquemas de control neuronal . . . . .	39
<b>4</b>	<b>Control Neuronal</b>	<b>43</b>
4.1	Introducción . . . . .	43
4.2	Descripción de los Elementos del Sistema . . . . .	45
4.2.1	Surtidor Centrífugo. . . . .	45
4.2.2	Válvulas solenoides. . . . .	45
4.2.3	Servoválvula. . . . .	46
4.2.4	Medidor de Flujo Visual. . . . .	46
4.2.5	Transmisor de Pulsos de Flujo. . . . .	47
4.2.6	Transmisor de Nivel Flotante (FLT) . . . . .	47
4.3	Identificación del Sistema . . . . .	47
4.3.1	Identificación durante la Apertura de la Servoválvula . . . . .	53

4.3.2	Identificación durante Cierre de la Servoválvula . . . . .	55
4.3.3	Modelo Paramétrico (ARX 1 1 10) . . . . .	58
4.4	Diseño del Controlador Clásico . . . . .	61
4.5	Entrenamiento de la Red Neuronal . . . . .	68
4.5.1	Controlador Neuronal de 5 capas . . . . .	70
4.5.2	Controlador Neuronal de 4 Capas . . . . .	74
4.6	Construcción de la Red Neuronal Bajo Control - Logix . . . . .	79
4.7	Supervisión del Sistema . . . . .	80
<b>5</b>	<b>Conclusiones</b>	<b>82</b>
	<b>Bibliografía</b>	<b>84</b>
<b>A</b>	<b>Controlador Proporcional Integral</b>	<b>87</b>
<b>B</b>	<b>Controlador Neuronal</b>	<b>89</b>
B.1	Código Fuente en Texto Estructurado del Control neuronal . . . . .	90
<b>C</b>	<b>Programación de la Red Neuronal</b>	<b>95</b>
C.1	Interfaz Gráfica . . . . .	96
C.2	Generador del Código Fuente en Texto Estructurado . . . . .	99

# Índice de Figuras

2.1	Estructura Interna Básica de un Controlador Lógico Programable . . . .	7
2.2	Ciclo de Scan . . . . .	9
2.3	Controladores Lógicos Programables de la Plataforma Logix . . . . .	10
2.4	Control - Logix . . . . .	11
2.5	Módulo de Entrada/Salida Analógica . . . . .	14
2.6	Diagrama Escalera . . . . .	17
2.7	Listado de Instrucciones . . . . .	18
2.8	Software de Programación para la plataforma Logix . . . . .	20
3.1	Neurona Biológica . . . . .	23
3.2	Relación entre las neuronas biológicas y artificiales . . . . .	24
3.3	Función Sigmoide Unipolar . . . . .	29
3.4	Función Sigmoide Bipolar . . . . .	29
3.5	Función Tangente Hiperbólica . . . . .	30
3.6	Función Lineal . . . . .	30
3.7	Red Monocapa . . . . .	31
3.8	Red Multicapa . . . . .	32
3.9	Aprendizaje Supervisado . . . . .	34
3.10	Aprendizaje No Supervisado . . . . .	36
3.11	Esquema de Control Clásico . . . . .	41
3.12	Adquisición de datos del Controlador Clásico . . . . .	41
4.1	Basic Process Rig 38-100 . . . . .	44
4.2	Relación entre Corriente y Altura . . . . .	49
4.3	Relación entre Medida de Flujo y Corriente . . . . .	50

4.4	Señal de Nivel y Corriente en la Servoválvula . . . . .	51
4.5	Nivel de Agua en el Tanque Superior y Corriente en la Servoválvula . . . . .	51
4.6	Identificación Aproximada . . . . .	52
4.7	Datos Experimentales . . . . .	53
4.8	Comparación entre el Modelo Identificado y Datos Reales . . . . .	54
4.9	Validación del Modelo Identificado durante Cierre de la Servoválvula . . . . .	54
4.10	Datos Experimentales . . . . .	55
4.11	Comparación entre el Modelo Identificado y Datos Experimentales . . . . .	56
4.12	Validación del Modelo Identificado durante Apertura de la Servoválvula . . . . .	56
4.13	Comparación entre el Modelo Identificado y Datos Experimentales durante Apertura de la Servoválvula . . . . .	58
4.14	Comparación entre el Modelo Identificado y Datos Experimentales durante Cierre de la Servoválvula . . . . .	58
4.15	Esquema del sistema en lazo abierto . . . . .	59
4.16	Curva de Estado Estacionario . . . . .	60
4.17	Diagrama de Control . . . . .	63
4.18	Respuesta del Sistema ante el Controlador PID . . . . .	64
4.19	Respuesta del Nivel de Agua ante el Controlador PID . . . . .	64
4.20	Acción de Control Proporcional Integral . . . . .	65
4.21	Error entre la Referencia y la Respuesta . . . . .	65
4.22	Respuesta del Sistema Controlado sometido a Perturbaciones . . . . .	66
4.23	Respuesta del Nivel de Agua Controlado sometido a Perturbaciones . . . . .	67
4.24	Acción de Control Proporcional Integral ante Perturbaciones . . . . .	67
4.25	Error entre la Referencia y la Respuesta . . . . .	68
4.26	Prueba del Controlador Neuronal de 5 Capas . . . . .	70
4.27	Respuesta del Sistema ante el Controlador Neuronal de 5 Capas . . . . .	71
4.28	Respuesta del Nivel de Agua ante el Controlador Neuronal de 5 Capas . . . . .	71
4.29	Acción de Control Neuronal de 5 Capas . . . . .	72
4.30	Respuesta del Sistema ante el Controlador Neuronal de 5 Capas sometido a Perturbaciones . . . . .	73

4.31	Respuesta del Nivel de Agua controlada ante el Controlador Neuronal de 5 Capas sometido a Perturbaciones . . . . .	73
4.32	Prueba del Controlador Neuronal de 4 Capas . . . . .	74
4.33	Respuesta del Sistema ante el Controlador Neuronal de 4 Capas . . . . .	75
4.34	Respuesta del Nivel de Agua ante el Controlador Neuronal de 4 Capas . . . . .	75
4.35	Acción de Control Neuronal ante el Controlador Neuronal de 4 Capas . . . . .	76
4.36	Respuesta del Sistema ante el Controlador Neuronal de 4 sometido a Perturbaciones . . . . .	76
4.37	Respuesta del Nivel de Agua ante el Controlador Neuronal de 4 sometido a Perturbaciones . . . . .	77
4.38	Respuesta del Sistema Controlado ante Nueva Referencia y Perturbaciones . . . . .	78
4.39	Respuesta del Nivel de Agua Controlado ante Nueva Referencia y Perturbaciones . . . . .	78
4.40	Esquema de Desarrollo de la Red Neuronal bajo Control - Logix . . . . .	80
4.41	Supervisión del Proceso en RSView32 . . . . .	81
A.1	Bloque de Control Clásico . . . . .	87
A.2	Rutina Continua para el acondicionamiento de valores enviados al controlador . . . . .	88
B.1	Manejo de los elementos del proceso y temporización de la acción de control neuronal . . . . .	89
C.1	Menu Principal del Software de Entrenamiento . . . . .	96
C.2	Definición de la Arquitectura neuronal . . . . .	97
C.3	Declaración de los Datos de Entrenamiento . . . . .	98
C.4	Proceso de Entrenamiento de la Red Neuronal . . . . .	99

# Agradecimientos

Mi más sincero agradecimiento a los Profesores Addison Ríos y Francklin Rivas, quienes con su invaluable colaboración y dedicación hicieron posible el desarrollo de este proyecto de grado.

De igual forma, deseo expresar mi gratitud a todos los profesores que participaron en mi formación académica permitiendo así la culminación de esta tesis.

Evidentemente mi familia es mi principal fuente de apoyo. Su ayuda hizo realidad esta meta.

# Resumen

Las redes neuronales artificiales son modelos matemáticos empleados en una gran variedad de aplicaciones que abarcan desde el procesamiento de señales hasta la ejecución de acciones de control complejas. Los dispositivos físicos utilizados para su implementación influyen directamente en el comportamiento de la red y su capacidad de procesamiento. Esto conlleva al uso de nuevas tecnologías para la automatización y la creación de mecanismos para ejecutar esquemas neuronales destinados al control de procesos.

En este estudio se desarrolla un control neuronal basado en un modelo existente; construyéndose una red neuronal destinada a imitar el comportamiento de un controlador clásico previamente diseñado. La técnica es implementada en el sistema de tanques Feedback Basic Process Rig 38-100 [15], comenzando con su identificación alrededor de un punto de operación y el manejo de señales reales a través del Control - Logix. Posteriormente, se diseña la ley de control convencional de acuerdo a las especificaciones de funcionamiento propuestas y las restricciones de los actuadores presentes en el proceso. Su ejecución es programada a través de bloques propios del autómeta y en función de su desempeño se recolectan datos para la construcción del controlador neuronal cuya topología se define mediante iteración sucesiva entre las posibles configuraciones. La evaluación ante los datos de prueba indican al diseñador si es conveniente o no generar el código para implementar la acción de control en el autómeta.

Además, se presenta un software diseñado en Matlab<sup>TM</sup> destinado a generar el código en texto estructurado correspondiente a la arquitectura neuronal indicada

por el usuario. El programa es utilizado dentro de RSLogix 5000 para producir la acción deseada en el Control - Logix y consecuentemente controlar el proceso, cuyos elementos y variables son modificados automáticamente a través de un sistema de supervisión desarrollado en RSView32.

**Descriptor:** Control Inteligente, Redes Neuronales, Controladores Lógicos Programables, Control de Procesos, Tecnologías para automatización

[www.bdigital.ula.ve](http://www.bdigital.ula.ve)

# Capítulo 1

## Introducción

Los sistemas de control han experimentado diversos cambios a lo largo de la historia. Desde tiempos antiguos, el hombre ha necesitado crear mecanismos que le permitan resolver problemas específicos; en la actualidad se han desarrollado numerosas técnicas y principios a través de los cuales se logra un desempeño óptimo de determinados procesos. La principal inquietud del ser humano ha sido el diseño de sistemas capaces de realizar sus tareas, de forma independiente, bajo un ambiente seguro inclusive en condiciones no aptas para el desenvolvimiento humano; es así como se inician los estudios correspondientes a redes neuronales [9] y controladores lógicos programables [14].

Las redes neuronales artificiales son sistemas inspirados en algunas funcionalidades del cerebro humano para el tratamiento de información y destinadas a ayudar al hombre en la ejecución de tareas complejas; su habilidad de aprendizaje y adaptabilidad a los cambios, las han convertido en un campo emergente dentro de los sistemas de control, siendo diseñadas, entrenadas y finalmente implementadas con la tecnología de hardware disponible. Los controladores lógicos programables (PLC) son los dispositivos ideales para este fin puesto que sus características permiten ejecutar todo tipo de acción de control donde es necesaria la interacción directa con los elementos del proceso, elevadas velocidades de procesamiento y respuesta en tiempo real [14].

La disponibilidad de tecnología de punta en autómatas programables, específicamente el Control - Logix, desarrollado por Rockwell Automation [17], propicia la implementación de una técnica de control neuronal en el equipo mediante los algoritmos y funciones básicas disponibles; permitiendo aprovechar las potencialidades matemáticas y de procesamiento del dispositivo, así como contribuir en la generación de mecanismos no construidos bajo esta plataforma para el control de un proceso específico como es el sistema de Tanques Feedback Basic Process Rig 38-100 [15].

El desarrollo del proyecto se basa en la necesidad de crear y estudiar sistemas de control con capacidad de adaptación a los cambios, y desempeño en cualquier ambiente, inclusive en medios hostiles no aptos para el desenvolvimiento humano. Por lo tanto, la investigación se orienta a la creación de un controlador neuronal y a la generación de la base de programación necesaria para implementar diferentes arquitecturas neuronales en el Control - Logix.

En el capítulo 2 y 3 se presentan los fundamentos teóricos correspondientes a controladores lógicos programables y redes neuronales artificiales; haciendo énfasis en el Control - Logix, los diferentes esquemas de control neuronal y la metodología adecuada para diseñar un controlador inteligente basado en un modelo existente. En el capítulo 4 se detalla la implementación de esta técnica en el sistema de tanques desarrollando cada uno de los pasos necesarios para llevar a cabo la acción neuronal y finalmente, el capítulo 5 se destina a exponer las conclusiones y aportes de la investigación así como las propuestas para la continuación de este trabajo.

# Capítulo 2

## Autómatas Programables

En este capítulo se presentan los conocimientos fundamentales sobre Controladores Lógicos Programables (PLC), haciendo énfasis en el Control-Logix, perteneciente a la nueva gama de autómatas de Rockwell Automation llamada Plataforma Logix y su uso dentro de los sistemas de control [18].

www.bdigital.ula.ve

### 2.1 Introducción

Los sistemas de control convencionales, basados en contactores y relés, presentan inconvenientes debido a la complejidad de los procesos industriales; estos requieren conocimientos técnicos especializados para su manejo y mantenimiento no viable desde el punto de vista de efectividad y presupuesto. Las limitaciones inherentes a estos sistemas impulsaron la creación de nuevas alternativas en el control de procesos, siendo una de ellas los autómatas programables. Estas son máquinas de tamaño reducido diseñadas para ejecutar acciones de control sobre procesos industriales secuenciales o combinatoriales en tiempo real [30], cuyo funcionamiento se basa en señales de entrada que generan respuestas de acuerdo a las instrucciones almacenadas en la memoria programable del aparato, otorgándole una gran versatilidad.

La principal desventaja de los sistemas de control antiguos es su instalación;

ante la necesidad de cambios, expansión o detección de posibles errores requerían la reconexión de todos sus elementos y por tanto la detención del proceso. Con la introducción de los autómatas programables en la industria, se minimizan los costos generados por este problema, ya que cualquier variación en el sistema de control es introducida mediante modificaciones en las instrucciones del controlador programable mientras continua operando, sin necesidad de detener los procesos. Adicionalmente, los autómatas permiten reutilizar la lógica de control creada para una aplicación cuantas veces se desee, verificar la existencia de errores en el código y efectuar las correcciones necesarias en las instrucciones antes de su puesta en marcha a través de los emuladores disponibles en el mercado.

## 2.2 Historia

Los autómatas programables fueron creados a finales de los años 60, como reemplazo para los sistemas basados en relés y contactores utilizados en empresas automotrices. En esta época, la División de Hidromática de la Corporación General Motors especificó los requerimientos de diseño del nuevo dispositivo, entre los cuales figuraban la reducción de costos, facilidad de programación y mantenimiento por parte del personal, así como la disminución del tamaño, posibilidad de operación en ambientes industriales y comunicación de datos a un sistema central para la monitorización del proceso [14].

El primer autómata programable en cumplir con todas estas especificaciones, sale al mercado en el año 1969[27]; poseía un tiempo de vida largo, permitía introducir cambios en los programas de forma sencilla sin desconexión de los dispositivos de control, y ocupaba menos espacio que los antiguos circuitos por estar constituido de relés de estado sólido. A pesar de que su programación aún era un poco tediosa, este controlador dio paso a la mejora y desarrollo de nuevas técnicas de control.

Rápidamente el uso de estos autómatas se extendió a empresas manufactureras,

apareciendo nuevas necesidades en su incorporación como la comunicación entre dos de ellos y el manejo de señales analógicas. Durante la década de los 80, los esfuerzos realizados en este campo se dirigieron a la búsqueda de la estandarización de los distintos protocolos de programación debido a la diversidad de empresas trabajando en el desarrollo de estos dispositivos, además de la búsqueda en la reducción del tamaño. Más adelante, el interés se enfocó en los lenguajes de programación utilizados, creándose protocolos de unificación internacional (IEC 61131-3) que aseguran el desarrollo de autómatas que soportan estos lenguajes estándares [14].

En la actualidad, los controladores lógicos programables no solo manejan operaciones lógicas, también son capaces de efectuar operaciones aritméticas complejas que permiten implementar distintas técnicas de control, además de contar con módulos especialmente diseñados para ejecutar acciones específicas; se han convertido en una parte fundamental de los sistemas de control distribuido por sus posibilidades de comunicación con otros autómatas y computadores.

## 2.3 Aplicaciones

Desde su creación, los controladores lógicos programables han sido utilizados en múltiples áreas de la industria para el desarrollo de tareas de control en procesos secuenciales cambiantes, en donde se cuenta con un espacio reducido, ambientes industriales o donde se requiere la monitorización centralizada de las operaciones, puesto que este es el propósito de su creación [27].

El principal atractivo para el uso de los autómatas es su modularidad y capacidad para manejar señales analógicas y digitales, lo cual permite la conexión directa a los elementos de control para la generación de señales de actuación en tiempo real con alta fiabilidad en la respuesta emitida, además, la comunicación a través de redes permite su conexión con sistemas de información y computadores personales, para aprovechar todas las características de estos potentes equipos en

control supervisorio y distribuido.

Algunos ejemplos de aplicación son la maniobra de máquinas e instalaciones de seguridad en la industria automotriz, azucarera, plantas embotelladoras, así como en señalización y control.

## 2.4 Estructura

### 2.4.1 Externa

**Compacta.** Los distintos elementos que conforman el controlador lógico programable se encuentran ubicados dentro de un mismo bloque; y son utilizados especialmente cuando el número de dispositivos de entrada y salida son relativamente pequeños.

**Modular.** Divididos en diferentes módulos que se encargan de realizar funciones específicas del sistema de control, haciendo más fácil su actualización y reparación ante fallas, mediante el reemplazo del componente implicado. Se ubican en un chasis que permite la localización compacta de los componentes. Existen dos clasificaciones de este tipo de estructura de acuerdo a la disposición de los módulos [14]:

- **Americana:** Los dispositivos de entrada y salida se encuentran separados del resto de componentes del autómata.
- **Europea:** Cada función del controlador es distribuida en módulos independientes.

## 2.4.2 Interna

La composición interna de los PLC puede clasificarse de acuerdo a tres componentes: entradas, unidad de procesamiento y salidas; cada una de las cuales se encuentran interrelacionadas para lograr el funcionamiento adecuado del dispositivo.

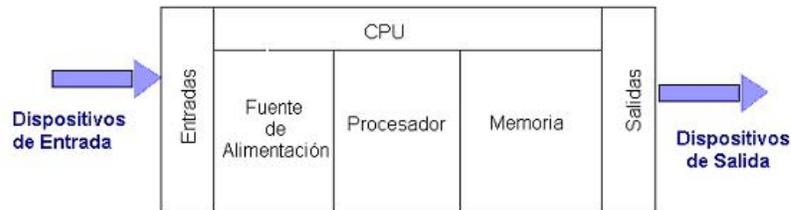


Figura 2.1: Estructura Interna Básica de un Controlador Lógico Programable

### Sección de Entradas.

Modulo encargado de recibir, adaptar y codificar las señales analógicas o digitales provenientes de los sensores, captadores, fines de carrera, entre otros. Poseen funciones encargadas de acondicionar la señal captada antes de su uso. Se considera la conexión de dos tipos de entradas, pasivas y activas. En las primeras los estados de activación cambian de acuerdo a acciones mecánicas, mientras que en las segundas el estado lógico varía según a la alimentación de una fuente de tensión externa [3].

**Sección de Salidas.** Conectadas directamente a los dispositivos de campo utilizados en el sistema de control, se encargan de efectuar la decodificación de las señales procesadas y enviarlas a los actuadores, lámparas, válvulas, relés o contactores.

**Unidad Central de Procesos.** Dirige las operaciones realizadas en el PLC, se encarga de transferir las señales introducidas por medio de los dispositivos de entrada, para su procesamiento en función de las instrucciones localizadas en el programa de usuario y enviar los resultados a los puntos de salida.

Esta unidad posee tres áreas de memoria, que almacenan los datos necesarios para el funcionamiento general del dispositivo [27].

**Memoria de programa.** Contiene la información del sistema o firmware.

**Memoria de Usuario.** Almacena el programa escrito por el usuario, utilizando memoria RAM en combinación con memoria del tipo EEPROM, cuyo borrado al fallar la fuente de alimentación es evitado mediante una batería incorporada en el autómata.

**Memoria de Datos.** Lugar donde se almacena la imagen de las entradas y salidas, temporizadores así como variables internas del controlador.

www.bdigital.ula.ve

## 2.5 Funcionamiento

Se basa en la ejecución secuencial y cíclica de las instrucciones almacenadas en memoria; este ciclo de trabajo constituido por la secuencia de lectura de las señales de entrada, su procesamiento, diagnósticos de comunicación y la generación de señales de mando, recibe el nombre de "scan".

Durante el scan, la lectura de los dispositivos de entrada y emisión de las señales de salida, son realizadas al mismo tiempo con el fin de optimizar el desempeño del controlador y agilizar el funcionamiento del PLC. La velocidad del procedimiento depende de la longitud del programa y de las instrucciones allí utilizadas, ya que cada código posee un tiempo de ejecución específico, además de influir los tiempos de retardo en la lectura de las entradas y salidas.

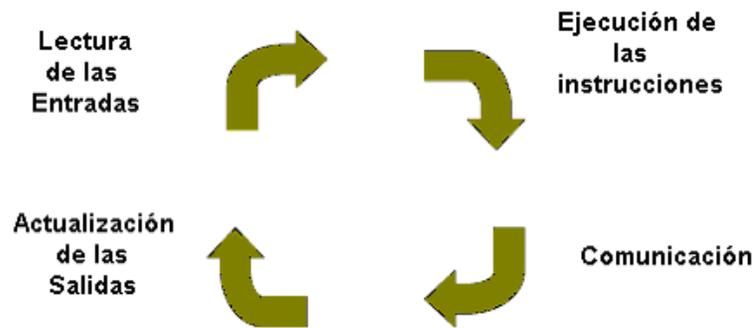


Figura 2.2: Ciclo de Scan

## 2.6 Plataforma Logix

La compañía Rockwell Automation ha desarrollado una arquitectura integrada que representa un gran avance en la tecnología de información y control [18]; la plataforma Logix forma parte de dicha arquitectura y pone a disposición de los usuarios distintos controladores manejados a través de un software común que proveen grandes ventajas en control integrado, reutilización de programas, flexibilidad de comunicación y facilidad de uso con lo cual se reducen gastos y se hace viable su uso en la industria.

Dentro de la plataforma Logix se encuentran los siguientes controladores programables: Control - Logix, CompactLogix, FlexLogix y SoftLogix; cada uno con características específicas que hacen posible la elección del autómata de acuerdo al proceso a controlar. El Control - Logix es utilizado en aplicaciones secuenciales, de movimiento o transporte, donde se requiere gran velocidad de procesamiento, mientras que para aplicaciones de propósito general de tamaño reducido se emplea el CompactLogix. En aquellos sistemas donde es necesaria la facilidad de mantenimiento y bajos costos de instalación es recomendable el uso del FlexLogix y para la integración de datos y control basado en computador se encuentra disponible el SoftLogix [18].



Figura 2.3: Controladores Lógicos Programables de la Plataforma Logix

## 2.7 Control - Logix

El Control - Logix es un autómata programable de gran velocidad de procesamiento, capacidad de memoria de 8 MegaBytes y comunicación en red a través del puerto de enlace del dispositivo que permite el control de procesos secuenciales, de transporte y movimiento a bajos costos [17]. Debido a su estructura modular, permite la adquisición de los componentes necesarios para las tareas de control que se deseen ejecutar, así como el uso de varios procesadores, elementos de comunicación y bloques de entrada/salida dentro de la misma unidad.

Las características del Control - Logix, lo convierten en el controlador ideal para sistemas donde se requiere un amplio rango de comunicación, resistencia a golpes y facilidad de mantenimiento, lo cual es posible a través de la capacidad de remoción e inserción bajo suministro de energía de los módulos, reduciéndose el tiempo necesario para reparaciones de cualquier tipo. La energía es proporcionada a los módulos a través de la fuente de alimentación situada en el chasis, que contiene un panel trasero el cual permite la comunicación a altas velocidades de los distintos componentes allí ubicados. Para la transferencia de información con otros dispositivos es utilizado el puerto de comunicaciones serial RS-232 incluido en el procesador, o distintas interfaces disponibles para la conexión a redes; por lo que varios módulos de comunicación pueden ser colocados dentro del mismo chasis para establecer un puente entre las

redes soportadas y manejar información entre ellas[17].

