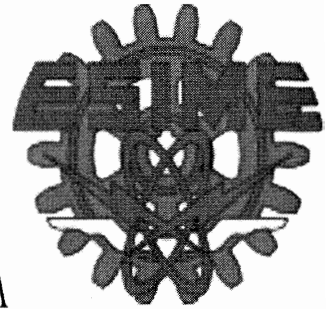


INSTITUTO POLITECNICO NACIONAL
ESCUELA SUPERIOR DE INGENIERIA MECANICA Y ELECTRICA
UNIDAD ZACATENCO.



DISEÑO Y DESARROLLO DE UN SIMULADOR
PARA UNA PLANTA VIRTUAL, POR MEDIO DE
REDES NEURONALES.

TESIS

PARA OBTENER EL GRADO DE

Ingeniero en Control y Automatización.

Asesor técnico: Salvador Cruz Del camino.

Asesor estructural: Enrique López Santini.

Presentan:

López Padilla A-ling Guadalupe.

García Correa Diego.

INSTITUTO POLITECNICO NACIONAL
ESCUELA SUPERIOR DE INGENIERIA MECANICA Y ELECTRICA
UNIDAD PROFESIONAL " ADOLFO LOPEZ MATEOS"

TEMA DE TESIS

**QUE PARA OBTENER EL TITULO DE
POR LA OPCION DE TITULACION
DEBERA(N) DESARROLLAR**

**INGENIERO EN CONTROL Y AUTOMATIZACIÓN
PROYECTO DE INVESTIGACIÓN SIP 20071510
C. A-LING GUADALUPE LÓPEZ PADILLA
C.- DIEGO GARCÍA CORREA**

**"DISEÑO Y DESARROLLO DE UN SIMULADOR PARA UNA PLANTA VIRTUAL, POR MEDIO
DE REDES NEURONALES"**

**DESARROLLAR UNA HERRAMIENTA DE SOFTWARE QUE PERMITA SIMULAR EL
COMPORTAMIENTO DINÁMICO DE UNA PLANTA DE PROCESO, APLICANDO REDES
NEURONALES, CON EL FIN DE APOYAR ACTIVIDADES ACADÉMICAS Y DE
INVESTIGACIÓN EN EL ÁREA DE CONTROL.**

- **INTRODUCCIÓN**
- **REDES NEURONALES**
- **CASOS DE ESTUDIO**
- **ANÁLISIS DE COSTOS**
- **CONCLUSIONES**
- **ANEXOS.**

MÉXICO D.F., 06 DE JUNIO DE 2008.


ASESORES



ING. RICARDO HURTADO RANGEL.



M. EN C. SALVADOR CRUZ DEL CAMINO.



ING. ENRIQUE LÓPEZ PADILLA

**ING. JOSÉ ÁNGEL MEJÍA DOMÍNGUEZ
JEFE DEL DEPARTAMENTO ACADÉMICO
DE INGENIERÍA EN CONTROL Y AUTOMATIZACIÓN.**

RESUMEN

En este trabajo se describe el diseño y programación de una herramienta de software que permita simular el comportamiento dinámico de una planta de proceso, mediante redes neuronales.

El proyecto se realiza en cuatro etapas: La primera consiste en realizar pruebas a la planta física, con el objeto de obtener los parámetros que permiten describir el comportamiento en lazo abierto; la segunda trata del diseño y entrenamiento de la red neuronal; tercera etapa, elaboración del modelo del proceso e implementación de la red neuronal en el mismo; finalmente la última etapa consiste en la validación de los resultados generados por el modelo.

Los resultados de la simulación que presenta el modelo de la planta en lazo abierto son satisfactorios, ya que las respuestas son cercanas o idénticas, en comparación con las obtenidas experimentalmente con la planta.

De igual forma, se presenta el análisis de costos del proyecto, con la finalidad de presentar las ventajas que representa el tener una planta de proceso simulada, en comparación con una planta Europea física de costoso mantenimiento.

ABSTRACT

This work describes the design and programming of a tool software which allows simulate the dynamic behavior of a process plant, by means of neural networks.

The project is realized in four stages: the first one consists of realizing tests to the physical plant, in order to obtain the parameters that allow to describe the behavior in opened bow; the second one treats of the design and training of the neural network; third stage, elaboration of the model of the process and implementation of the neural network in the same one; finally the last stage consists of the validation of the results generated by the model.

Based on the results from the simulation the model of the plant obtained in opened bow shows that this one is satisfactory, since the answers are nearby or identical, in comparison with the ones obtained experimentally from the plant.

Of equal form, one presents the analysis of costs of