Múltiples beneficios son originados por la combinación de estas características, como la rapidez en la transferencia de datos, escalabilidad ante la expansión del sistema, tolerancia a las condiciones hostiles del ambiente de trabajo, fácil integración a sistemas existentes basados en controladores lógicos programables, al igual que la capacidad para cumplir con los requerimientos de muchas aplicaciones en las cuales el espacio de trabajo es limitado.

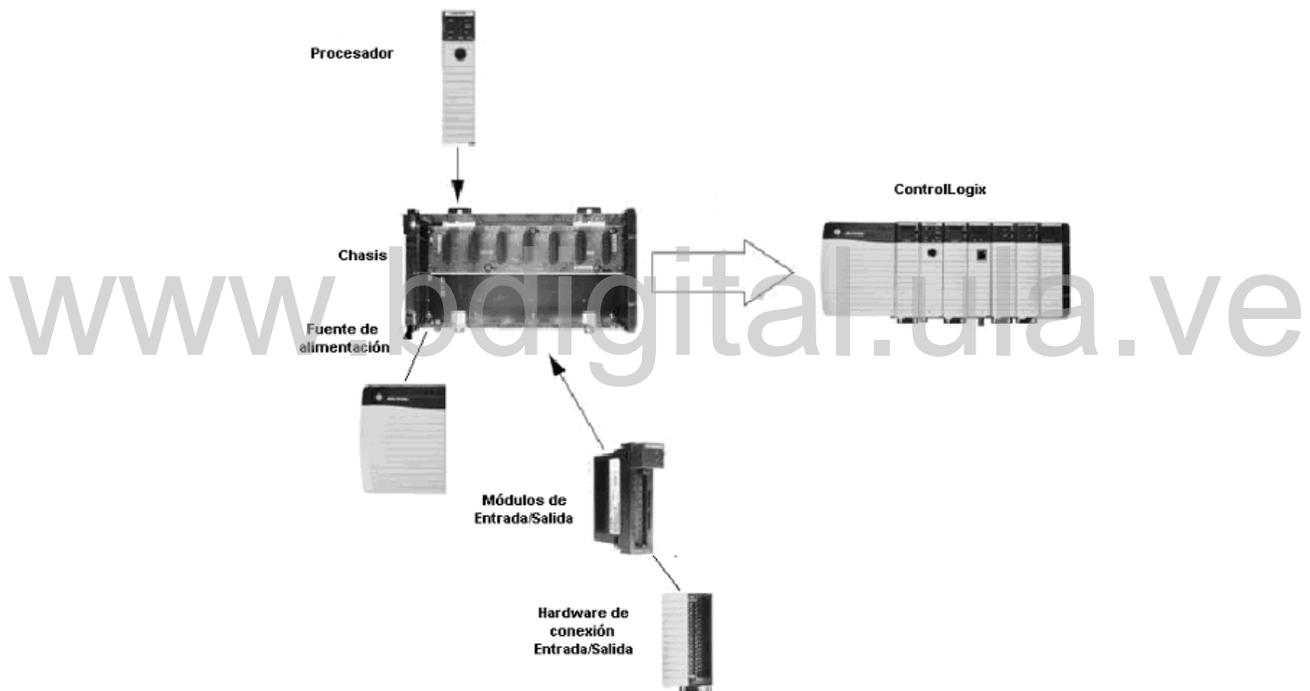


Figura 2.4: Control - Logix

Uno de los elementos que conforma el Control - Logix, es su procesador, Logix5550, altamente funcional para la ejecución de control distribuido a través de los distintos módulos de comunicación y entrada/salida; se trata de un controlador enfocado en desarrollar el conjunto de instrucciones almacenadas en el área de memoria, mientras que al estar ubicado en una de las ranuras del chasis puede monitorizar y controlar los demás elementos por medio de su comunicación con el

”backplane”.

Este controlador posee una memoria modular adaptable a las necesidades específicas del proceso; inicialmente cada procesador posee una memoria de 160 Kbytes, factible para aplicaciones distribuidas de menor escala y bajo costo, luego es posible incrementar esta cantidad de memoria en el caso de sistemas de mayor tamaño, lo que hace del Logix5550 una solución escalable para los procesos actuales [17].

En la parte frontal del controlador, se encuentra ubicada la batería de litio que permite el funcionamiento del autómata cuando falla o no existe alimentación de energía externa; allí mismo se localiza el puerto RS-232 usado como entrada para el sistema de control y puente de comunicación en la transferencia de datos entre el computador, el panel trasero del chasis y cualquier red de información.

Existen una gran variedad de módulos de entrada salida que proveen información sobre el proceso de control durante el tiempo de operación y momentos críticos. Estos son configurados a través de los paquetes de programación de la plataforma Logix y su desempeño es fácilmente monitoreado por medio de pantallas de diagnóstico para módulos locales o remotos.

### 2.7.1 Módulos de Entrada/Salida Analógicas

Ejecutan las conversiones de datos necesarias para la comunicación directa con el proceso, transforman las señales analógicas de voltaje o corriente en valores digitales para su manejo por el Control - Logix y convierte los datos digitales en valores analógicos que puedan ser enviados a las salidas con una resolución de 16 bits. Estos módulos pueden ser configurados para emitir valores específicos de corriente o voltaje en caso de fallas para brindar seguridad al proceso, adicionalmente se caracterizan por [17]:

- Datos en forma punto flotante con 32 bits de precisión o enteros de 16 bits.

- Desinstalación e instalación de los módulos mientras reciben alimentación eléctrica mediante el chasis.
- Comunicación entre los distintos módulos y dispositivos, donde, sin que se haya efectuado la recolección de datos, cada módulo genera información (productor/consumidor).
- Sello de hora continuo de datos. Indica la hora exacta incluyendo los milisegundos, en que se muestrean o envían los datos.
- Sello de hora de los datos del sistema. Indica el momento de comunicación entre el módulo y el procesador local.
- Escalado a unidades de ingeniería y generación de alarmas a nivel de software, mientras que en el hardware cada módulo posee un LED indicador del estado operativo del módulo.
- Posibilidad de calibración por canal o módulo.
- Inhabilitación de módulos en caso de configuración errónea.

Cada módulo de E/S del Control - Logix debe ser configurado como propiedad de un controlador Logix5550, que almacena la información correspondiente al comportamiento del módulo, cuyos datos son enviados al procesador durante la descarga del programa; para la conexión de los cables a los elementos de campo se utilizan bloques de terminales extraíbles o módulos de interfase. Los elementos

principales del módulo son [17].

- Conector de ControlBus, el cual conecta el módulo al panel trasero del Control - Logix.
- Pines de los conectores, realizan la conexión del módulo a la fuente de alimentación, tierra y entradas/salidas.
- Lengüeta de bloqueo, sujeta el bloque de terminales (RTB) al módulo.
- Ranuras de codificación, evitan conexiones accidentales al módulo.
- Indicadores de estado, muestran el estado de comunicación del elemento.
- Guías superior e inferior, sirven para encajar el bloque de terminales.

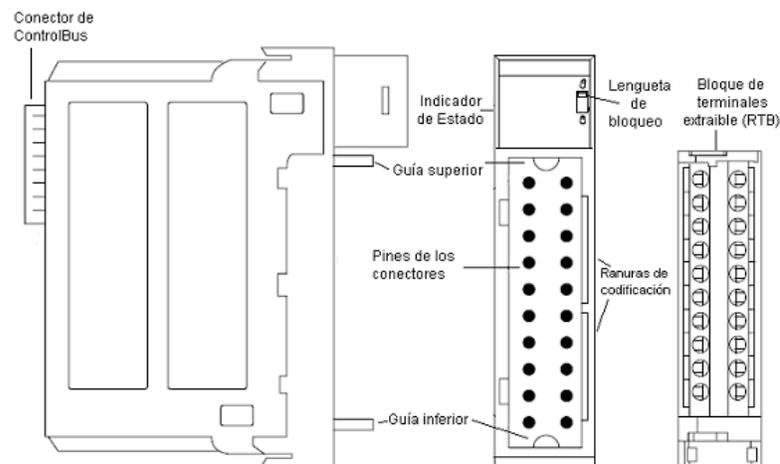


Figura 2.5: Módulo de Entrada/Salida Analógica

Los módulos de entrada/salida difunden sus datos periódicamente, el intervalo de tiempo que transcurre entre cada emisión depende de dos parámetros establecidos durante la configuración correspondientes al muestreo en tiempo real (RTS), en el caso de módulos de entrada, y al intervalo entre paquetes solicitados (RPI) para entradas y salidas. El RTS, ordena la lectura de los canales de entrada para almacenar los datos en la memoria y difundirlos al panel trasero del chasis local, mientras que el RPI solicita la difusión de los datos contenidos en la memoria al panel trasero sin previa actualización de los canales.

### **2.7.2 Módulos de Entrada/Salida Digitales**

Permiten detectar la activación o desactivación de los actuadores conectados al módulo; cuentan básicamente con las mismas características y elementos de las entradas/salidas analógicas, diferenciándose en la disposición de los indicadores de estado, ya que posee LEDs que señalan los bits del módulo en funcionamiento. La difusión de los datos depende de la configuración de los parámetros de intervalo entre paquetes solicitados (RPI) y cambio de estado (COS) solo para el modulo de entrada digital.

El intervalo entre paquetes establece el tiempo para la difusión de los datos del modulo y su configuración, mientras que el COS ordena al módulo la transferencia de información sólo en caso de producirse una transición de estado en alguno de los puntos de entrada, por lo cual es más eficaz que el RPI.

### **2.7.3 Módulo de interfase de Comunicación Ethernet**

Permite el intercambio de mensajes encapsulados dentro de protocolos estándar TCP/UDP/IP, a través de un entorno común con las interfases de comunicación ControlNet y DeviceNet y el uso de dispositivos físicos AUI o 10 Base T [16]. Al igual que el resto de módulos del Control - Logix acepta su desinstalación e instalación

mientras exista alimentación eléctrica además de no existir límite en cuanto a la cantidad de elementos de este tipo en el mismo chasis.

Su configuración es realizada por herramientas Bootp estándar o RSLinx [17], en donde no es necesario especificar la agenda de trabajo de la red ni establecer las tablas de enrutamiento de la misma, permitiendo la comunicación a través de la puerta de enlace con otros módulos en el chasis local. Se establece como un controlador de entrada/salida en tiempo real en conjunto con el procesador, actuando como un adaptador para el envío y recepción de mensajes entre el controlador y la red.

## 2.8 Lenguajes de Programación

Inicialmente las empresas dedicadas al diseño de autómatas programables, creaban lenguajes de programación propios para cada uno de sus equipos, teniendo como consecuencia una escasa flexibilidad para encontrar solución a los problemas de control; debido a la rápida aparición de un gran número de lenguajes, la Comisión Internacional Electrotécnica (IEC) [2] define una normativa de estandarización a mediados de los años 90, en su estándar IEC 61131, donde define 4 lenguajes normalizados.

- Diagrama de contactos (LD).
- Diagrama de Bloques funcionales (FBD).
- Texto Estructurado (ST).
- Lista de Instrucciones (IL).

La estandarización ha permitido la reducción de gastos en recursos humanos, formación y mantenimiento, así como el incremento de la flexibilidad en la comunicación con otros dispositivos de distinto fabricantes, quedando la elección del lenguaje de programación bajo el criterio del programador de acuerdo a la estructura del sistema,

su interrelación con otros procesos o características especiales del problema.

### 2.8.1 Diagrama de Contactos

Constituido por un conjunto de instrucciones gráficas similares a los símbolos utilizados en los sistemas de control basados en relés; posee dos barras verticales correspondientes a la tensión (izquierda) y tierra (derecha), conectadas por renglones en los cuales se ubican los distintos símbolos correspondientes a variables lógicas de estado falso o verdadero.

Las instrucciones son ejecutadas en forma secuencial de izquierda a derecha, comenzando por el primer renglón del diagrama hasta llegar al último.

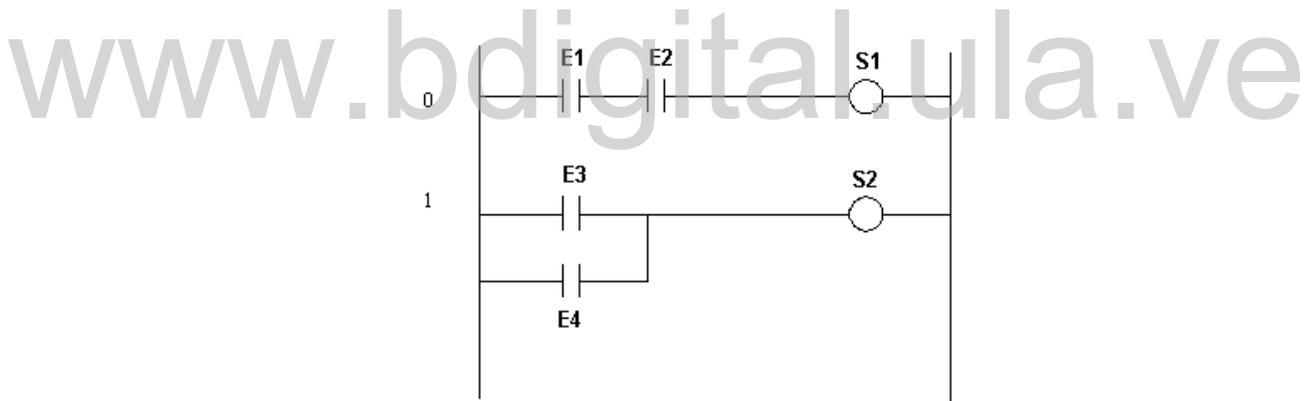


Figura 2.6: Diagrama Escalera

### 2.8.2 Funciones de Diagramas de Bloque

Utilizados frecuentemente en el control de procesos, debido al fácil manejo de los elementos como bloques gráficos conectados entre sí en forma similar a circuitos electrónicos.

### 2.8.3 Texto Estructurado

Lenguaje de alto nivel con sintaxis similar a Pascal, es utilizado para desarrollar expresiones complejas con un amplio rango en el manejo de datos, además posee instrucciones para ejecutar lazos de repetición, estructuras condicionales y funciones aritméticas.

### 2.8.4 Listado de instrucciones

Lenguaje de bajo nivel, basado en funciones de álgebra lógica en donde solo es permitida una operación por línea y cuyo manejo puede ser tedioso para grandes aplicaciones por ser similar al lenguaje ensamblador; cada instrucción consta de un componente mnemónico de operación y una dirección que identifica elementos particulares dentro del controlador.



Operación	Operando
OUT	O:003

Figura 2.7: Listado de Instrucciones

## 2.9 Software de Programación

La plataforma Logix de Rockwell Automation trabaja con un software de programación común denominado RSLogix 5000 [19], el cual brinda herramientas para la programación de aplicaciones a través de lenguajes estándares, estableciendo un ambiente de programación que eleva la productividad, ofrece facilidad de mantenimiento y brinda estabilidad al sistema; conforme a la normativa IEC-61131-3 dispone editores con estas características para el diseño de rutinas en diagrama escalera, texto estructurado, diagrama de bloques y gráficos de funciones secuenciales.

El paquete RSLogix posee herramientas prácticas para desarrollo de programas utilizando un editor simple, que permite modificar varias instrucciones lógicas simultáneamente, e inclusive insertar código arrastrando directamente los iconos al espacio de trabajo; tolera el direccionamiento simbólico y el establecimiento de ámbitos de validez para las variables, lo que favorece la manipulación continua de las rutinas, además de facilitar la conversión de programas existentes escritos en software para PLC-5 al formato del RSLogix 5000.

La plataforma Logix hace uso de otros paquetes como RSLinx o algunos componentes de RSView32 [22] para establecer la ruta de comunicación con el dispositivo y generar gráficos de desempeño con los datos en tiempo real de las variables declaradas en el programa, las cuales pueden ser importadas y/o exportadas a través de otras aplicaciones como Microsoft Excel, para su reutilización en otros sistemas de control.

RSLinx es el servidor de comunicación que establece la conectividad entre los dispositivos de campo y el resto de aplicaciones de programación de Rockwell; este software emplea diversas interfases o mecanismos para la recolección y análisis de datos, visualizados fácilmente a través del ambiente gráfico que despliega el estado de la red, y de los distintos componentes conectados a ella, incluyendo los módulos pertenecientes al Control - Logix.

El siguiente paquete utilizado por esta plataforma es RSView32, un software diseñado para crear aplicaciones de control supervisorio destinadas a monitorizar y controlar máquinas y procesos de automatización [22]; se caracteriza por la interacción directa con otros productos de Rockwell, registro de datos en Microsoft Excel, actualización de proyectos en funcionamiento sin necesidad de apagar el proceso e intercambio de datos en tiempo real con otros dispositivos o aplicaciones. Este paquete brinda una vista organizada del proyecto mientras se crean las animaciones gráficas, histogramas del proceso, registro de alarmas y eventos, lo que facilita el trabajo durante su edición y ejecución.

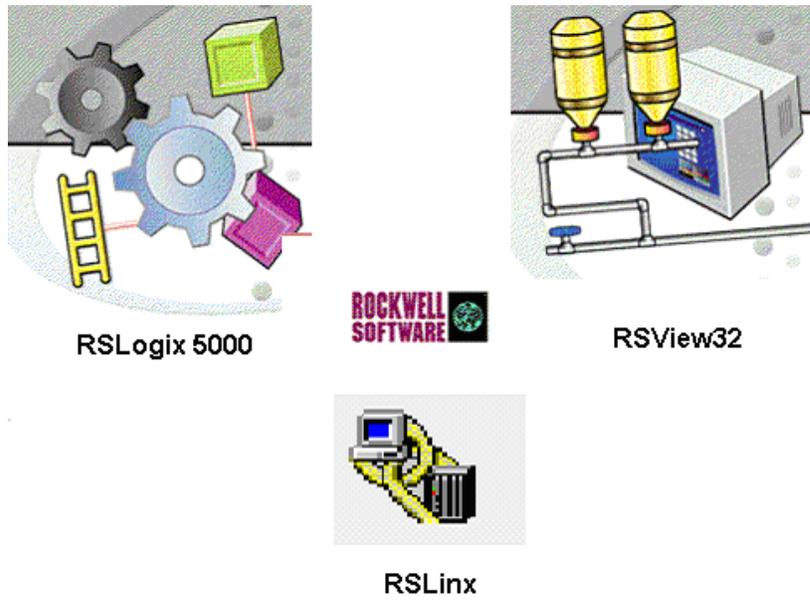


Figura 2.8: Software de Programación para la plataforma Logix

[www.bdigital.ula.ve](http://www.bdigital.ula.ve)

# Capítulo 3

## Redes Neuronales

En este capítulo se describen los conceptos relacionados con las redes neuronales artificiales, su relación con las neuronas biológicas, funcionamiento, capacidad y aplicación en distintos campos de la industria, especialmente dentro del control de procesos. Se definen las diversas clasificaciones de los esquemas de control neuronal haciendo énfasis en la estructura basada en el modelo de un controlador existente.

### 3.1 Introducción

Las redes neuronales artificiales se encuentran inspiradas en el funcionamiento de las redes neuronales biológicas; con ellas se pretende resolver problemas complejos que no pueden ser tratados mediante algoritmos secuenciales, e inclusive encontrar soluciones óptimas a situaciones donde anteriormente se requería el desarrollo de algoritmos sofisticados para cumplir parcialmente con los objetivos planteados.

Aunque han sido enormes los avances en el campo de la inteligencia artificial con respecto al desarrollo de sistemas basados en redes neuronales artificiales, no se ha logrado igualar el aprendizaje y procesamiento de la información efectuado por las redes biológicas. Solo en las operaciones aritméticas simples, el computador es más

rápido que el cerebro, sin embargo, algunas tareas simples procesadas en cuestión de segundos por este órgano, pueden necesitar días de cómputo para su culminación inclusive en la supercomputadora más veloz existente. Actualmente, se han logrado emular a cierta escala algunas características de las redes neuronales biológicas mediante computadores, entre ellos la adaptabilidad, aceptación de fallas, capacidad de aprendizaje y procesamiento paralelo de la información lo cual ha permitido su uso en el modelo e identificación de sistemas, reconocimiento de patrones, predicción y control de procesos, entre otros.

## 3.2 Redes Neuronales Biológicas

El cerebro humano es un órgano complejo, capaz de procesar simultáneamente grandes cantidades de datos a través de los billones de células que le conforman. Se encuentra compuesto por aproximadamente 100.000 millones de neuronas conectadas masivamente entre sí formando una red en la cual se distribuye la información por medio de procesos bioquímicos [8].

La neurona es una célula biológica anatómica caracterizada por su incapacidad de reproducción y encargada de recibir, procesar y emitir datos provenientes de otras neuronas o estímulos externos, así como de generar impulsos nerviosos. En su estructura se encuentran presentes diversas ramificaciones conocidas como dendritas que parten desde el núcleo de la neurona (cuerpo celular o soma) y sirven como medio de entrada de la información hacia la célula durante el proceso de comunicación. Estos datos son conducidos al núcleo para ser almacenados produciéndose una acumulación de energía en función de la cual la neurona pasa a un estado excitatorio; si la neurona es activada, se genera una señal eléctrica que es transportada a la terminación cilíndrica ubicada al final de cada una de estas células, llamada axón.

La información ubicada en el axón de una neurona es transmitida a las dendritas de la siguiente célula, a través del proceso de sinapsis basado en la liberación

de partículas denominadas neurotransmisores; esta operación oscila en el orden de los milisegundos, lo cual es relativamente lento al hacer una comparación con la velocidad de procesamiento de los computadores actuales, sin embargo la potencialidad de las redes neuronales biológicas se encuentra en su capacidad de procesamiento paralelo y distribuido de las operaciones, lo que le permite ejecutar la mayoría de las actividades con mayor velocidad que cualquier computador digital.

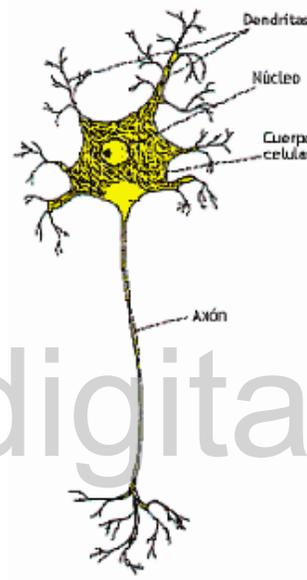


Figura 3.1: Neurona Biológica

### 3.3 Redes Neuronales Artificiales

Inspiradas en el desempeño del modelo biológico, las redes neuronales artificiales son un conjunto de representaciones matemáticas operativas en paralelo mediante la unión de una serie de elementos que pretenden emular a pequeña escala el comportamiento de las neuronas presentes en el cerebro humano; por lo tanto, poseen características similares a las redes biológicas como la capacidad de aprendizaje, generalización y abstracción, lo cual ha permitido su uso en el reconocimiento de patrones, procesamiento de señales, control de sistemas y en general en cualquier problema que

no cuenta con un algoritmo plenamente definido para cumplir con las metas impuestas.

Su estructura se encuentra definida en función de los elementos que componen a la neurona biológica, existiendo una relación directa entre las unidades orgánicas y artificiales. Cada red neuronal artificial posee un conjunto de elementos de procesamiento sencillos interconectados entre sí conformados por varias entradas, una salida y alguna función de transferencia que establece el estado de la neurona de acuerdo al resultado equivalente al proceso sináptico biológico dado por el peso de las conexiones entre cada neurona y sus entradas provenientes de estímulos externos u otras neuronas, según el cual el grado de influencia de cada entrada es ponderado para determinar la activación o inhibición de la neurona. Luego de la integración de las entradas con sus respectivos pesos se produce la generación de la salida de la neurona a través de la aplicación de la función de activación a la señal acumulada; esta salida es transmitida a otra neurona siguiendo el mismo procedimiento.

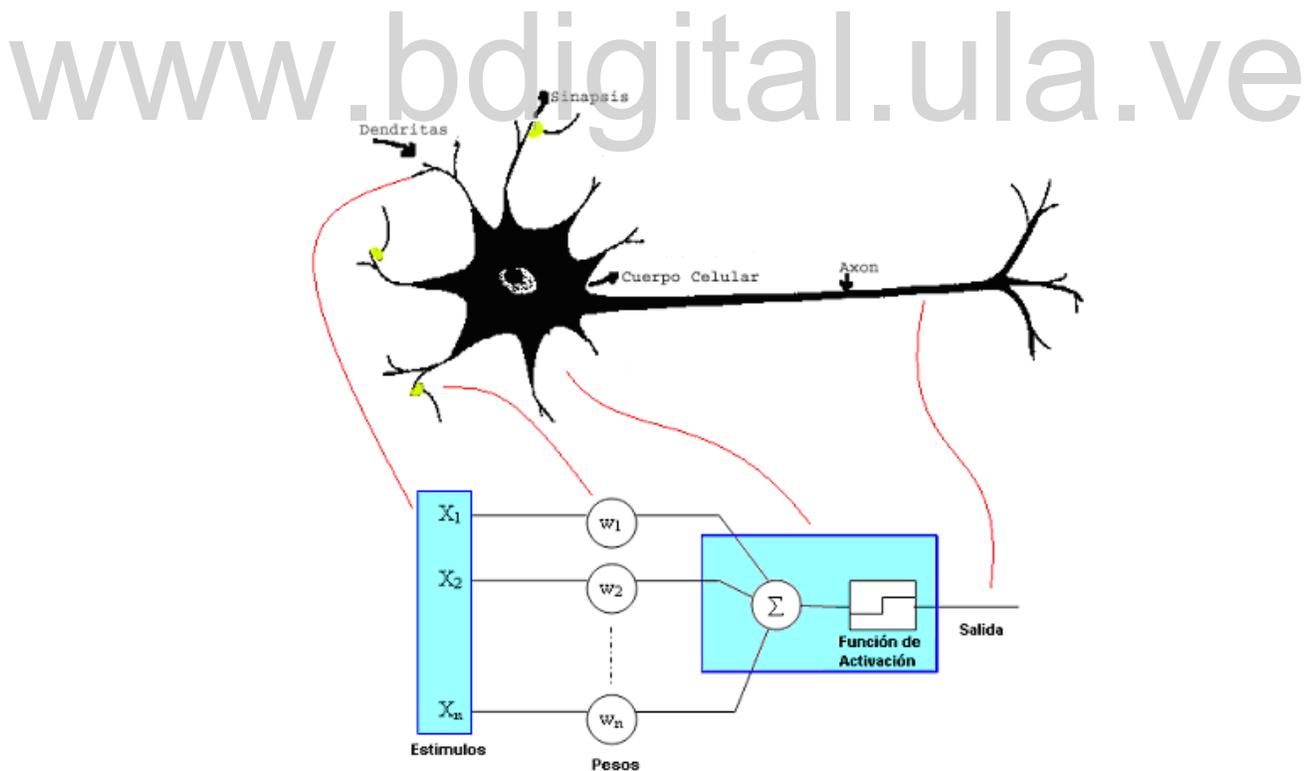


Figura 3.2: Relación entre las neuronas biológicas y artificiales

El vínculo directo entre los componentes principales de la neurona biológica y artificial, así como la similitud entre sus procesos para la transmisión de información, favorecen la emulación de múltiples características propias de la célula, como:

- Adaptación a los cambios basado en aprendizajes iniciales, generando nuevos comportamientos ante situaciones desconocidas.
- Generalización de la información de entrada cuando no es muy clara o está incompleta.
- Tolerancia a fallos en los datos, reconociendo las señales de ruido presentes, además de soportar errores en algunas neuronas sin influir significativamente la respuesta de la red debido a la disponibilidad de los datos necesarios para trabajar en las demás unidades.
- Paralelismo, logrado mediante el trabajo conjunto de los elementos de procesamiento (neuronas), donde se distribuye la información presente en el sistema.

Como consecuencia de estas características, las redes neuronales artificiales operan en entornos de tiempo real y son fácilmente implementadas mediante software o hardware dentro de aplicaciones existentes, por su facilidad de entrenamiento y comprobación, lo cual las hace aplicables a un gran número de aplicaciones incluyendo el control de procesos.

### 3.3.1 Antecedentes históricos

Desde la antigüedad el hombre ha presentado un marcado interés por el pensamiento humano y la manera como éste se produce en el sistema biológico. En el año 1936,

Alan Turing propone al cerebro como modelo computacional [5], lo que impulsa la creación mediante circuitos eléctricos del primer prototipo de una neurona artificial en 1943 por el neurofisiólogo Warren S. McCulloch y el matemático Walter Pitts, quienes además establecieron las bases para la computación neuronal en su libro "Un Cálculo Lógico de la Inminente Idea de la Actividad Nerviosa" [28]; en dicha publicación se demostró la posibilidad de construir una red capaz de resolver problemas utilizando cálculos matemáticos y diferentes clases de algoritmos, comenzando de tal forma la era del interés real por la inteligencia artificial.

Una nueva publicación aparece en 1949 para introducir el concepto de entrenamiento de las neuronas; el libro escrito por Donald Hebb y de nombre "La organización del comportamiento" [4], trató el tema a nivel psicológico y explicaba los cambios ocurridos a partir del aprendizaje mediante el reforzamiento o debilitamiento de las conexiones entre las neuronas biológicas. Estas teorías constituyeron la base de las Redes Neuronales y por ello son implementadas en la mayoría de los algoritmos de adiestramiento empleados actualmente.

Más adelante se realizaron toda una serie de experimentos que permitieron determinar el manejo distribuido de la información en el cerebro y unificar el trabajo multidisciplinario entre grupos de especialistas dedicados a crear simulaciones computacionales, entre los cuales se encuentra Frank Rosenblatt quien diseñó y desarrolló la red neuronal más antigua, el perceptrón durante el año 1957; un modelo de neurona capaz de procesar varias entradas, producir una salida y generalizar comportamientos de su entrenamiento para adaptarse a nuevas situaciones [21]. Unos años mas tarde aparece el modelo de elementos lineales adaptativo (Adaline) desarrollado por Widroff y Hoffsin, aplicado como filtro para eliminar ecos en las líneas telefónica; sin embargo, los científicos Marvin Minsky y Seymour Papert se encargaron al mismo tiempo de demostrar matemáticamente los puntos débiles del perceptrón, como la incapacidad para manejar operaciones no lineales relativamente sencillas y ciertas deficiencias en el crecimiento exponencial del tiempo necesario para el aprendizaje al aumentar el numero de entradas a la red [11].

Las críticas realizadas provocaron una pérdida de interés en el campo y retraso en las investigaciones, siendo en 1974 cuando se diseña el algoritmo de retropropagación para permitir la aplicación de las redes neuronales en problemas de mayor escala y solventar una parte de la problemática planteada; este algoritmo en conjunto con el modelo de redes neuronales asociativas presentado por John Hopfield [6], lograron despertar nuevamente el interés en el campo de la inteligencia artificial.

En la actualidad son innumerables las investigaciones sobre redes neuronales que se realizan y muchas las aplicaciones que se le han dado a las mismas; en el control de sistemas específicamente para el modelado permiten incluir dentro de los patrones no linealidades, perturbaciones, cambios en el sistema así como parámetros no tomados en cuenta durante el entrenamiento inicial, con lo cual se logra un modelado más efectivo y por ende un control adecuado de la planta en cuestión. Su campo de aplicación se extiende al diseño de controladores para procesos, dentro de los cuales se puede mencionar el control neuronal imitador de alguna técnica previamente diseñada o el control predictivo.

### 3.3.2 Modelo matemático

La salida de una neurona artificial  $y_k$ , viene dada por la siguiente ecuación matemática:

$$y_k = \Gamma(z) \quad (3.1)$$

$$z = \sum_{i=1}^n \omega_i x_i + \omega_{n+1} b \quad (3.2)$$

En donde, la variable  $x_i$  representa las señales de entrada,  $w_i$  a los pesos sinápticos de la neurona, y  $b$  indica el umbral, sesgo neuronal o predisposición a la inhibición que posee la neurona. Para simplificar las ecuaciones matemáticas, se suelen

utilizar expresiones vectoriales en la representación del modelo de las redes neuronales.

$$Y = \Gamma(Z) \quad (3.3)$$

$$Z = W^T X \quad (3.4)$$

$$X = \begin{bmatrix} x_1 \\ \vdots \\ x_n \\ b \end{bmatrix}, \quad (3.5)$$

$$W = \begin{bmatrix} w_1 \\ \vdots \\ w_n \\ w_{n+1} \end{bmatrix} \quad (3.6)$$

La función matemática aplicada al resultado de la sumatoria de las entradas por sus pesos sinápticos correspondientes, recibe el nombre de función de activación y se encarga de calcular el estado excitatorio o inhibitorio de la neurona; debido a que los algoritmos de entrenamiento se basan en la minimización del error entre la respuesta deseada y la obtenida, se utilizan funciones de activación que permitan el manejo de derivadas. Las más utilizadas son la función sigmoide unipolar, sigmoide bipolar, tangente hiperbólica y lineal, cuyas expresiones matemáticas son:

Función Sigmoide Unipolar.

$$f(x) = \frac{1}{1+e^{-\lambda x}} \Rightarrow \lambda = \text{coeficiente de aprendizaje} \quad (3.7)$$

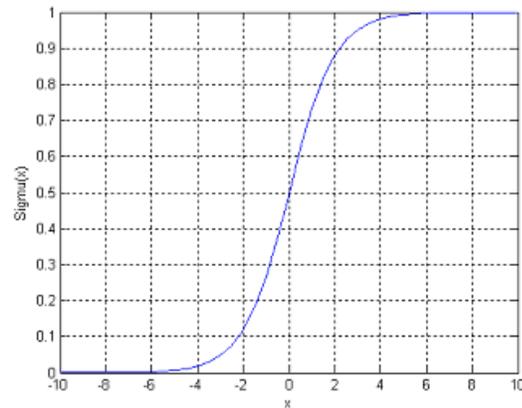


Figura 3.3: Función Sigmoide Unipolar

Función Sigmoide Bipolar.

$$f(x) = \frac{2}{1+e^{-\lambda x}} - 1 \Rightarrow \lambda = \text{coeficiente de aprendizaje} \quad (3.8)$$

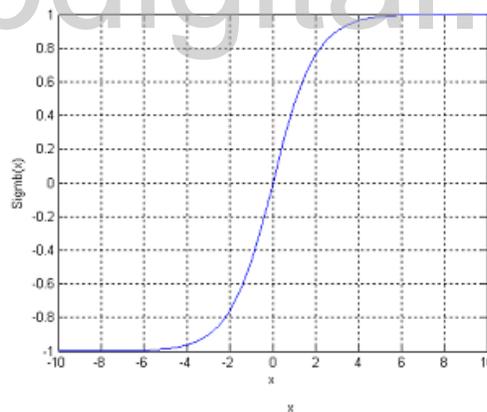


Figura 3.4: Función Sigmoide Bipolar

Función Tangente Hiperbólica.

$$f(x) = \tanh(x) \quad (3.9)$$

$$f(x) = \frac{2}{1+e^{-2x}} - 1 \quad (3.10)$$

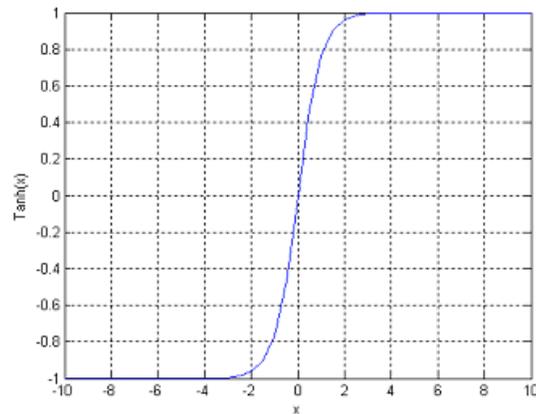


Figura 3.5: Función Tangente Hiperbólica

Función Lineal.

$$f(x) = x \quad (3.11)$$

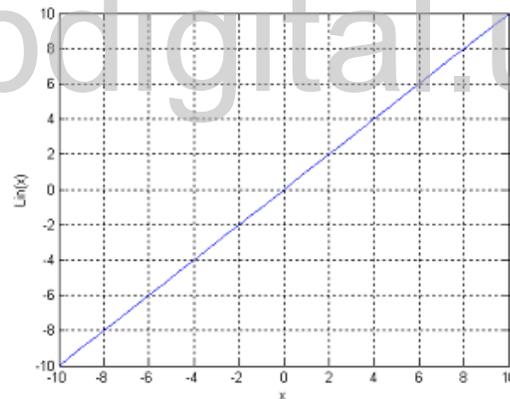


Figura 3.6: Función Lineal

## 3.4 Clasificación de los Esquemas Neuronales Artificiales

La disposición de los elementos de proceso en una Red Neuronal y la estructura de las conexiones sinápticas constituyen los parámetros fundamentales para la clasificación

de diversos tipos de arquitecturas neuronales.

### 3.4.1 Según su Topología

Las Redes Neuronales se organizan en capas, con un determinado número de unidades de procesamiento por bloque con un comportamiento similar; se distinguen tres tipos, entre capas de entrada, salidas y ocultas, siendo las últimas del tipo interno. La cantidad de niveles determinan la clasificación topológica en:

#### Redes Monocapa

Poseen una sola capa de neuronas en su estructura, destinadas a recibir señales del exterior y emitir la respuesta de la red simultáneamente. En este tipo de red, las conexiones entre las diferentes neuronas son laterales, cruzadas o autorrecurrentes y son utilizadas principalmente para regenerar información de entrada incompleta o con señales de ruido.

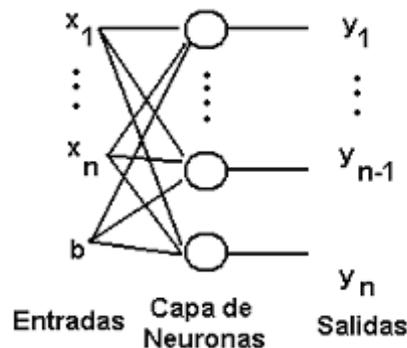


Figura 3.7: Red Monocapa

## Redes Multicapa

La red se encuentra conformada por varios niveles de redes monocapa dispuestas en cascada, donde la entrada de una capa es la salida de otra; en este tipo de redes existe la presencia de por lo menos una capa oculta entre los niveles de entrada y salida de la misma.

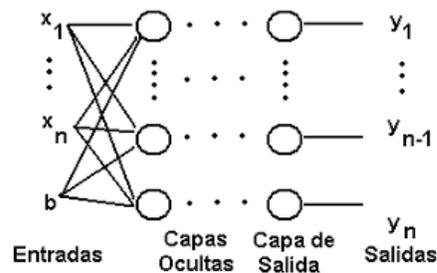


Figura 3.8: Red Multicapa

### 3.4.2 Según el Patrón de Conexión

#### Redes Unidireccionales

El flujo de los datos se realiza en una sola dirección sin posibilidad de retroalimentación, la salida de una neurona es la entrada de la capa siguiente por tanto no existen conexiones entre la salida de una capa y la entrada de la misma o de niveles previos.

#### Redes Recurrentes

Contienen lazos de realimentación, con lo cual adquieren un aspecto dinámico que les permiten evolucionar hacia nuevos estados de equilibrio.

### 3.4.3 Según la Naturaleza de las Señales de Entrada

Se distinguen tres clasificaciones principales, entre redes analógicas, digitales e híbridas; las primeras se encargan de procesar datos continuos y producir salidas del mismo tipo, acotadas en el intervalo  $[-1, 1]$  o  $[0, 1]$  empleando alguna función de activación continua, mientras que las redes binarias trabajan sólo con entradas y salidas discretas  $(0, 1)$  y las redes híbridas procesan entradas analógicas para producir respuestas digitales.

## 3.5 Proceso de Aprendizaje

Definido como la capacidad de modificar comportamientos y adquirir nuevas habilidades a partir de experiencias  $[x]$ ; el aprendizaje se basa en la excitación o inhibición de las conexiones sinápticas existentes entre las distintas neuronas que conforman el cerebro. En términos de redes neuronales artificiales, este proceso consiste en la actualización de la arquitectura de la red y pesos de las conexiones hasta conseguir la estructura que permita producir la salida deseada ante algún conjunto de entradas.

Los métodos utilizados para establecer la fuerza de las conexiones presentes en la red, se denominan algoritmos de aprendizaje; constituyen problemas de optimización puesto que su objetivo es encontrar los pesos que minimicen el error entre la salida esperada y la obtenida [24]. De acuerdo a la existencia de un ente externo encargado del entrenamiento, el proceso se divide en aprendizaje supervisado y no supervisado.

### 3.5.1 Aprendizaje Supervisado

Durante el proceso de entrenamiento se encuentran disponibles una serie de patrones entrada/salida que le indican a la red cual debería ser su comportamiento ante determinados estímulos, y a su vez permiten modificar los pesos correspondientes a las

conexiones sinápticas para encontrar aquellos que minimicen el error existente entre la respuesta emitida y el estado de salida deseado.

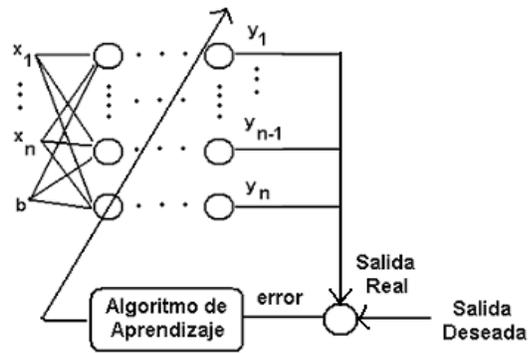


Figura 3.9: Aprendizaje Supervisado

Este tipo de aprendizaje es similar al impartido por un maestro encargado de indicar a la red las respuestas correctas que debe emitir ante ciertos estímulos desconocidos por la misma; el proceso se realiza de manera iterativa hasta que se logra un comportamiento similar al indicado por el experto, momento en el cual concluye el proceso de entrenamiento y la red se encuentra lista para ser expuesta a nuevos estímulos ante los cuales podrá generar una respuesta adecuada si el entrenamiento se ejecutó de forma óptima. Se divide en tres categorías.

### Por Corrección de Error

La modificación del peso de las conexiones neuronales se basa en la disminución gradual del error existente entre la respuesta obtenida y el patrón de salida deseado. En este algoritmo, la red es sometida a un estímulo y el comportamiento adoptado es comparado con el deseado, la diferencia es empleada junto con el valor anterior de la conexión para calcular el nuevo peso, lo cual permite el reforzar o inhibir la conexión.

### **Por Refuerzo**

En este caso, no existe disponibilidad de los patrones de salida deseados de la red ante determinados estímulos; el ente de supervisión conoce el comportamiento general deseado y en función del mismo se encarga de emitir señales de éxito o fracaso a partir de las respuestas generadas por la red, con el fin de reforzar o inhibir las conexiones sinápticas a través de la variación de sus pesos siguiendo mecanismos probabilísticos.

### **Estocástico**

Este tipo de aprendizaje establece una analogía entre la red neuronal y algún objeto de estado sólido, siendo el objetivo primordial disminuir la cantidad de energía almacenada en el mismo, la cual indica el grado de estabilidad de la red y su proximidad a un comportamiento deseado. Se basa en el planteamiento de variaciones en los pesos de las conexiones y la evaluación de sus efectos en función de la cantidad de energía, donde una disminución implica la cercanía al valor deseado y por ende la aceptación del cambio, mientras que un incremento conlleva a la variación de los pesos basada en funciones probabilísticas.

## **3.5.2 Aprendizaje No Supervisado**

En el aprendizaje no supervisado no existe conocimiento alguno sobre el comportamiento deseado de la red ante los estímulos a los cuales es sometida; es decir no existen patrones de entrenamiento o algún tipo de información externa que le indique a la red como debería efectuar el ajuste de las conexiones sinápticas. Debido a la ausencia de supervisión externa, la red se autoorganiza realizando un estudio exhaustivo de los datos de entrada suministrados, basado en las características, regularidades y categorización de los mismos. Generalmente este tipo de aprendizaje se utiliza en redes cuyo propósito es encontrar analogías entre datos actuales y pasados así como generar versiones codificadas de la entrada que solo mantengan los aspectos relevantes de la información suministrada [24]. Adicionalmente suelen emplearse para

clasificar la información de entrada de la red y obtener los prototipos de las clases creadas.

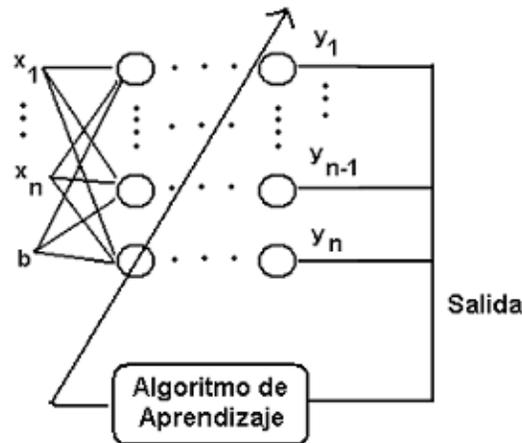


Figura 3.10: Aprendizaje No Supervisado

www.bdigital.ula.ve

### Hebbiano

Este es el postulado de aprendizaje más antiguo, planteado por Donald Hebb en el año 1949 en su libro *La organización del Comportamiento*, explica que la conexión sináptica de dos neuronas será reforzada si las neuronas implicadas se activan simultáneamente, de lo contrario el enlace se debilita.

### Competitivo

Las neuronas que conforman la red cooperan entre sí para realizar una tarea específica, compiten entre sí por activarse, la neurona con mayor actividad es excitada mientras que las demás se inhiben.

## 3.6 Algoritmo de Entrenamiento de Retropropagación

Para el entrenamiento de redes neuronales multicapa, generalmente se emplea el algoritmo de retropropagación del error, el cual consiste en minimizar el error cuadrático medio existente entre la respuesta deseada y la salida generada por la red ante un determinado patrón de entrenamiento [24]. El modo de operación de este algoritmo, consiste en la inicialización aleatoria de los pesos de las conexiones para calcular la salida de la red ante la muestra de entrada y su comparación con el valor deseado; dicha relación genera un error, usado para ajustar los pesos de cada neurona ubicada en la capa de salida.

Este error es propagado hacia las capas anteriores para ser utilizado por métodos iterativos basados en derivadas para actualizar los pesos de las capas intermedias; el proceso se detiene al obtener aquellos pesos que permitan alcanzar un error menor al deseado para todos los patrones de entrenamiento presentados a la red. Debido al uso de todos los patrones de entrenamiento, el algoritmo debe ejecutar largos periodos de entrenamiento para lograr los parámetros óptimos lo cual constituye una desventaja del mismo; además existe la posibilidad de generar redes no deseadas en caso de encontrar mínimos locales que indiquen el final del proceso de entrenamiento cuando en realidad el objetivo aún no ha sido alcanzado. Estos problemas suelen ser solventados aplicando métodos particulares para el ajuste de las conexiones neuronales e incrementando el número de capas ocultas de la red lo cual disminuye la posibilidad de encontrar mínimos locales.

### 3.6.1 Regla Delta Generalizada

La regla delta propuesta en el año 1960 por Bernard Widrow [29], es una regla de entrenamiento sencilla que propone el ajuste de los pesos sinápticos de acuerdo a la diferencia entre la salida deseada y la respuesta obtenida, mediante el uso del

método de descenso por gradiente; a pesar de ser el caso más sencillo del algoritmo de retropropagación, esta regla trabaja solo con arreglos neuronales continuos de una capa [20], por lo cual se creó una generalización funcional para redes de varias capas y funciones de activación derivables llamada regla delta generalizada.

El algoritmo funciona con la inicialización aleatoria de los pesos de la red neuronal para observar la salida producida ante algún patrón de entrenamiento.

$$X = \begin{bmatrix} x_1 \\ \vdots \\ x_n \\ b \end{bmatrix}, \quad (3.12)$$

$$W = \begin{bmatrix} w_1 \\ \vdots \\ w_n \\ w_{n+1} \end{bmatrix} \quad (3.13)$$

$$Y = \Gamma(Z) \quad (3.14)$$

$$Z = W^T X \quad (3.15)$$

A partir del primer valor obtenido, el algoritmo comienza a iterar, tomando como función de costo a minimizar el error entre el valor producido y el especificado en el patrón de entrenamiento; el cual viene dado por la siguiente ecuación matemática.

$$J(E_p) = \frac{1}{2} \sum_{j=1}^n (Y_d - Y_k)^2 \quad (3.16)$$

$$J(E_p) = \frac{1}{2} \|(Y_d - Y_k)\|^2 \quad (3.17)$$

La variación en los pesos de la red neuronal se escoge proporcionalmente al valor del gradiente descendiente,

$$\Delta\omega_{ji} = -\alpha \frac{\partial J(E_p)}{\partial \omega_j} \quad (3.18)$$

En donde  $\alpha$  corresponde al coeficiente de aprendizaje, posteriormente se aplica la regla de la cadena para propagar un porcentaje del error a las capas anteriores y efectuar un cálculo similar que permite variar los pesos de estas capas.

## 3.7 Control Neuronal

Actualmente, las redes neuronales son empleadas en una gran variedad de aplicaciones que requieren cálculos matemáticos complejos, tales como reconocimiento de voz, imágenes, inferencia, clasificación de patrones, filtrado de señales y control de procesos, entre otros [1]. La existencia de no linealidades, variación de parámetros en el tiempo, saturación en los actuadores y otros problemas incentivaron la aplicación de las redes neuronales en el control de sistemas, diseñándose modelos neuronales que permiten la identificación de sistemas e implementación de distintos esquemas de control neuronal capaces de generalizar patrones de comportamiento ante la ausencia del modelo matemático de la planta o presencia de variaciones importantes en los parámetros de la misma; además las altas velocidades de procesamiento alcanzadas por las redes neuronales las convierte en una alternativa para el desempeño de leyes de control en procesos complejos [25].

### 3.7.1 Esquemas de control neuronal

Han sido numerosas las aplicaciones dadas a las redes neuronales dentro del control de sistemas, por ellos se han creado clasificaciones de acuerdo a la forma como se implementa la red neuronal o el papel que cumple la misma dentro del proceso. Los esquemas de control existentes se distribuyen según las siguientes categorías [1].

- Control neuronal indirecto
- Control neuronal directo

- Control adaptativo para sistemas no lineales
- Otros esquemas de control neuronal

En cada una de estas clasificaciones la red neuronal se diferencia por su actuación dentro del sistema; los esquemas de control neuronal indirecto como su nombre lo indica actúan indirectamente sobre el proceso, es decir, no envían la señal de control al mismo sino se desempeñan como indicadores del comportamiento del sistema para producir referencias a un controlador determinado, mientras que los controladores neuronales directos se encargan de generar la señal de control para enviarla directamente al proceso. Por su parte existen otros esquemas de control neuronal caracterizados por incluir técnicas adaptativas y predictivas. Cada una de estas clasificaciones incluye un gran número de controladores, de los cuales se tomó para la realización de este trabajo el control neuronal directo basado en el modelo de un controlador.

www.bdigital.ula.ve

### **Control Neuronal basado en el modelo de un controlador**

Pertenciente al esquema de control neuronal directo; este es el modelo de control neuronal existente más simple, se dedica a modelar el comportamiento de algún controlador previamente diseñado y emitir directamente la señal de control sobre el proceso[1]. Puesto que su función es imitar el desempeño de un controlador existente, su entrenamiento es realizado en función de las señales de entrada y salida correspondientes a la ley de control previamente definida.

El proceso de diseño de este tipo de controladores consiste en la siguiente serie de pasos [25]:

1. Identificación de la planta o proceso que se desea controlar para conocer la estructura de la misma.
2. Diseño del controlador deseado para satisfacer los requerimientos de

funcionamiento del proceso.

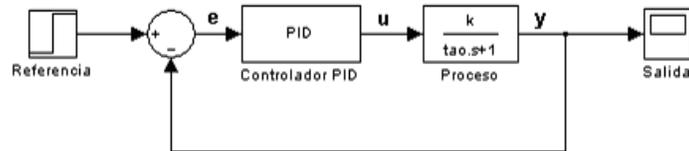


Figura 3.11: Esquema de Control Clásico

- Adquisición de los datos del controlador, poniendo en funcionamiento el controlador directamente en el proceso y obteniendo la señal de entrada con respecto a la señal de salida para obtener los datos de entrenamiento del controlador neuronal.

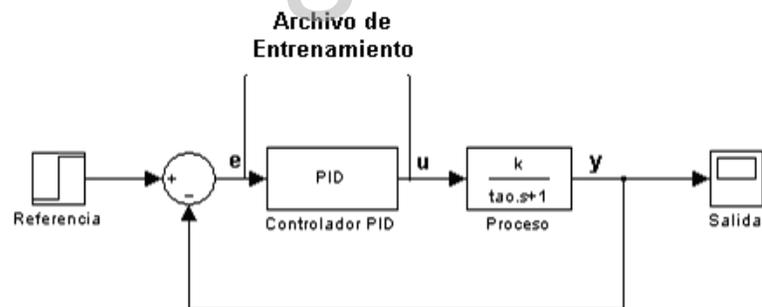


Figura 3.12: Adquisición de datos del Controlador Clásico

- Entrenamiento, realizado a partir de las señales de error y control adquiridas.
- Comprobación del funcionamiento; el controlador clásico es sustituido por el control neuronal para realizar pruebas y verificar el correcto funcionamiento

del controlador; en caso de encontrar errores en el desempeño del controlador, se adquieren nuevamente los datos de entrenamiento para efectuar un mejor entrenamiento de la red neuronal que permita un mejor desenvolvimiento del control neuronal.

[www.bdigital.ula.ve](http://www.bdigital.ula.ve)

# Capítulo 4

## Control Neuronal

En este capítulo se procede al diseño de un controlador neuronal para el sistema de tanques feedback Basic Process Rig 38-100 según el esquema basado en un modelo de control existente. El desarrollo del capítulo se basa en los pasos enumerados previamente para la implementación de este esquema de control neuronal; presentándose secciones para cada uno de ellos con las muestras experimentales y simulaciones correspondientes.

### 4.1 Introducción

El sistema seleccionado para desarrollar el esquema de control neuronal es el feedback Basic Process Rig 38 - 100 el cual permite el estudio de los aspectos fundamentales del control de procesos por ser un modelo sencillo en lazo simple de equipos utilizados en la industria, donde se controla el nivel y/o flujo de agua presente en el sistema. Este proceso consiste en un circuito de tuberías completamente contenido en sí mismo compuesto por los siguientes elementos ubicados estratégicamente en el dispositivo [15]:

- Bomba centrífuga
- 4 válvulas manuales

- 3 válvulas solenoides
- Servoválvula
- Regla de Flujo Visual
- Transmisor de Flujo
- Transmisor de nivel de tipo flotador
- Dos Tanques
- Dispositivo manejador de señales (Process Interface PI)

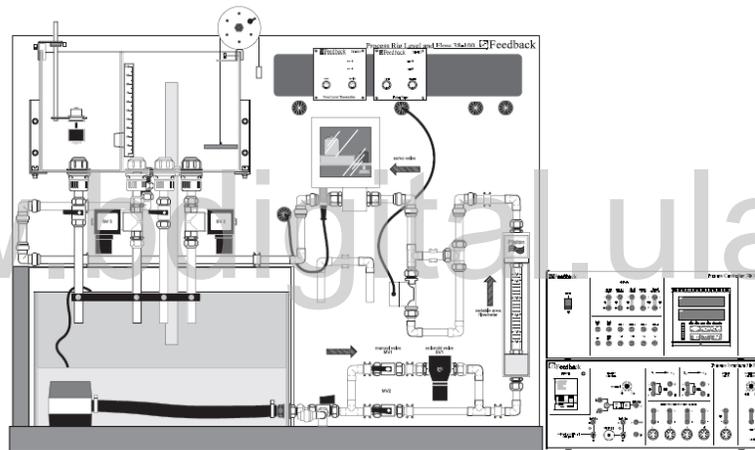


Figura 4.1: Basic Process Rig 38-100

El funcionamiento del sistema consiste en el envío del agua contenida en el tanque inferior a un tanque superior de menor dimensión y doble compartimiento, mediante el bombeo de un motor ubicado en el depósito inferior; el agua fluye a través de la red de tuberías de acuerdo al estado de apertura o cierre de las válvulas manuales y solenoides que posee el dispositivo, así como de la apertura de la servoválvula que limita el paso del flujo a la proporción deseada por el usuario. El sistema es un modelo de lazo simple debido a que las tuberías de desagüe envían el agua nuevamente al tanque inferior, cumpliendo un ciclo de operación.

## 4.2 Descripción de los Elementos del Sistema

### 4.2.1 Surtidor Centrífugo.

Bomba sumergible situada en el fondo del tanque inferior, encargada de enviar el agua hacia las tuberías que conforman el sistema. La energía para su funcionamiento es transferida por un motor eléctrico de corriente continua de 12 V alimentado por la interfaz del Proceso (PI) que enciende o apaga la bomba centrífuga sin permitir valores intermedios [13]. El líquido que entra a la base del rotor es movilizado hacia las tuberías, impartándole energía cinética al fluido para incrementar su presión y movilarlo a través de la bomba por el efecto del rotor [15].

El surtidor se caracteriza por poseer un motor de enfriado para reducir las altas temperaturas alcanzadas durante su operación y un mecanismo contra el cierre de aire que reduce la posibilidad de falla del aparato.

### 4.2.2 Válvulas solenoides.

Permiten o bloquean el paso del agua a través de la red de tuberías mediante sus dos estados de operación: abierto o cerrado, controlados remotamente con señales eléctricas recibidas por una bobina contenida en el interior del dispositivo. La bobina genera un campo electromagnético para movilar el vástago de la válvula y así permitir el flujo de líquido; cuando la señal eléctrica deja de ser aplicada, el campo magnético desaparece y el vástago regresa a su posición de cierre bloqueando el paso de caudal.

El sistema de tanques posee tres válvulas de este tipo, alimentadas con señales eléctricas de 24 Vdc y de diferentes diámetros de apertura.

- SV1 6mm

- SV2 6mm
- SV3 3mm

### 4.2.3 Servoválvula.

Válvula eléctrica cuyo principio de operación es el mismo de los servosistemas; diseñada de tal forma que la salida siga una entrada deseada con el menor error posible [15]. Su apertura o cierre es controlado por una señal de 4-20mA, que indica el estado de la compuerta de la válvula, la cual puede encontrarse en cualquier posición desde 100% abierta a 100 % cerrada.

De acuerdo a la cantidad de corriente suministrada a la servoválvula se produce un movimiento vertical de la compuerta que cambia el área de apertura de la válvula en forma transitoria, transcurriendo cierto tiempo para establecer la nueva apertura de la compuerta ante cualquier cambio en el suministro de corriente de entrada.

### 4.2.4 Medidor de Flujo Visual.

La calibración visual del flujo es realizada mediante un rotámetro; un instrumento de medición de área variable que indica la cantidad de líquido en litros por minuto circulante por la red de tuberías. El medidor consiste en un tubo de vidrio estrecho con un flotador que genera una caída de presión constante al paso del líquido, llegando a una posición estable cuando el peso sumergido es balanceado por el empuje del fluido.

La posición del flotador indica la tasa de flujo entre 0.4 y 4.4 litros por minuto, con una precisión entre +/-3 a +/- 0.2 por ciento [15].

### 4.2.5 Transmisor de Pulsos de Flujo.

Dispositivo para la transmisión de señales de corriente entre 4 y 20 mA de acuerdo a la información emitida por el sensor de pulsos de flujo ubicado en las tuberías. El sensor consiste en una pequeña rueda giratoria de velocidad proporcional al flujo que circula por la instalación; el movimiento de la rueda es detectado por un sensor infrarrojo que genera un tren de pulsos de frecuencia igual a la tasa de rotación, enviado al transmisor de pulsos de flujo.

### 4.2.6 Transmisor de Nivel Flotante (FLT)

El mecanismo se encarga de transmitir la información del nivel de líquido contenido en el tanque superior en señales de corriente de 4 a 20 mA. El sensor es un potenciómetro conectado a través de una fuente de bajo voltaje encendida por un disco flotante [15]; a medida que el nivel de agua se incrementa, el disco produce el giro del potenciómetro incrementando el voltaje a través de él y pasando esta información al transmisor. Al igual que el transmisor de flujo, el manejo de señales en corriente permite la comunicación de datos a larga distancia y la compatibilidad con otros dispositivos.

## 4.3 Identificación del Sistema

El primer paso para diseñar el esquema de control neuronal es la obtención de un modelo del sistema de tanques feedback Basic Process Rig que aporte la información correspondiente al comportamiento dinámico de la planta, para proceder al ajuste de los parámetros del controlador clásico que satisfagan los requerimientos de diseño establecidos [10].

La identificación de sistemas permite encontrar un modelo adecuado del proceso bajo estudio a partir de señales de entrada y salida experimentales; el nivel de agua en el tanque superior corresponde a la señal de salida mientras que la entrada del sistema

se encuentra dada por la apertura de la servoválvula, ambas en unidades de corriente entre 4 y 20 mA relacionadas directamente con el nivel de agua en centímetros y el flujo circulante por la red de tuberías.

Corriente (mA)	Altura (cms)
4	0.0
4.4264	1.3
5.2747	2.5
6.3752	4.0
6.9946	4.9
7.7568	6.0
9.1572	8.0
10.0448	9.3
10.6192	10.1
10.6522	10.2
11.9295	11.8
12.0017	11.9
12.0853	12.0
12.4091	12.5
13.1204	13.5
13.8340	14.5
14.2514	15.2

Tabla 4.1: Relación entre la Corriente emitida por el Sensor de Nivel y Altura del Agua en el Tanque

Ajuste de los datos a una línea recta

$$Altura(cms) = 1,42457218 \times Corriente - 5.12 \quad (4.1)$$

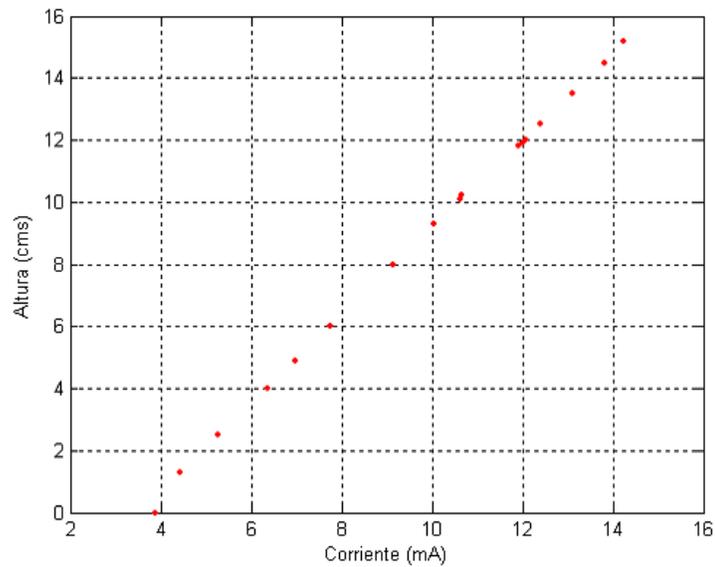


Figura 4.2: Relación entre Corriente y Altura

Flujo (Lts/min)	Sensor de Flujo (mA)
0.0	4.0154
0.6	6.3000
1.0	7.2200
1.2	7.9300
1.8	10.0700
2.4	12.6800
2.8	13.9850
3.0	14.5400
3.2	15.2000

Tabla 4.2: Relación entre la Corriente de la Servoválvula y la Regla de Flujo Visual

Ajuste de los datos a una línea recta

$$\text{Sensor de Flujo (mA)} = 13,54961 \times \text{Flujo} + 3.905 \quad (4.2)$$

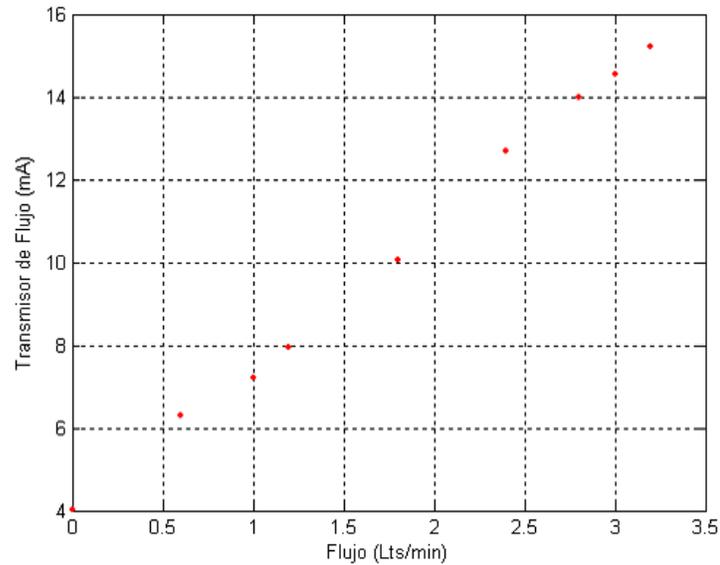


Figura 4.3: Relación entre Medida de Flujo y Corriente

Debido a la no linealidad del proceso, la identificación se realiza entorno a un punto de operación en el cual estabiliza el sistema ante una apertura dada de la servoválvula; efectuando pequeñas variaciones en la señal de entrada para observar el comportamiento del proceso. La lectura del transmisor de nivel es recibida por el módulo de entrada analógico mientras la apertura de la servoválvula es controlada mediante el módulo de salida analógico del Control - Logix, configurado para emitir valores de corriente entre 4.2 y 19 mA con el fin de prevenir posibles averías ocasionadas por el cierre o apertura total de la válvula.

El registro de los datos experimentales manejados por los módulos es realizado en hojas de datos de Microsoft Excel utilizando el servidor OPC de RSLinx y configurando el programa RSView32 como cliente, con tiempo de muestreo de 1 segundo, establecido, según el teorema de muestreo de Shannon [7], y criterios para minimizar los costes computacionales producto de la adquisición de datos.

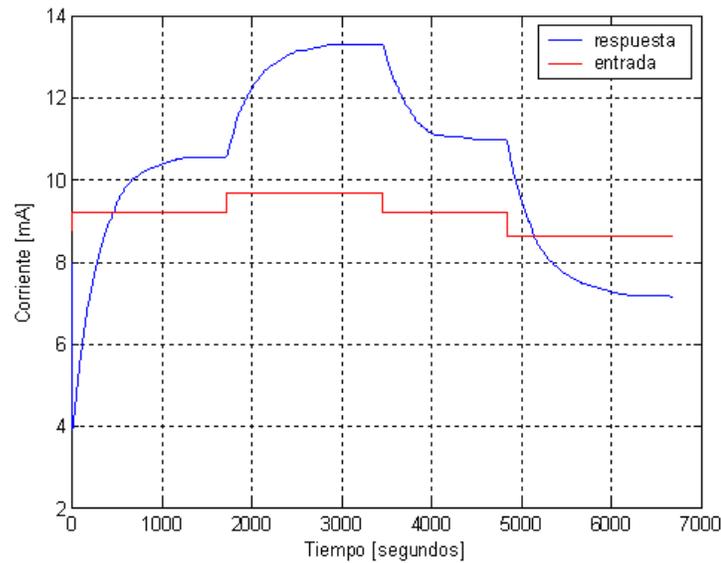


Figura 4.4: Señal de Nivel y Corriente en la Servoválvula

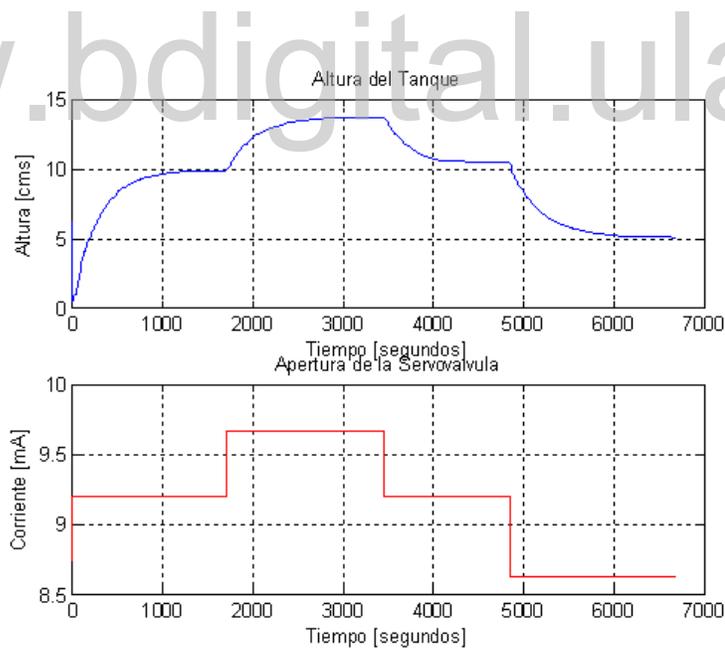


Figura 4.5: Nivel de Agua en el Tanque Superior y Corriente en la Servoválvula

Los datos recolectados indican que la dinámica de la planta corresponde a un sistema de primer orden con retardo; por lo tanto es posible identificar los parámetros

del modelo según el método aproximado para sistemas de este tipo a partir de la respuesta del proceso en lazo abierto de la siguiente forma.

$$G(s) = \frac{k}{\tau s + 1} e^{-t_r s} \quad (4.3)$$

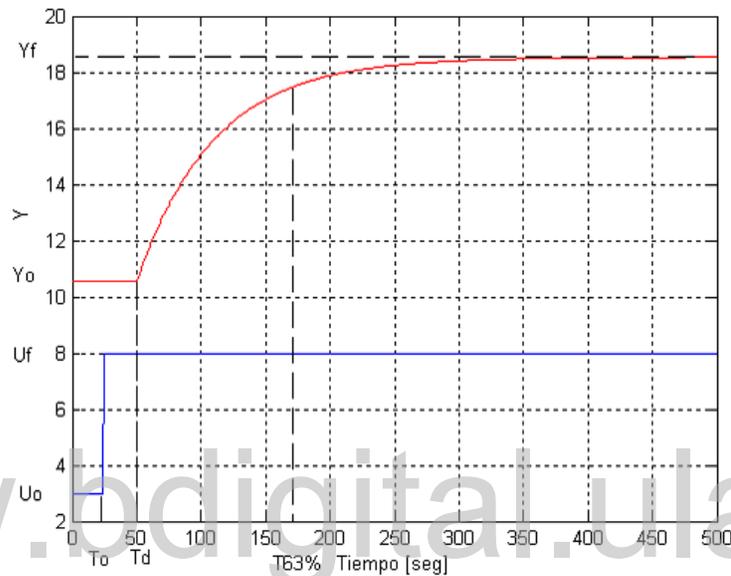


Figura 4.6: Identificación Aproximada

En donde los parámetros son calculados según:

$$k = \frac{Y_f - Y_o}{U_f - U_o} \quad (4.4)$$

$$t_r = t_d - t_o \quad (4.5)$$

$$\tau = t_{63\%} - t_d \quad (4.6)$$

Siguiendo este método se realizaron dos procesos de identificación correspondientes a la variación positiva de la señal de entrada y a su disminución para realizar la identificación del proceso en distintas zonas de los datos experimentales recolectados, elaborando las pruebas de validación respectivas para ambos casos con una zona diferente de datos.

### 4.3.1 Identificación durante la Apertura de la Servoválvula

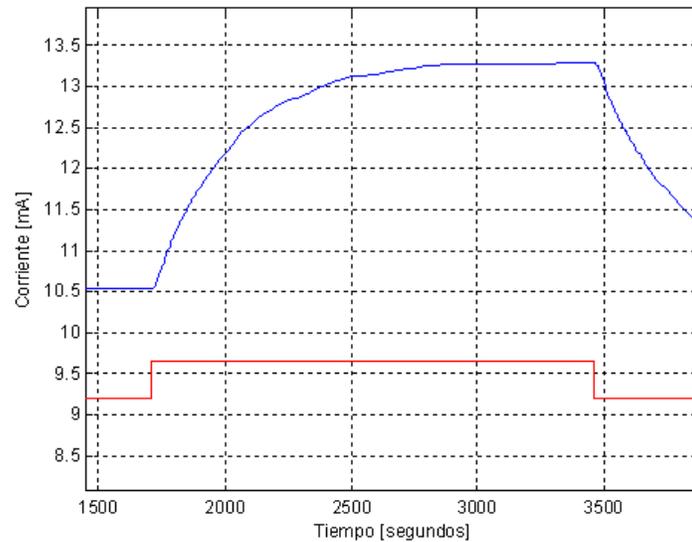


Figura 4.7: Datos Experimentales

$$U_{\text{operación}} = 9.2 \text{ mA} \quad U_{\text{final}} = 9.66 \text{ mA}$$

$$Y_{\text{operación}} = 10.533 \text{ mA} \quad Y_{\text{final}} = 13.28 \text{ mA}$$

$$k = \frac{Y_f - Y_o}{U_f - U_o} \quad (4.7)$$

$$k = 5.9717 \quad (4.8)$$

$$t_r = t_d - t_o \quad (4.9)$$

$$t_r = 1720 \text{ seg} - 1710 \text{ seg} \quad (4.10)$$

$$t_r = 10 \text{ seg} \quad (4.11)$$

$$\tau = t_{63\%} - t_d \quad (4.12)$$

$$\tau = 2022 \text{ seg} - 1720 \text{ seg} \quad (4.13)$$

$$\tau = 302 \quad (4.14)$$

## Función de Transferencia Obtenida

$$G(s) = \frac{5.9717}{302s + 1} e^{-10s} \quad (4.15)$$

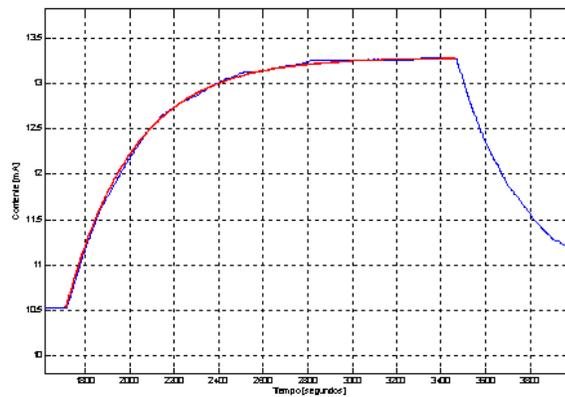


Figura 4.8: Comparación entre el Modelo Identificado y Datos Reales

Validación ante el Cierre de la Servoválvula

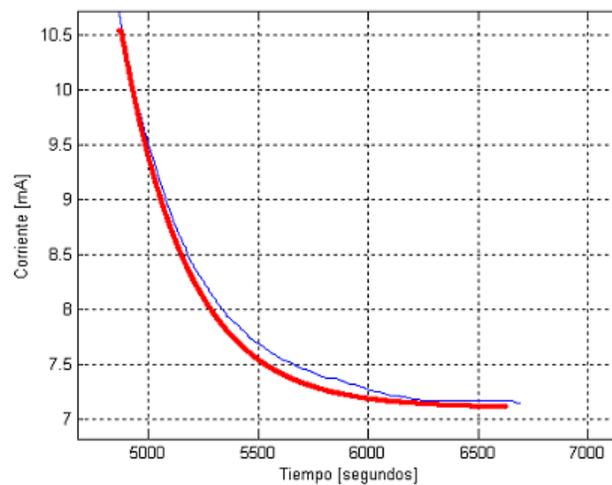


Figura 4.9: Validación del Modelo Identificado durante Cierre de la Servoválvula

### 4.3.2 Identificación durante Cierre de la Servoválvula

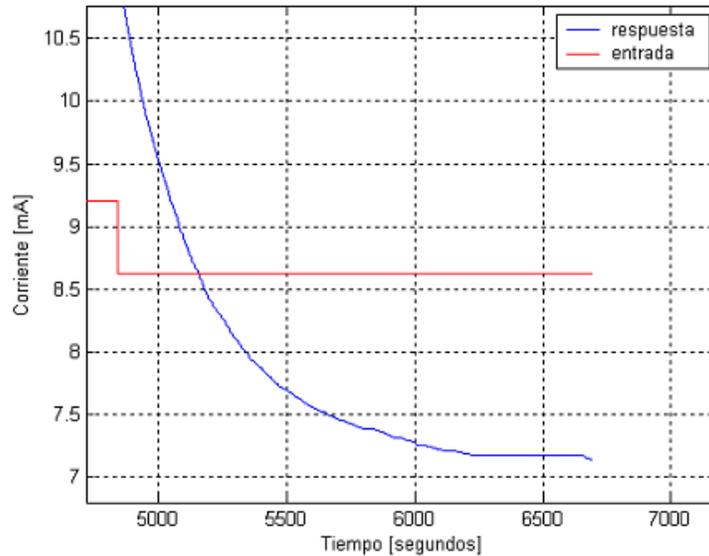


Figura 4.10: Datos Experimentales

$$U_{\text{operación}} = 9.2 \text{ mA} \quad U_{\text{final}} = 8.626 \text{ mA}$$

$$Y_{\text{operación}} = 10.533 \text{ mA} \quad Y_{\text{final}} = 7.177 \text{ mA}$$

$$k = \frac{Y_f - Y_o}{U_f - U_o} \quad (4.16)$$

$$k = 5.846 \quad (4.17)$$

$$t_r = t_d - t_o \quad (4.18)$$

$$t_r = 4845 \text{ seg} - 4839 \text{ seg} \quad (4.19)$$

$$t_r = 6 \text{ seg} \quad (4.20)$$

$$\tau = t_{63\%} - t_d \quad (4.21)$$

$$\tau = 5198 \text{ seg} - 4845 \text{ seg} \quad (4.22)$$

$$\tau = 353 \quad (4.23)$$

Función de Transferencia Obtenida

$$G(s) = \frac{5.846}{353s + 1} e^{-6s} \quad (4.24)$$

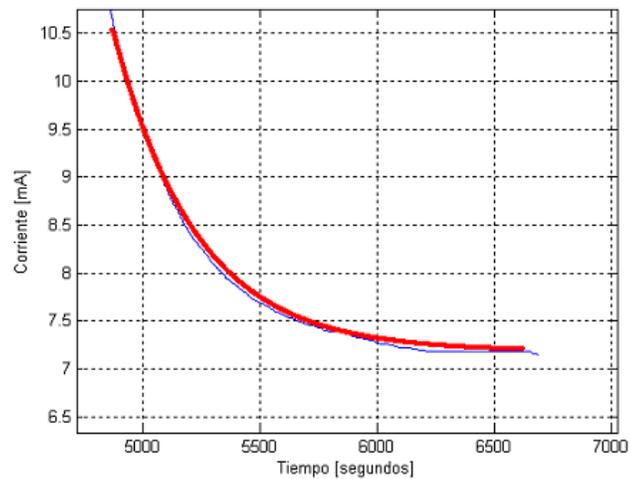


Figura 4.11: Comparación entre el Modelo Identificado y Datos Experimentales

Validación ante apertura de la servoválvula

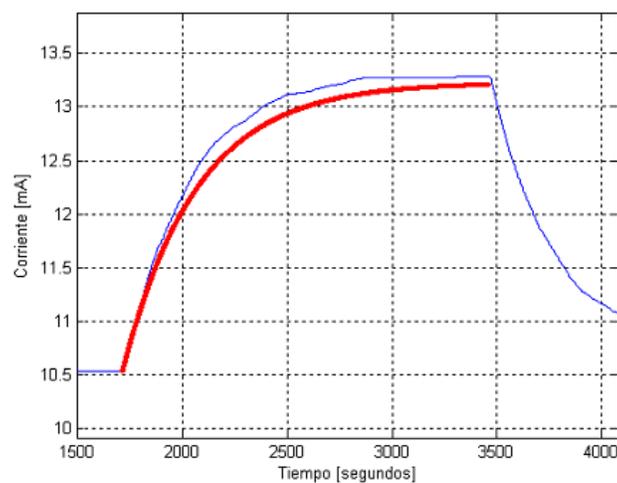


Figura 4.12: Validación del Modelo Identificado durante Apertura de la Servoválvula

Ambos modelos obtenidos a través del método aproximado presentan un buen ajuste con respecto a los datos experimentales; sin embargo, al efectuar una comparación entre ellos, se observa un error menor en la representación obtenida durante la apertura de la servoválvula. Por lo tanto, este modelo será utilizado para comparación con los resultados emitidos por otros métodos de identificación.

El System Identification Toolbox de Matlab<sup>TM</sup> es ampliamente utilizado para ajustar los parámetros a modelos lineales de sistemas dinámicos [23]; trabaja con datos experimentales almacenados en forma matricial y ofrece herramientas de filtrado para el tratamiento previo a la obtención de las representaciones del proceso. Este toolbox cuenta con numerosas funciones para el manejo de modelos paramétricos y no paramétricos, además de ofrecer al usuario la posibilidad de efectuar conversiones entre ambos tipos.

Como resultado de la iteración sucesiva entre los diferentes modelos paramétricos disponibles en el toolbox de Matlab<sup>TM</sup> y la variación de sus parámetros se obtiene un tercer modelo, eligiendo la estructura ARX 1 1 10 correspondiente a un sistema con retardo de 10 segundos, un polo y ningún cero, por presentar el mejor ajuste a los datos de salida.

### 4.3.3 Modelo Paramétrico (ARX 1 1 10)

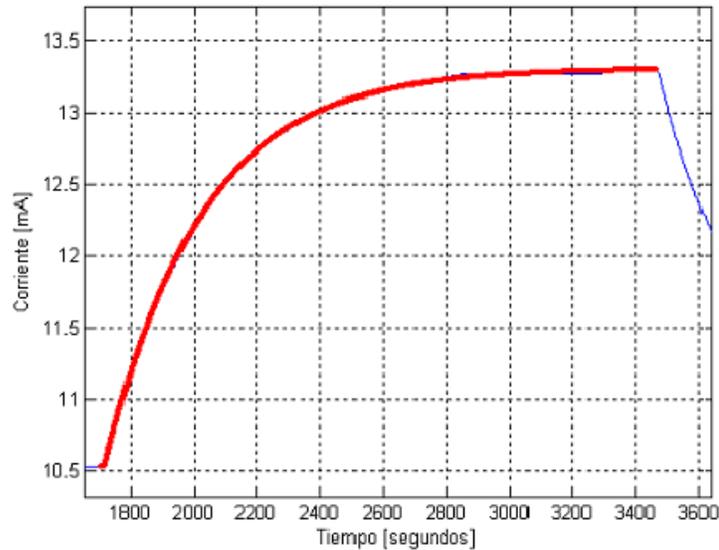


Figura 4.13: Comparación entre el Modelo Identificado y Datos Experimentales durante Apertura de la Servoválvula

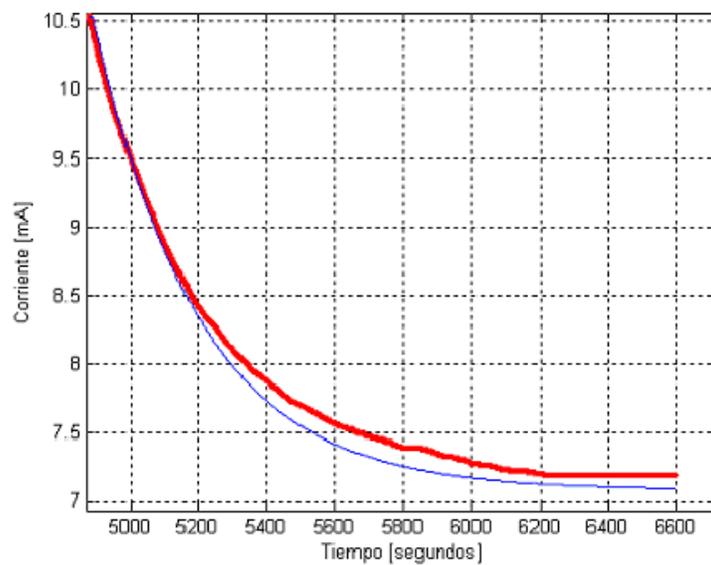


Figura 4.14: Comparación entre el Modelo Identificado y Datos Experimentales durante Cierre de la Servoválvula

Función de Transferencia Obtenida

$$G(s) = \frac{6.0277}{306.2224s + 1} e^{-10s} \quad (4.25)$$

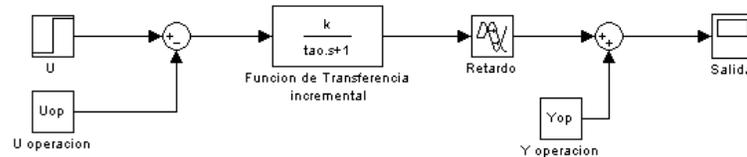


Figura 4.15: Esquema del sistema en lazo abierto

De los tres modelos identificados, es necesario escoger el mejor ajuste para implementar la técnica de control correspondiente; por tal razón se evalúa el error entre la respuesta del modelo identificado y los datos reales según el criterio de la Integral del Error Cuadrático (ISE) para los modelos 1 y 3.

Integral del error cuadrático

$$ISE = \int_0^{\infty} e(t)^2 dt \quad (4.26)$$

Integral del Error Cuadrático (s×mA <sup>2</sup> )	Modelo 1	Modelo 3
Apertura Servoválvula	0.9478	0.3416
Cierre Servoválvula	14.2331	20.0988
ISE Total	15.1809	20.4404

Tabla 4.3: Error entre los Datos Reales y Modelos identificados

La tabla 4.3 muestra los resultados obtenidos al aplicar el criterio de la integral del error cuadrático; allí puede observarse un mejor ajuste del segundo modelo en la región con apertura de la servoválvula, sin embargo ante disminución de la entrada, el mejor ajuste se presenta en el modelo 1. Al calcular los errores totales, se decide

elegir el primer modelo como la representación matemática del sistema de tanques por presentar menor discrepancia con respecto a los datos experimentales.

Modelo seleccionado

$$G(s) = \frac{5.9717}{302s + 1} e^{-10s} \quad (4.27)$$

Para comprobar el rango de validez del modelo lineal escogido, se construye la curva de estado estacionario del sistema a partir de variaciones ascendentes realizadas en la apertura de la servoválvula y registro de los valores de estabilización del nivel de agua en el tanque superior [26]; el intervalo válido para el modelo lineal corresponde a la parte de la curva abarcada por la línea tangente en el punto de operación, el cual es bastante amplio para el caso del sistema de tanques y comprendido entre 5 y 14 cms del nivel de agua aproximadamente.

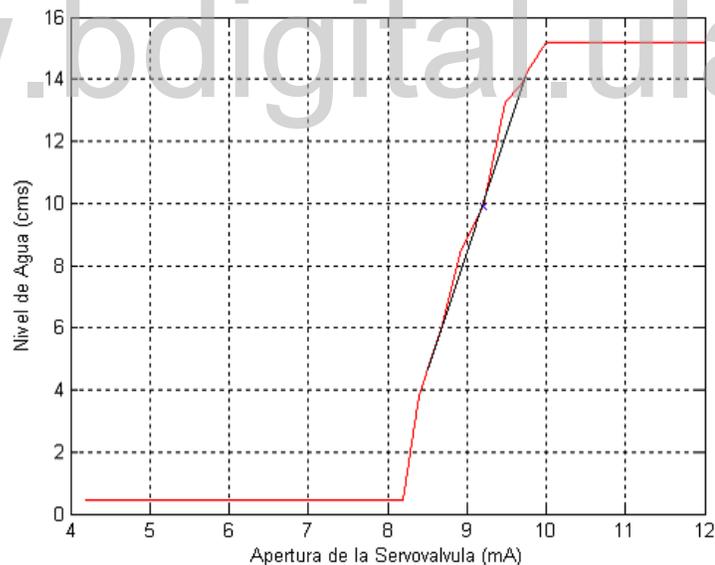


Figura 4.16: Curva de Estado Estacionario

## 4.4 Diseño del Controlador Clásico

El nivel de agua en el tanque superior del sistema de tanques Feedback Basic Process Rig 38-100 constituye la variable a controlar en el proceso mediante la regulación de la cantidad de flujo de entrada indicado por la apertura de la servoválvula; lo que equivale a la modificación sucesiva en la cantidad de corriente enviada por el módulo de salida analógico del Control - Logix. El objetivo de este control es incrementar la velocidad de respuesta del proceso, minimizar los efectos de las perturbaciones y disminuir el error en estado estacionario, estableciendo las siguientes especificaciones de diseño.

- Tiempo de establecimiento = 100 segundos
- Coeficiente de amortiguación  $> 1$  (respuesta sobreamortiguada)

Estos criterios pueden cambiar dependiendo del diseñador, sin embargo el tiempo de establecimiento debe fijarse mayor a 52 segundos para respetar las limitaciones en la servoválvula. A partir de los requerimientos planteados se escoge como ley de control la acción proporcional - integral (PI) por su capacidad para minimizar el error en estado estacionario y proporcionar robustez al sistema.

Sistema en Lazo Abierto

$$G(s) = \frac{5.9717}{302s + 1} e^{-10s} \quad (4.28)$$

En el diseño del controlador se desprecia el efecto del retardo en el modelo matemático, por ser menor que la constante de tiempo del sistema y tratarse de un modelo de primer orden; en este tipo de procesos los efectos del retardo son manejados con el ajuste adecuado de la ganancia proporcional [12].

Sistema en Lazo Cerrado

$$G_o(s) = \frac{5.9717(k_p s + k_i)}{302s^2 + (5.9717k_p + 1)s + 5.9717k_i} \quad (4.29)$$

$$G_o(s) = \frac{0.0198(k_p s + k_i)}{s^2 + \frac{(5.9717k_p + 1)}{302}s + 0.0198} \quad (4.30)$$

$$(4.31)$$

Polinomio característico deseado

$$P_{cd}(s) = s^2 + 2\xi\omega_n s + \omega_n^2 \quad (4.32)$$

donde:

$$ts = \frac{4}{\xi\omega_n} \quad (4.33)$$

$$2\xi\omega_n = \frac{8}{ts} \quad (4.34)$$

$$\omega_n^2 = \left(\frac{4}{ts\xi}\right)^2 \quad (4.35)$$

$$(4.36)$$

Igualando términos entre el polinomio característico deseado y el polinomio en lazo cerrado se obtiene.

$$\frac{5.9717k_p + 1}{302} = 2\xi\omega_n \quad (4.37)$$

$$\frac{5.9717k_p + 1}{302} = \frac{8}{ts} \quad (4.38)$$

$$k_p = \frac{302\frac{8}{ts} - 1}{5.9717} \quad (4.39)$$

$$\omega_n^2 = 0.0198k_i = \left(\frac{4}{ts\xi}\right)^2 \quad (4.40)$$

$$k_i = \frac{\left(\frac{4}{ts\xi}\right)^2}{0.0198} \quad (4.41)$$

$$(4.42)$$

Para:

$$\xi = 2 \quad (4.43)$$

$$k_p = 3.878 \quad (4.44)$$

$$k_i = 0.02 \quad (4.45)$$

$$(4.46)$$

Los parámetros de la ley de control corresponden a un controlador incremental, calculados a partir de un modelo de la planta identificado alrededor de un punto de operación, por lo tanto al implementar la acción de control se sigue el siguiente esquema expresado en términos del punto de operación utilizado.

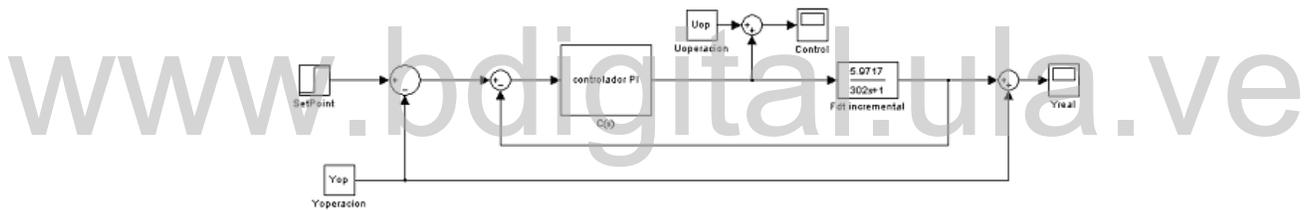


Figura 4.17: Diagrama de Control

De acuerdo a esta representación se implementa el controlador proporcional integral utilizando el bloque de instrucción PID del RSLogix 5000, el cual actúa como una tarea periódica con un tiempo de ejecución de 5 segundos y nivel de referencia de 12 cms.

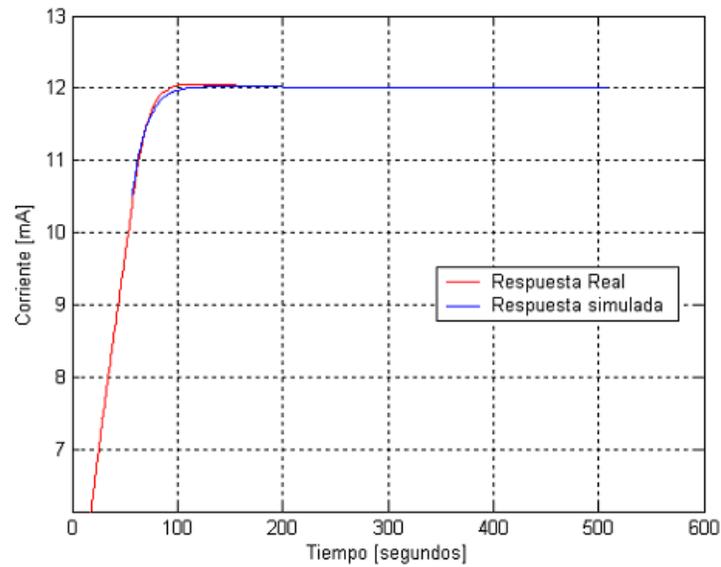


Figura 4.18: Respuesta del Sistema ante el Controlador PID

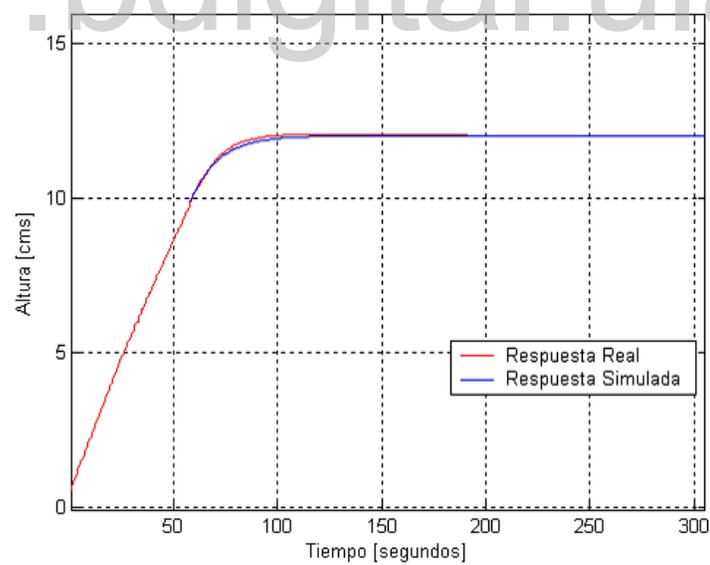


Figura 4.19: Respuesta del Nivel de Agua ante el Controlador PID

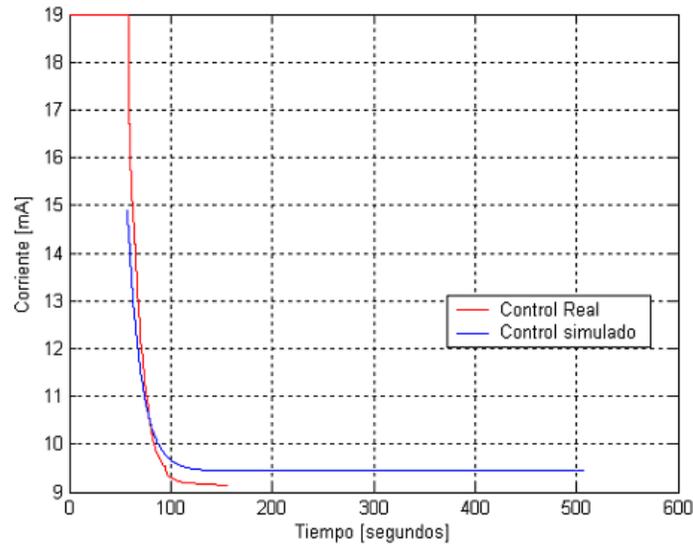


Figura 4.20: Acción de Control Proporcional Integral

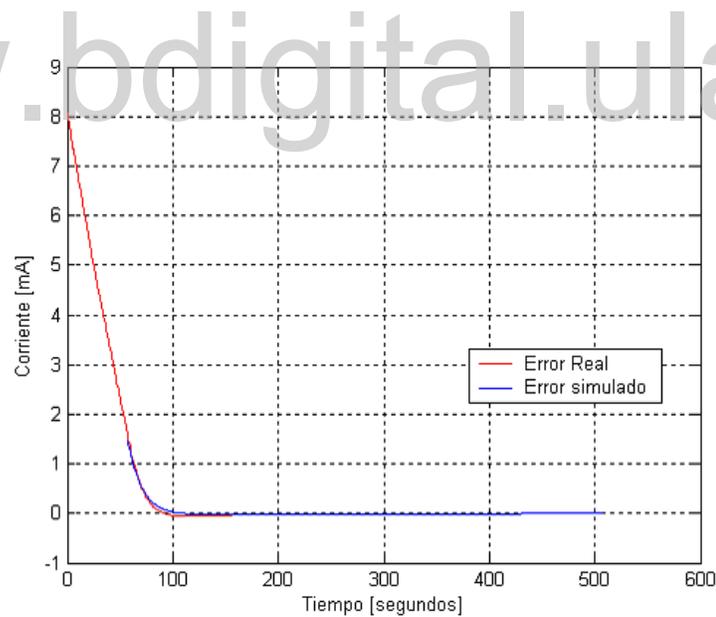


Figura 4.21: Error entre la Referencia y la Respuesta

Al comparar la respuesta simulada de la planta y la obtenida al implementar el controlador mediante el Control - Logix, se observa un correcto ajuste entre ambas y el cumplimiento de los requerimientos de diseño planteados inicialmente. La discrepancia

entre la ley de control real y simulada se deben a las perturbaciones propias del proceso presentes en la salida de agua y al retardo de la servoválvula en responder a los cambios en la señal de control; sin embargo, el control actúa para compensar las deficiencias y lograr el desempeño deseado.

La capacidad de rechazo de perturbaciones del sistema se evalúa verificando la respuesta del sistema ante las siguientes pruebas.

- Apertura de la válvula manual.
- Simulación de falla en el sensor de nivel.
- Cierre de la válvula de salida.
- Agregar un caudal externo.

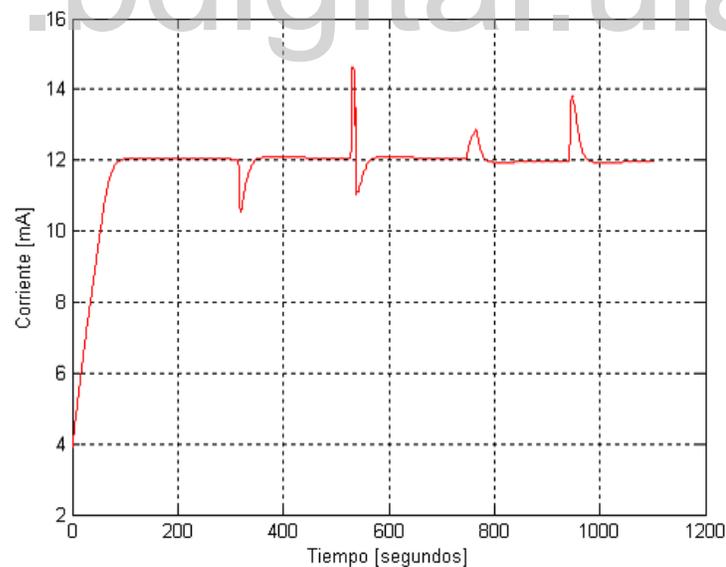


Figura 4.22: Respuesta del Sistema Controlado sometido a Perturbaciones

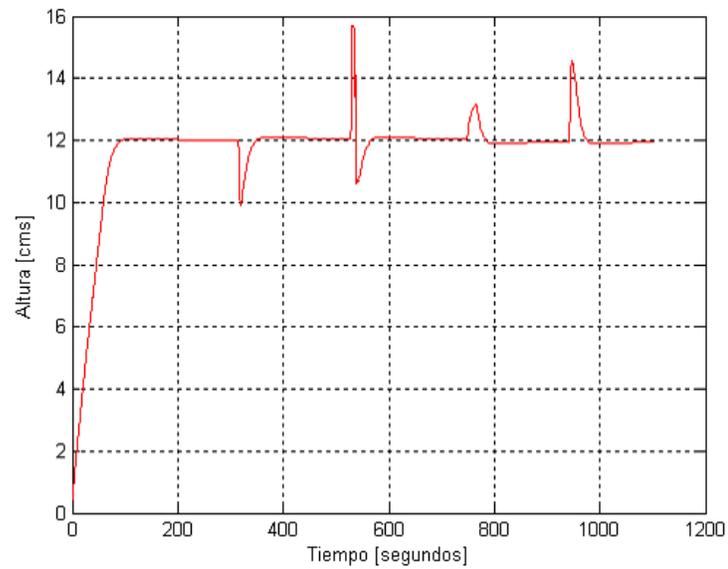


Figura 4.23: Respuesta del Nivel de Agua Controlado sometido a Perturbaciones

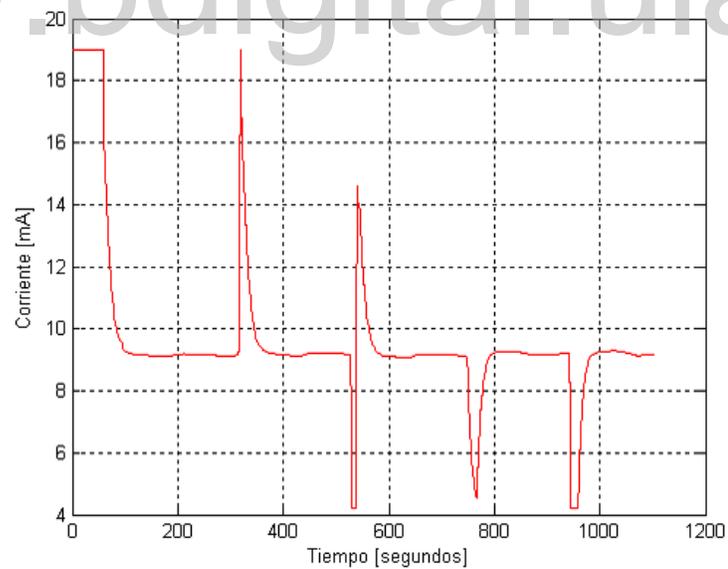


Figura 4.24: Acción de Control Proporcional Integral ante Perturbaciones

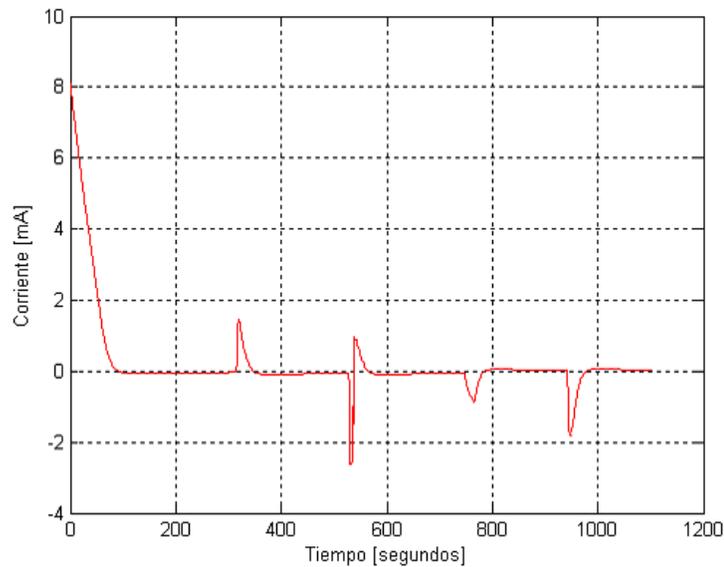


Figura 4.25: Error entre la Referencia y la Respuesta

En las gráficas se observa el cambio instantáneo en la ley de control para compensar las perturbaciones y llevar el sistema de regreso a la referencia deseada, lo cual indica que el sistema de control diseñado es capaz de rechazar las perturbaciones a las que se somete y además cumplir con los requerimientos de funcionamiento especificados.

## 4.5 Entrenamiento de la Red Neuronal

A partir del controlador clásico implementado se procede al entrenamiento de la acción de control neuronal; siendo necesario suministrar a la red la información correspondiente a la dinámica del controlador diseñado [1], la cual se encuentra definida por la diferencia instantánea y anterior entre la referencia deseada y la respuesta del proceso, así como por la salida del controlador en un instante previo en el caso de la acción de control proporcional integral (PI) [7].

$$u(k) = u(k - 1) + k_p(e(k) - e(k - 1)) + k_i T(e(k) - e(k - 1)) \quad (4.47)$$

Los datos recolectados a través de RSView32 correspondientes al sistema controlado mediante la acción proporcional integral, son procesados para generar archivos de entrenamiento y prueba conformados por el error actual, error anterior, control previo y actual que definen la estructura de la ley de control clásica y conforman el patrón de adiestramiento idóneo para la red neuronal. Estos archivos son manejados a través de un software diseñado en Matlab<sup>TM</sup> [23] para el entrenamiento de la red y generación del código fuente en STL (Structured Text Language) para implementar cualquier red neuronal en el Control - Logix.

La definición de la topología de la red neuronal es realizada mediante el software estableciendo los rangos de variación en los datos de entrada, número de capas, neuronas y función de activación en cada una de ellas; con estas especificaciones, se crea una red de conexión hacia adelante (feedforward) a entrenar mediante el algoritmo de retropropagación (Backpropagation) durante un número de periodos señalados por el usuario o hasta alcanzar un error mínimo. El sistema permite el manejo de datos de adiestramiento y prueba almacenados en archivos de texto o en variables disponibles en el espacio de trabajo en formato matricial acorde con la topología definida. Los resultados del proceso de entrenamiento, son utilizados para generar archivos en texto estructurado correspondientes a la estructura definida de la red, el cual es ejecutado en RSLogix5000 para implementar la ley de control neuronal deseada.

Para construir el controlador neuronal seleccionado, la topología corresponde a una red con tres elementos de entrada y una salida, mientras que la configuración apropiada de las capas ocultas es determinada por ensayo y error mediante la iteración sucesiva entre distintas estructuras y la evaluación del aprendizaje logrado. Luego de varios experimentos se implementaron dos controladores neuronales correspondientes a los patrones de entrenamiento con mejor ajuste a los datos de prueba obteniéndose los siguientes resultados.

### 4.5.1 Controlador Neuronal de 5 capas

Número de capas: 5 capas (entrada, salida y tres capas ocultas).

Neuronas de Entrada: 3 (error actual, error y control anterior).

Neuronas de Salida: 1 (control actual).

Neuronas en las capas ocultas: 4, 5 y 3.

Funciones de Activación:

Capa de Entrada: Lineal.

Capas ocultas: Tangente hiperbólica.

Capa de Salida: Lineal.

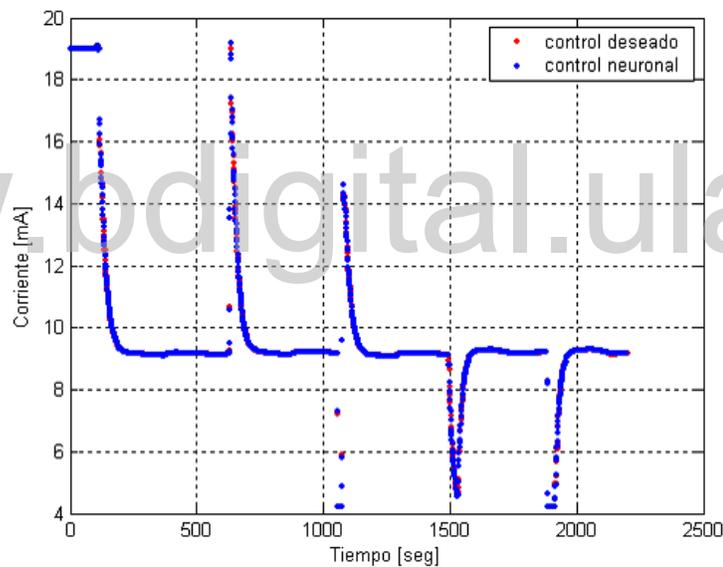


Figura 4.26: Prueba del Controlador Neuronal de 5 Capas

El entrenamiento, prueba y validación de la red arroja un buen ajuste entre los datos de control deseado y el calculado por el controlador neuronal; al implementar el código generado en texto estructurado dentro del RSLogix 5000 y poner en funcionamiento el proceso bajo la ley de control neuronal mediante el Control - Logix se observa una correspondencia entre la respuesta del sistema simulada, su implementación por control clásico y control neuronal, con la existencia de ciertas

discrepancias que no resultan relevantes por encontrarse en un bajo orden de magnitud.

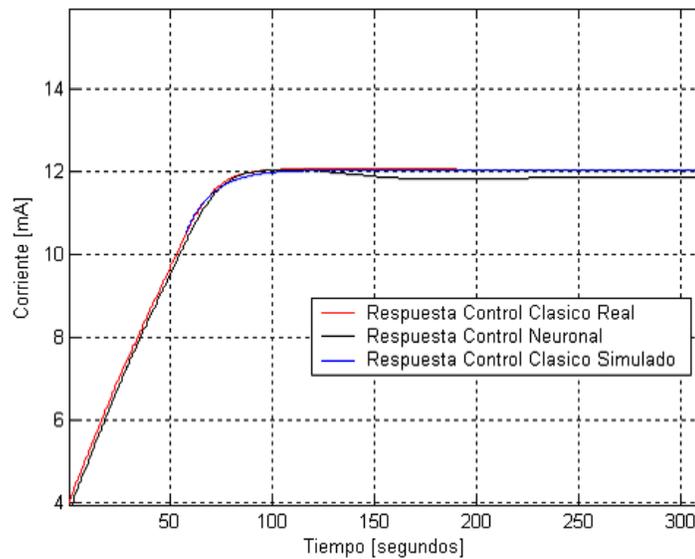


Figura 4.27: Respuesta del Sistema ante el Controlador Neuronal de 5 Capas

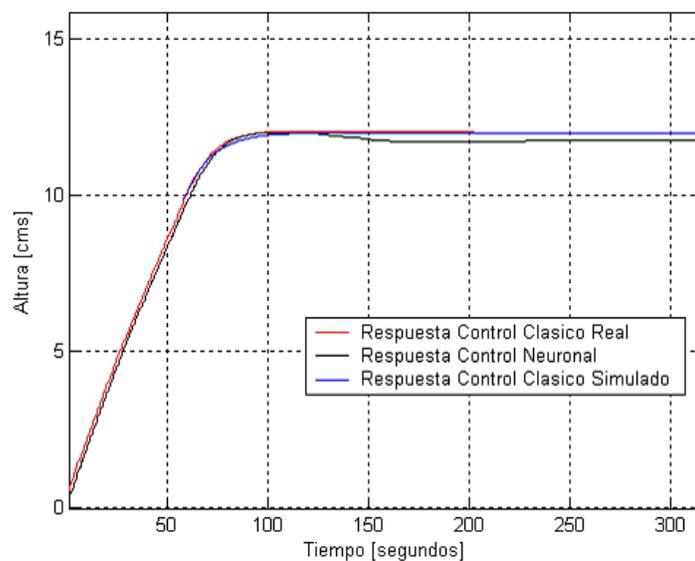


Figura 4.28: Respuesta del Nivel de Agua ante el Controlador Neuronal de 5 Capas

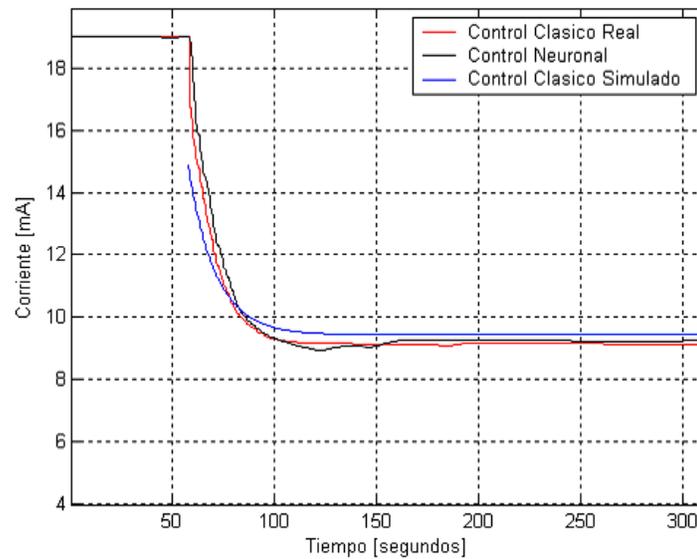


Figura 4.29: Acción de Control Neuronal de 5 Capas

A pesar de cumplir con los requerimientos de diseño indicados para el controlador clásico, esta ley no admite perturbaciones siendo incapaz de devolver el sistema a la referencia deseada luego de agregar un caudal de flujo externo al proceso, por lo cual esta arquitectura no es adecuada para controlar el sistema.

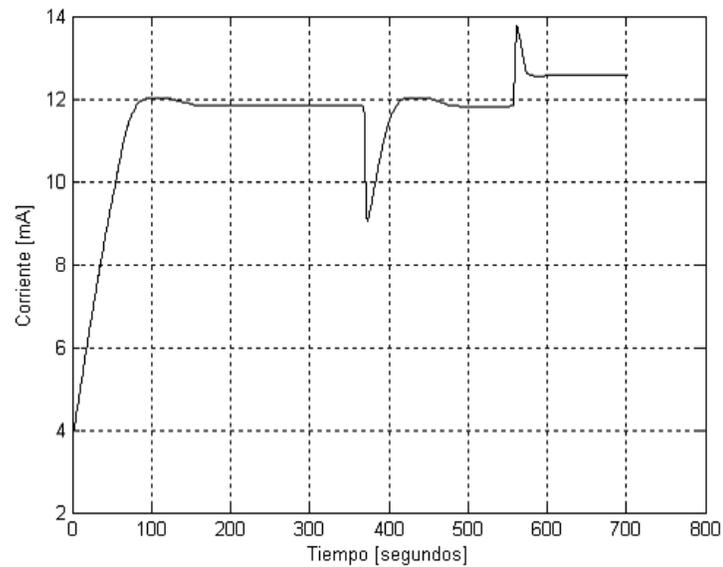


Figura 4.30: Respuesta del Sistema ante el Controlador Neuronal de 5 Capas sometido a Perturbaciones

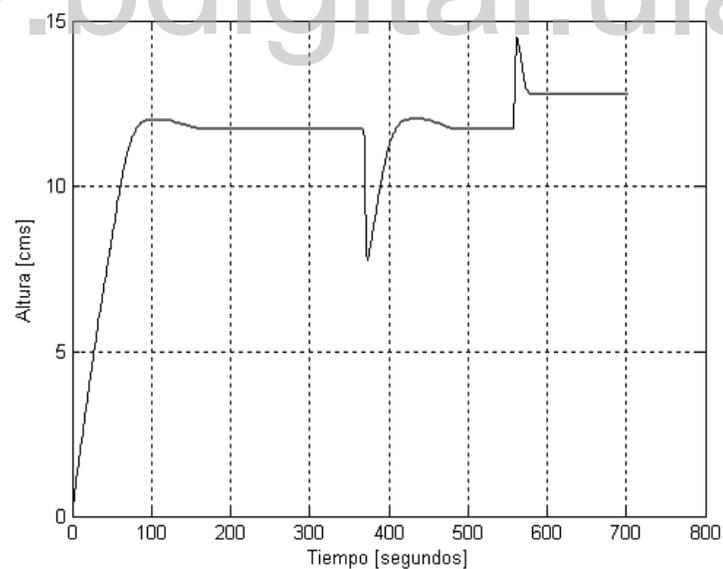


Figura 4.31: Respuesta del Nivel de Agua controlada ante el Controlador Neuronal de 5 Capas sometido a Perturbaciones

### 4.5.2 Controlador Neuronal de 4 Capas

Número de capas: 4 capas (entrada, salida y dos capas ocultas).

Neuronas de Entrada: 3 (error actual, error anterior y control anterior).

Neuronas de Salida: 1 (control actual).

Neuronas en las capas ocultas: 4 y 3.

Funciones de Activación:

Capa de Entrada: Lineal.

Capas ocultas: Tangente hiperbólica.

Capa de Salida: Lineal.

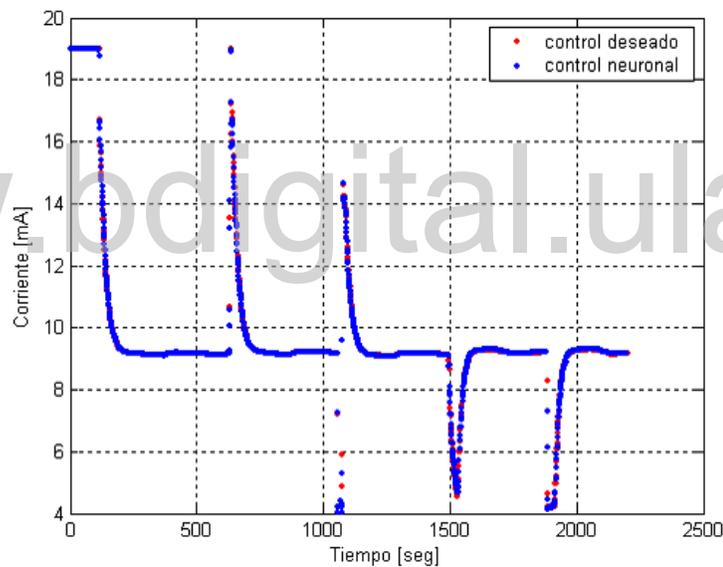


Figura 4.32: Prueba del Controlador Neuronal de 4 Capas

Al cambiar la topología de la red se observa un mejor desempeño del controlador neuronal y disminución de la discrepancia con respecto a la respuesta mediante control clásico y su simulación, además de ofrecer comportamiento adecuado ante las perturbaciones introducidas al sistema como cierre de la válvula de salida o existencia de un caudal de entrada externo al tanque superior.

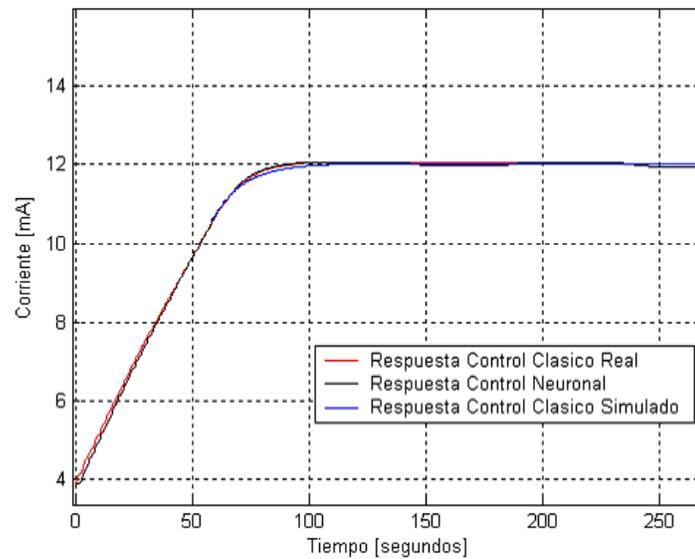


Figura 4.33: Respuesta del Sistema ante el Controlador Neuronal de 4 Capas

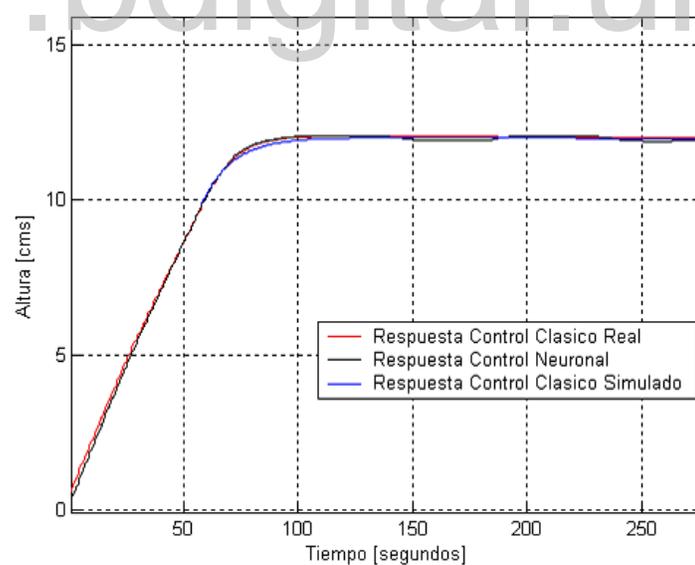


Figura 4.34: Respuesta del Nivel de Agua ante el Controlador Neuronal de 4 Capas

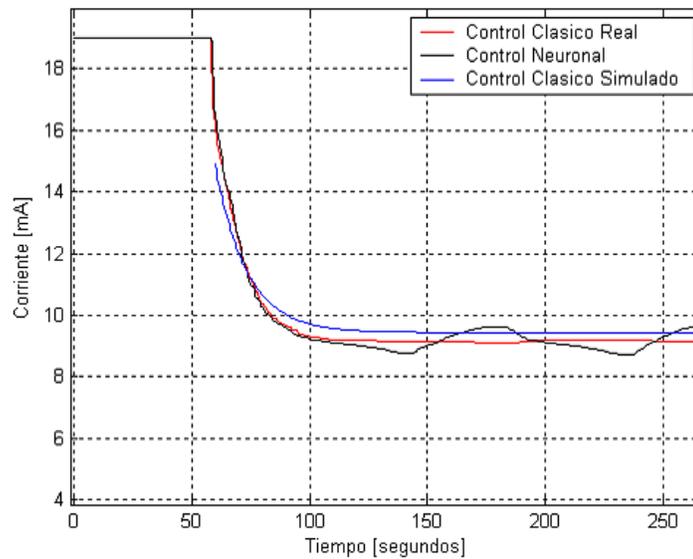


Figura 4.35: Acción de Control Neuronal ante el Controlador Neuronal de 4 Capas

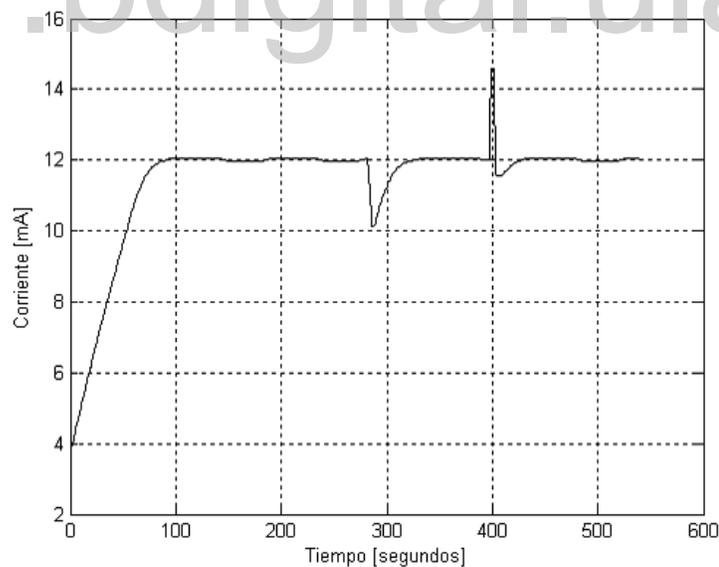


Figura 4.36: Respuesta del Sistema ante el Controlador Neuronal de 4 sometido a Perturbaciones

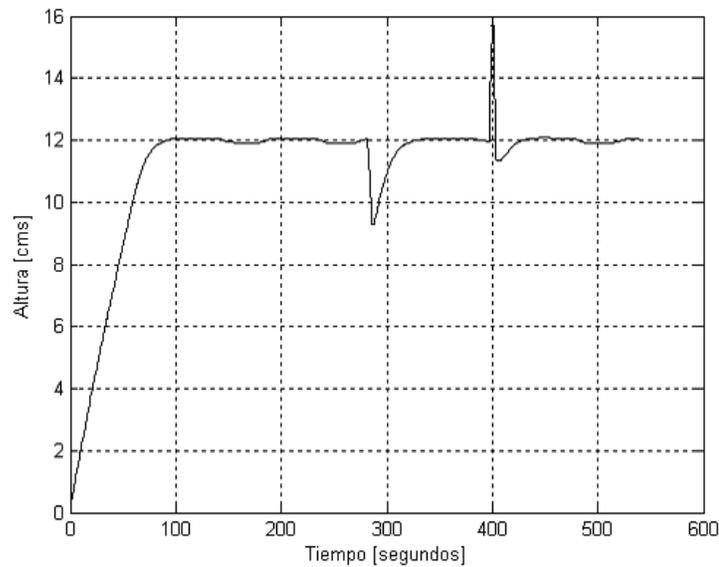


Figura 4.37: Respuesta del Nivel de Agua ante el Controlador Neuronal de 4 sometido a Perturbaciones

Además del rechazo a perturbaciones se desea el funcionamiento del controlador a pesar de cambios en el punto de referencia al cual debe ser llevado, verificándose su comportamiento a una altura de 10 cms y observando el correcto desempeño del sistema.

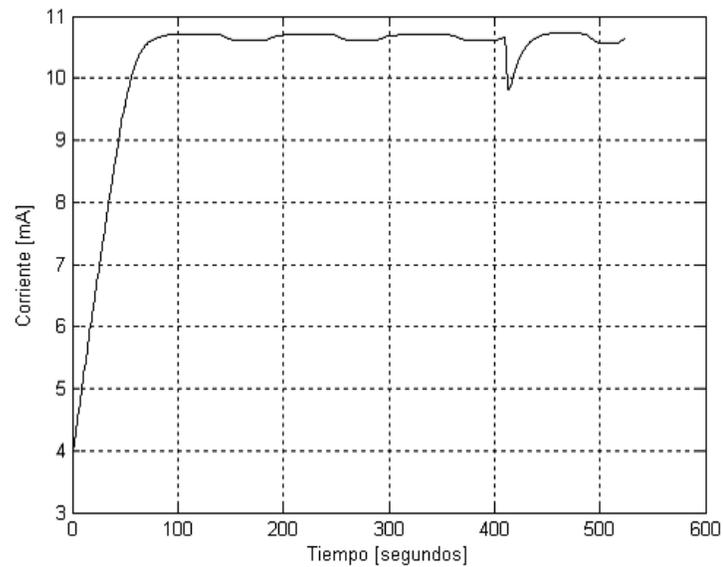


Figura 4.38: Respuesta del Sistema Controlado ante Nueva Referencia y Perturbaciones

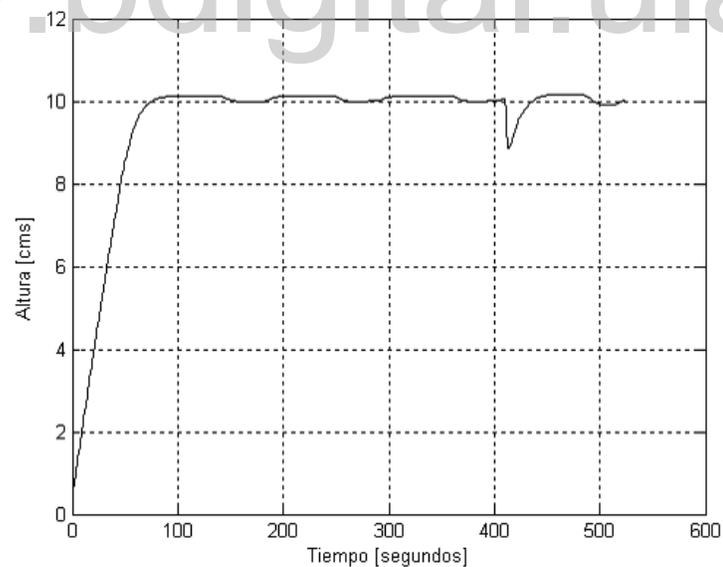


Figura 4.39: Respuesta del Nivel de Agua Controlado ante Nueva Referencia y Perturbaciones

## 4.6 Construcción de la Red Neuronal Bajo Control - Logix

Antes de programar la red neuronal en el lenguaje del Control - Logix, se diseñó una estructura general de programación para implementar cualquier red neuronal a través de C++ y funciones matemáticas elementales. El código funciona a partir de los pesos neuronales almacenados matricialmente e información topológica de la red como número de capas, cantidad de neuronas por capa y funciones de activación en cada una de ellas; posteriormente se efectúa el cálculo de la salida ante una entrada dada mediante operaciones matemáticas simples por capa.

Una vez diseñado el código general para redes neuronales en C++ se traduce la información a texto estructurado, siendo necesario programar también la operación exponencial en una subrutina en texto estructurado, por ser esencial para la ejecución de las funciones de activación de las redes. Conocida la programación de una red neuronal bajo el Control - Logix, el siguiente paso consiste en generar el código dentro del software de entrenamiento que permita escribir archivos en texto estructurado en el formato general encontrado inicialmente y aprovechar la disponibilidad de la información de la red neuronal entrenada.

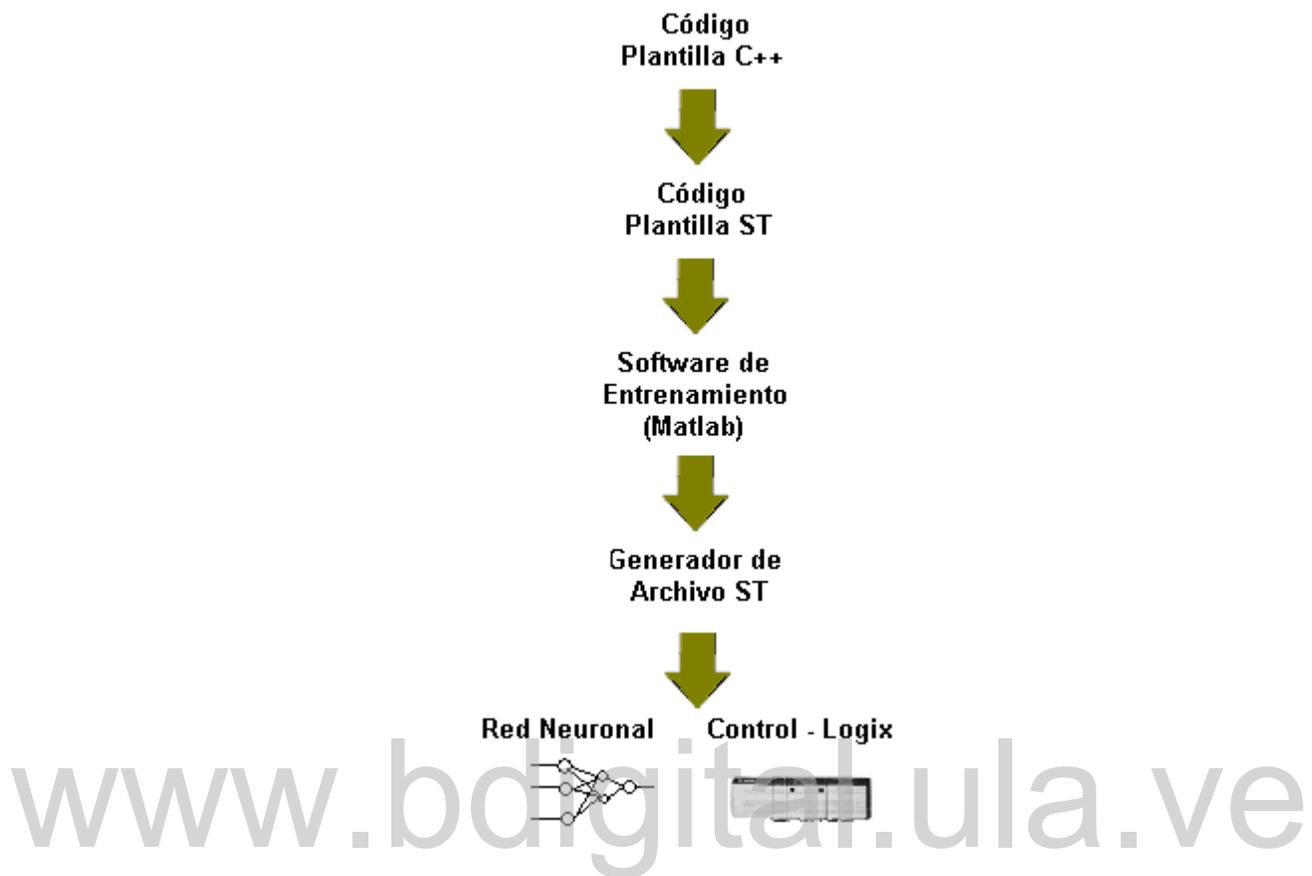


Figura 4.40: Esquema de Desarrollo de la Red Neuronal bajo Control - Logix

## 4.7 Supervisión del Sistema

El proceso de supervisión del sistema es efectuado a través de una interfaz desarrollada en RSVIEW32 donde se observa la evolución del sistema en término de los errores entre la salida y la referencia. Desde el ambiente gráfico se manejan algunos elementos del sistema de Tanques, específicamente, las válvulas solenoides de entrada y salida a través del módulo digital. La lógica implementada previene la ocurrencia de fallas y permite el funcionamiento conjunto de estos elementos; al existir caudal de salida se permite la entrada de flujo al tanque superior y de igual forma, la ausencia de caudal de entrada bloquea la salida del líquido.

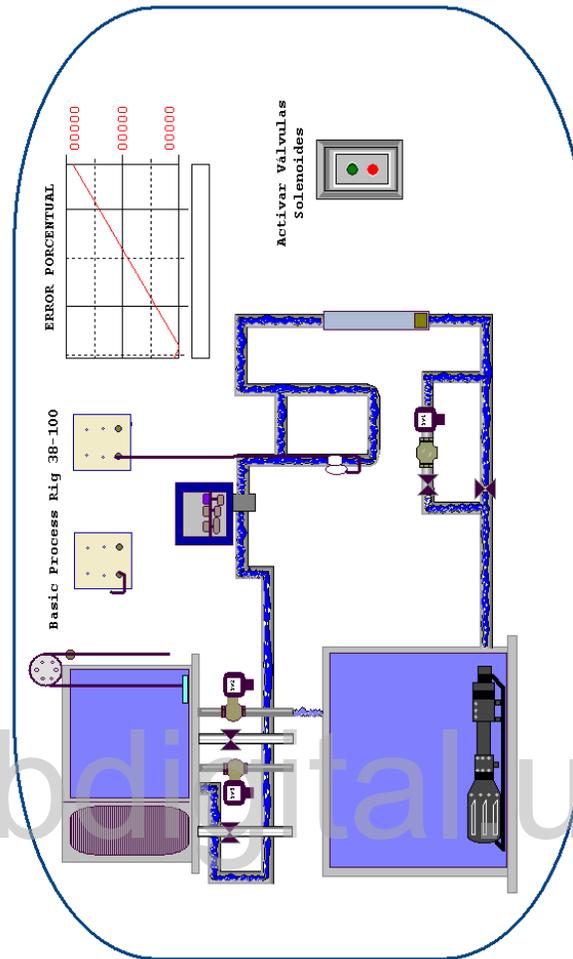


Figura 4.41: Supervisión del Proceso en RSView32

# Capítulo 5

## Conclusiones

El campo de aplicación de las técnicas de control inteligente ha crecido considerablemente en los últimos años; los nuevos sistemas producto de los avances tecnológicos requieren controladores sofisticados acordes a la complejidad de cada proceso. Las características de las redes neuronales ofrecen una alternativa para el control de procesos, y su unión con las nuevas tecnologías para automatización como es el caso del Control - Logix plantea una evolución hacia sistemas avanzados que reúnen las ventajas proporcionadas por ambos, en cuanto a adaptabilidad, autonomía, tolerancia a fallas, aprendizaje y operatividad en condiciones adversas.

Este proyecto constituye un aporte en el estudio e implementación de técnicas de control inteligente basadas en redes neuronales bajo Control - Logix a través del controlador neuronal desarrollado y la creación de la estructura para generar el código en el lenguaje de programación del dispositivo que permite tratar diferentes arquitecturas neuronales en un tiempo de ejecución muy corto ubicado en el orden de los nanosegundos, por lo cual es posible diseñar controladores neuronales inclusive para procesos con un tiempo de respuesta corto.

Las simulaciones e implementación del esquema de control neuronal basado en un modelo existente, produce resultados satisfactorios en el control de nivel de agua en el sistema de tanques y permite observar una aplicación de control neuronal

haciendo uso de la plataforma creada para redes neuronales en el Control - Logix. Así mismo, la creación del sistema de supervisión en RSView 32 permite el control remoto del proceso y la manipulación de sus elementos y variables aprovechando totalmente las potencialidades del dispositivo en cuanto a comunicaciones a distancia y velocidad de procesamiento.

Los avances logrados con este estudio deben ser aprovechados en su continuación con la construcción de otras técnicas de control neuronal bajo Control - Logix que permitan realizar un control adaptativo del proceso e inclusive desarrollar la base de programación necesaria para ejecutar otros tipos de control inteligente mediante el dispositivo.

[www.bdigital.ula.ve](http://www.bdigital.ula.ve)

# Bibliografía

- [1] José B. Cotero. Control clásico y control inteligente. <http://www.fumec.org.mx/>.
- [2] Comisión Internacional de Electrotécnica. Página principal de la iec. <http://www.iec.ch/>, 2005.
- [3] D.I.S.A. Introducción a los dispositivos de control: Plcs. <http://lorca.umh.es/isa/es/temas/plc/doc/Tema2.PDF>, 2002.
- [4] Donald Hebb. *Organization of Behavior: A neuropsychological theory*. John Wiley and Sons Inc., 1949.
- [5] Andrew Hodges. *Turing*. Phoenix, Gran Bretaña, 1997.
- [6] J. J. Hopfield. *Neural networks and physical systems with emergent collective computational abilities*. Proceedings of the National Academy of Sciences, USA, 1982.
- [7] Rolf Isermann. *Digital Control Systems*, volume 1. Segunda edición, 1989.
- [8] David M. Skapura James A. Freeman. *Neural Networks: Algorithms, Applications and Programming Techniques*. Addison-Wesley, 1991.
- [9] Clifford Lau. *Neural Networks, Theoretical Foundations and Analysis*. IEEE Press, Arlington, Virginia, 1991.
- [10] María Elena López. Identificación de sistemas. aplicación al modelado de un motor de continua. <http://www.depeca.uah.es>, 2002.

- [11] S. Papert M. Minsky. *Perceptrons: An Introduction to Computational Geometry*. The MIT Press, 1969.
- [12] Jesus Martínez. Comparación de diferentes estrategias de control para sistemas con retardo. *IV Congreso de Automatización y Control*, page 62, Noviembre 2003.
- [13] Eduardo Pabón. Simulación de una planta de variables de nivel y caudal. Pasantía de investigación, Universidad de los Andes, junio 2001.
- [14] A Porras. *Autómatas Programables*. McGraw - Hill, España, primera edition, 1986.
- [15] Procon. *Workbook Feedback 38-001*, 2000.
- [16] Rockwell-Automation. Comunicaciones plataforma logix. <http://www.ab.com/logix/controllogix/communication.html>, 2005.
- [17] Rockwell-Automation. Controllogix platform. <http://www.ab.com/logix/controllogix/index.html>, 2005.
- [18] Rockwell-Automation. Logix platform. <http://www.ab.com/logix/>, 2005.
- [19] Rocwell-Automation. Rslogix 5000 enterprise series software. (9324-PP001B-EN-P), marzo 2004.
- [20] Addison Ríos-Bolívar. *Metodologías para la Construcción de Sistemas de Detección y Diagnóstico de Fallas*. PhD thesis, Universidad de los Andes, Facultad de Ingeniería, Junio 2003.
- [21] Frank Rosenblatt. *The perceptron: A perceiving and recognizing automation*. Cornel Aeron, 1957.
- [22] Rockwell Software. *Product Training for RSVIEW32*. Rockwell Automation, USA, 2000.
- [23] The-MathWorks. Página principal de matlab. <http://www.mathworks.com>, 2005.

- [24] Ricardo Valverde. *Control de Sistemas mediante Redes Neuronales Aprendizaje por Refuerzo*. PhD thesis, Universidad Carlos III de Madrid, Departamento de Ingeniería Eléctrica, Electrónica y Automática, 1999.
- [25] Célida C. Vega. *Un Controlador Lineal Basado en Redes Neuronales*. Julio 1995.
- [26] Rafael Vásquez. Estrategias para compensar no linealidades generadas por válvulas en lazos de control de flujo. 2005.
- [27] Ian G. Warnock. *Programmable controllers Operation and Application*. Prentice Hall, 1988.
- [28] W. Pitts Warren McCulloch. *Logical Calculus of the Ideas Immanent in Nervous Activity*. *Bulletin of Mathematical Biophysics*. 1943.
- [29] Bernard Widrow and Hoff Marcian. *Adaptive Switching Circuits*. IRE WESCON Convention Record, New York, USA, 1960.
- [30] Robert E. Wilhelm. *Programmable Controller Handbook*. Hayden Book Company, 1984.

# Anexo A

## Controlador Proporcional Integral

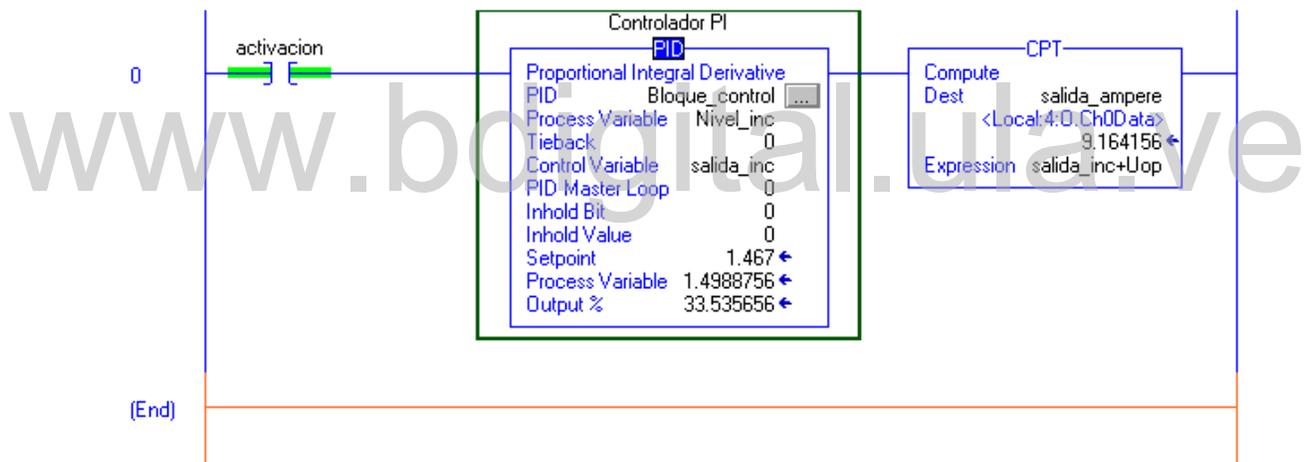


Figura A.1: Bloque de Control Clásico

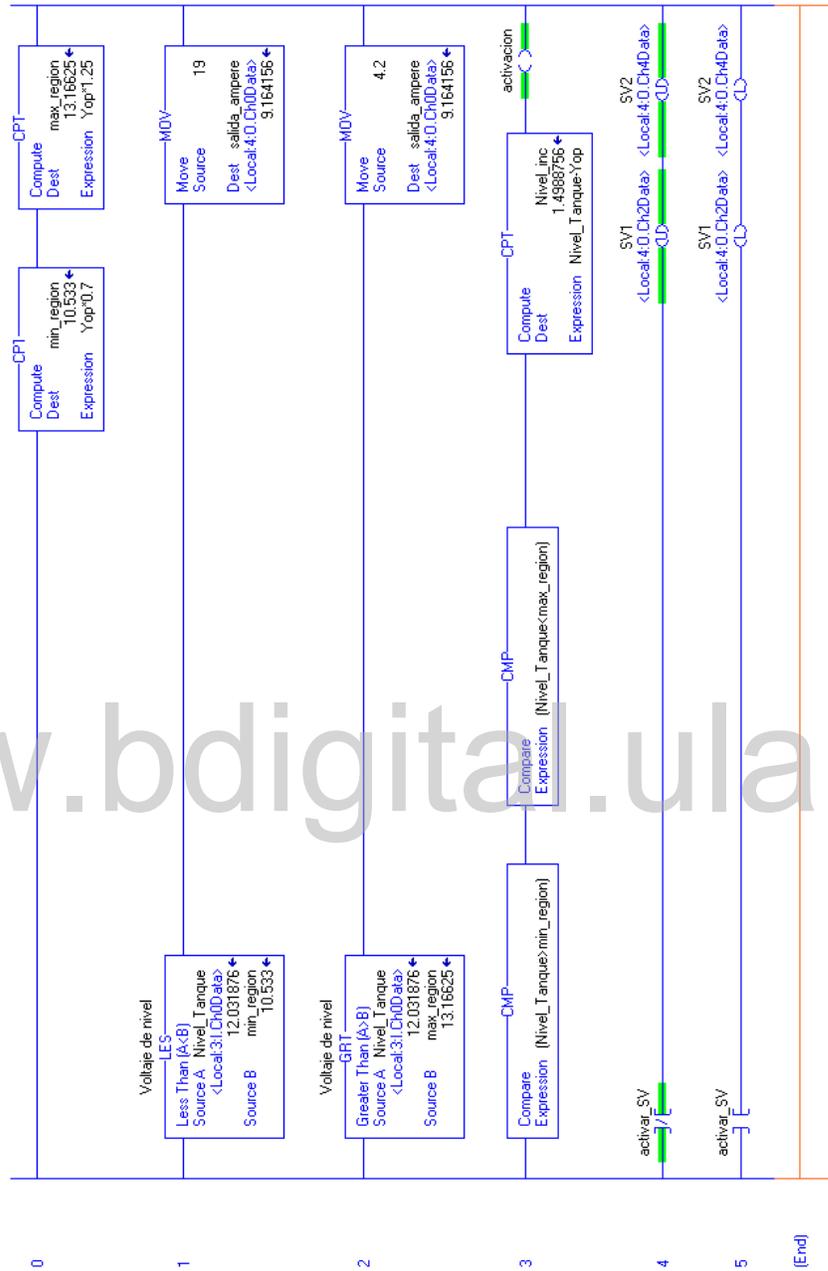


Figura A.2: Rutina Continua para el acondicionamiento de valores enviados al controlador

# Anexo B

## Controlador Neuronal

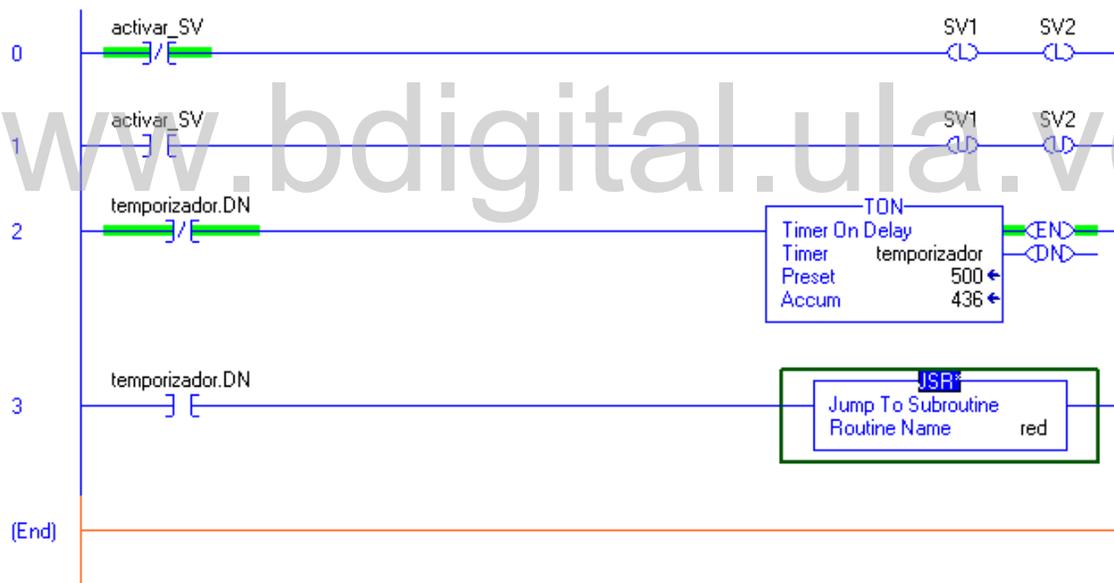


Figura B.1: Manejo de los elementos del proceso y temporización de la acción de control neuronal

## B.1 Código Fuente en Texto Estructurado del Control neuronal

(\* VALOR DE LOS PESOS DE LA RED \*)

Pesos[0,0]:=-2.0537;

Pesos[0,1]:=-1.4411;

Pesos[0,2]:=-5.075;

Pesos[0,3]:=-0.12253;

Pesos[1,0]:=-0.89884;

Pesos[1,1]:=2.5177;

Pesos[1,2]:=-4.7363;

(\* Arquitectura de la Red \*)

fdt[0]:=0;

fdt[1]:=1;

fdt[2]:=1;

fdt[3]:=0;

ncapas:=4;

nneuronas[0]:=3;

nneuronb[0]:=4;

nneuronas[1]:=4;

nneuronb[1]:=5;

nneuronas[2]:=3;

```
nneuronb[2]:=4;
nneuronas[3]:=1;
nneuronb[3]:=1;

(* Inicialización de la Matriz de Salida *)
```

```
for i:=0 to 11 do
  for j:=0 to 11 do
    nentrada[i,j]:=0;
    valor_final[i,j]:=0;
  end_for;
end_for;
```

```
(* Especificación de las Entradas, modificar ante cambios
de proceso *)
```

```
error_ac:=referencia-Nivel_Tanque;
nentrada[0,0]:=error_ac;
nentrada[0,1]:=error_an;
nentrada[0,2]:=u_an;
```

```
(* Calculo de las salidas de la red *)
```

```
aux:=0;
for i:=0 to ncapas-2)do
  for j:=0 to (nneuronas[i+1]-1) do
```

```
for k:=0 to (nneuronb[i]-1) do
  if (k=(nneuronb[i]-1))then
    if (j=0) then
      nentrada[i,k]:=1;
    end_if;
  end_if;
  nentrada[(i+1),j]:=nentrada[i+1,j]+nentrada[i,k]*Pesos[k+aux,j];
end_for;
case fdt[i+1] of
  1: x:=nentrada[i+1,j];
    if (x>20) then
      valor_final[i+1,j]:=1;
      nentrada[i+1,j]:=valor_final[i+1,j];
    else
      if (x<-20) then
        valor_final[i+1,j]:=-1;
        nentrada[i+1,j]:=valor_final[i+1,j];
      else
        JSR(exponencial,1,x);
        aux:=exp;
        x:=-nentrada[i+1,j];
        JSR(exponencial,1,x);
        valor_final[i+1,j]:=(auxt-exp)/(auxt+exp);
        nentrada[i+1,j]:=valor_final[i+1,j];
      end_if;
    end_if;
end_if;
```

```
2: x:=nentrada[i+1,j];
   if (x>20) then
       valor_final[i+1,j]:=1;
       nentrada[i+1,j]:=valor_final[i+1,j];
   else
       if (x<-30) then
           valor_final[i+1,j]:=0;
           nentrada[i+1,j]:=valor_final[i+1,j];
       else
           x:=-nentrada[i+1,j];
           JSR(exponencial,1,x);
           valor_final[i+1,j]:=1/(1+exp);
           nentrada[i+1,j]:=valor_final[i+1,j];
       end_if;
   end_if;
   else
       valor_final[i+1,j]:=nentrada[i+1,j];
   end_case;
end_for;
aux:=aux+nneuronb[i];
end_for;

(* Prevención de daño del actuador *)

if (valor_final[ncapas-1,0]>19) then
    valor_final[ncapas-1,0]:=19;
```

```
end_if; if (valor_final[ncapas-1,0]<4.2) then
    valor_final[ncapas-1,0]:=4.2;
end_if; error_an:=error_ac; salida_ampere:= valor_final[ncapas-1,0];
u_an:=salida_ampere;
```

[www.bdigital.ula.ve](http://www.bdigital.ula.ve)

[www.bdigital.ula.ve](http://www.bdigital.ula.ve)

## Anexo C

# Programación de la Red Neuronal

### C.1 Interfaz Gráfica

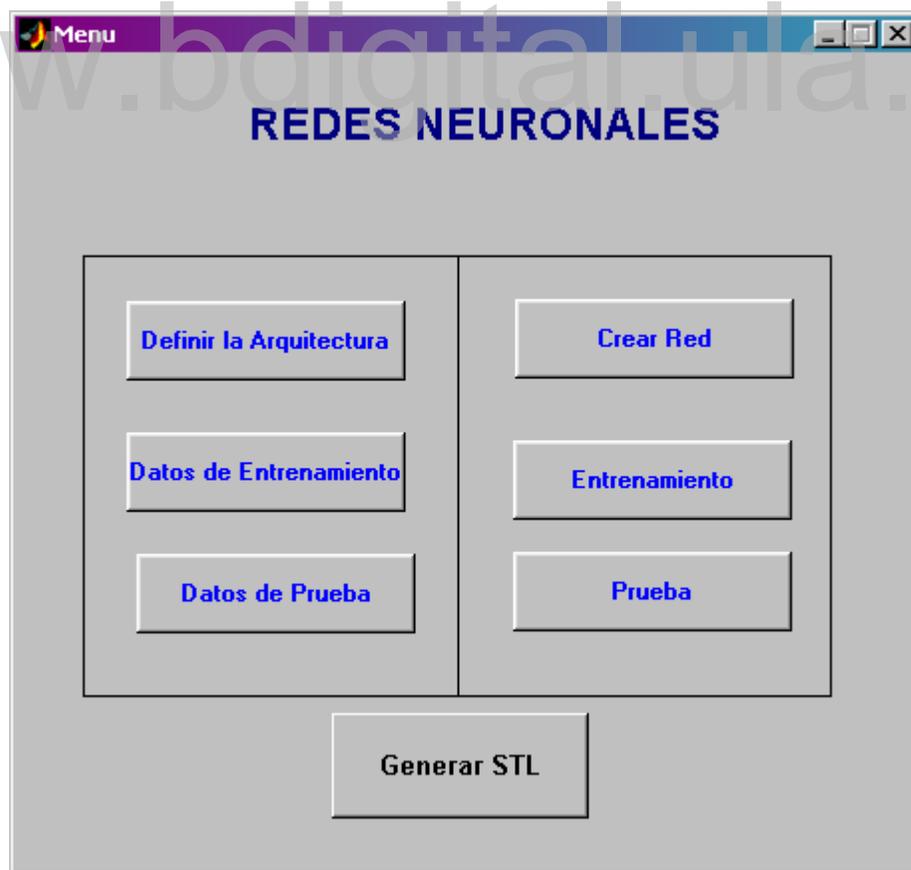


Figura C.1: Menu Principal del Software de Entrenamiento

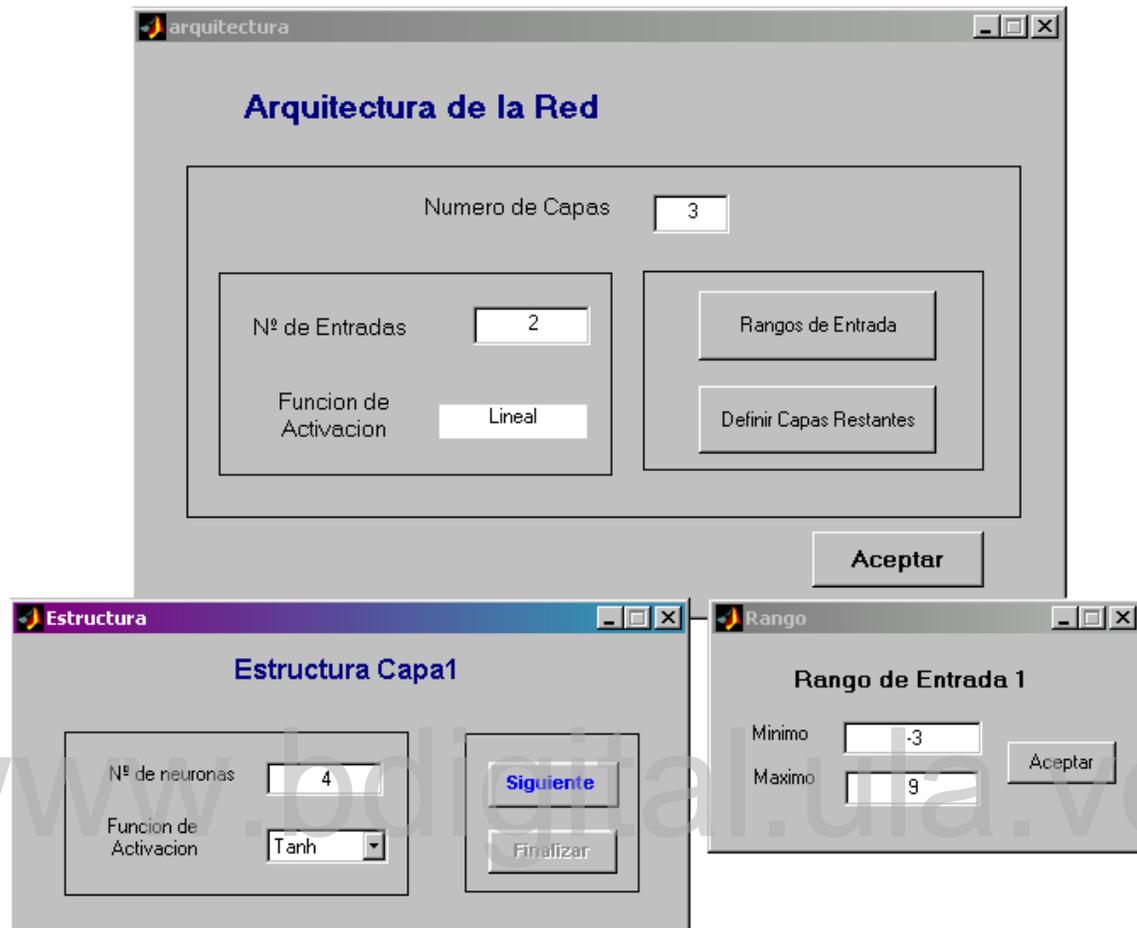


Figura C.2: Definición de la Arquitectura neuronal

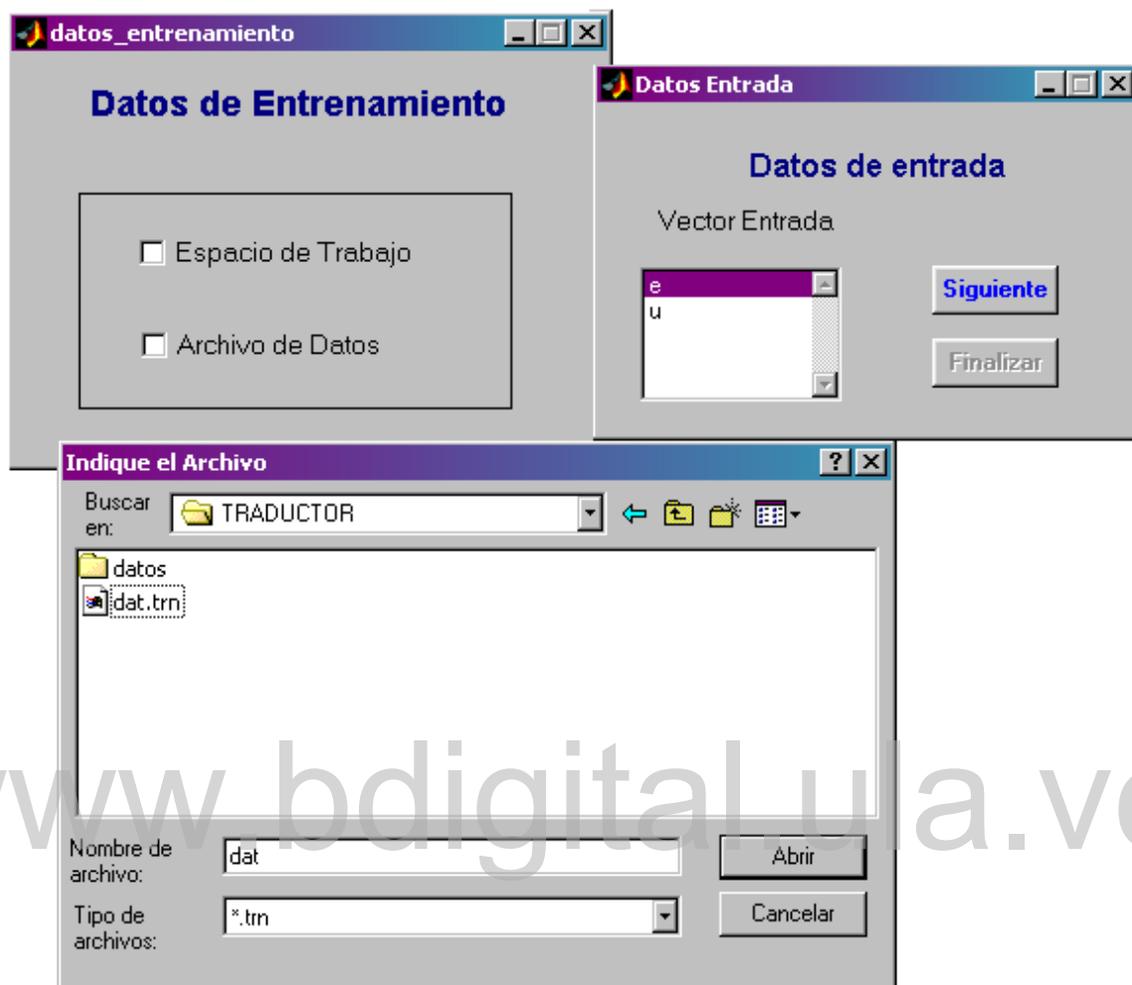


Figura C.3: Declaración de los Datos de Entrenamiento

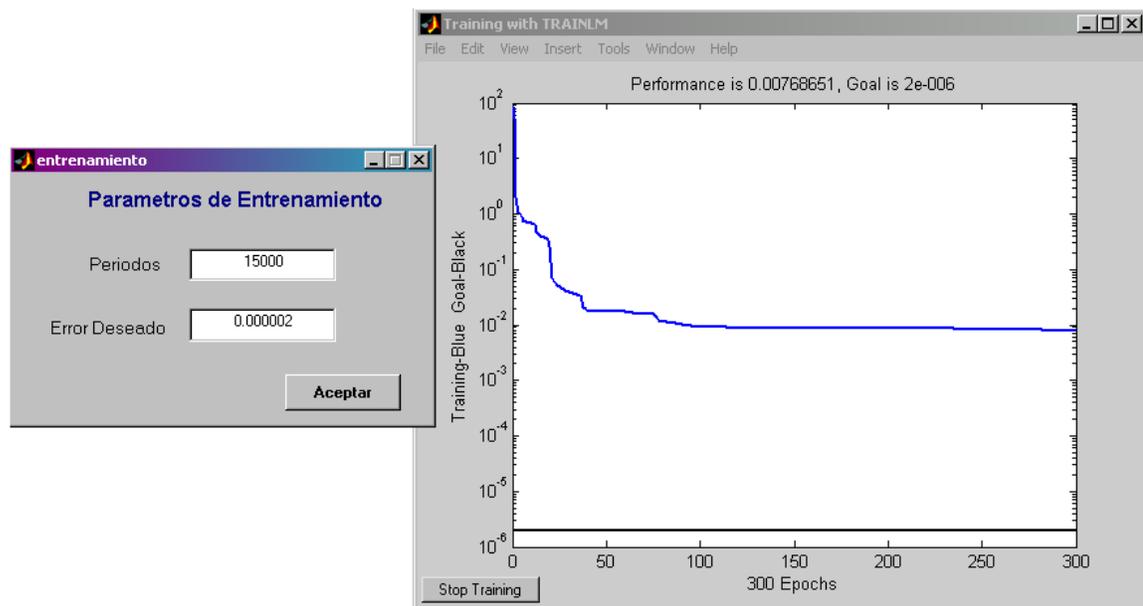


Figura C.4: Proceso de Entrenamiento de la Red Neuronal

## C.2 Generador del Código Fuente en Texto Estructurado

```
% Apertura del Archivo
```

```
fid=fopen('red.txt','w');
```

```
% Escritura de los Pesos de las Conexiones en la Red Neuronal%
```

```
max=nneuronas(1);
```

```
acum=max+(ncapas-1);
```

```
for i=2:1:length(nneuronas)
```

```
    if (max<nneuronas(i))
```

```
        max=nneuronas(i);
```

```
end
if (i<length(nneuronas))
    acum=acum+nneuronas(i);
end
end
for i=1:1:acum
    for j=1:1:max
        cadena=strcat('Pesos[' ,num2str(i-1), ', ', num2str(j-1), ']:= ',
            num2str(pesos(i,j)), ', '\n');
        fprintf(fid,cadena);
    end
end
end
% Escritura de la Arquitectura de la Red%

for i=1:1:length(fd)
    cadena=strcat('fdt[' ,num2str(i-1), ']:= ', num2str(fd(i)), ', '\n');
    fprintf(fid,cadena);
end

cadena=strcat('ncapas:= ', num2str(ncapas), ', '\n');
fprintf(fid,cadena);

for i=1:1:length(nneuronas)
    cadena=strcat('nneuronas[' ,num2str(i-1), ']:= ', num2str(nneuronas(i)), ', '\n');
    fprintf(fid,cadena);
```

```
    if (i==length(nneuronas))
        cadena=strcat('nneuronb[',num2str(i-1),']:=',num2str(nneuronas(i)),';\n');
    else
        cadena=strcat('nneuronb[',num2str(i-1),']:=',num2str(nneuronas(i)+1),';\n')
    end
    fprintf(fid,cadena);
end

fprintf(fid,'\nfor i:=0 to 11 do');
fprintf(fid,'\n\tfor j:=0 to 11 do');
fprintf(fid,'\n\t\ttnentrada[i,j]:=0;');
fprintf(fid,'\n\t\ttvalor_final[i,j]:=0;');
fprintf(fid,'\n\t\tend_for;');
fprintf(fid,'\nend_for;');

for i=1:1:nneuronas(1)
    cadena=strcat('\nnentrada[0,',num2str(i-1),']:=1;\n');
    fprintf(fid,cadena);
end

% Escritura del Funcionamiento General de Cualquier Red Neuronal

fprintf(fid,'aux:=0;');
```

```
fprintf(fid,'\\nfor i:=0 to (ncapas-2) do');
fprintf(fid,'\\n\\tfor j:=0 to (nneuronas[i+1]-1) do');
fprintf(fid,'\\n\\t\\tfor k:=0 to (nneuronb[i]-1) do');
fprintf(fid,'\\n\\t\\t\\tif (k=(nneuronb[i]-1))then');
fprintf(fid,'\\n\\t\\t\\t\\tif (j=0) then');
fprintf(fid,'\\n\\t\\t\\t\\t\\tnentrada[i,k]:=1;');
fprintf(fid,'\\n\\t\\t\\t\\t\\tend_if;');
fprintf(fid,'\\n\\t\\t\\tend_if;');
fprintf(fid,'\\n\\t\\t\\t\\tnentrada[(i+1),j]:=nentrada[i+1,j]+
        nentrada[i,k]*Pesos[k+aux,j];');
fprintf(fid,'\\n\\t\\tend_for;');
fprintf(fid,'\\n\\t\\tcase fdt[i+1] of');
fprintf(fid,'\\n\\t\\t\\t1: x:=nentrada[i+1,j];');
fprintf(fid,'\\n\\t\\t\\t\\tif (x>20) then');
fprintf(fid,'\\n\\t\\t\\t\\t\\tvalor_final[i+1,j]:=1;');
fprintf(fid,'\\n\\t\\t\\t\\t\\t\\tnentrada[i+1,j]:=valor_final[i+1,j];');
fprintf(fid,'\\n\\t\\t\\t\\t\\telse');
fprintf(fid,'\\n\\t\\t\\t\\t\\t\\tif (x<-20) then');
fprintf(fid,'\\n\\t\\t\\t\\t\\t\\tvalor_final[i+1,j]:=-1;');
fprintf(fid,'\\n\\t\\t\\t\\t\\t\\t\\tnentrada[i+1,j]:=valor_final[i+1,j];');
fprintf(fid,'\\n\\t\\t\\t\\t\\t\\telse');
fprintf(fid,'\\n\\t\\t\\t\\t\\t\\t\\tJSR(exponencial,1,x);');
fprintf(fid,'\\n\\t\\t\\t\\t\\t\\t\\t\\tauxt:=exp;');
fprintf(fid,'\\n\\t\\t\\t\\t\\t\\t\\t\\tx:=-nentrada[i+1,j];');
fprintf(fid,'\\n\\t\\t\\t\\t\\t\\t\\tJSR(exponencial,1,x);');
fprintf(fid,'\\n\\t\\t\\t\\t\\t\\t\\t\\tvalor_final[i+1,j]:=(auxt-exp)/(auxt+exp);');
```

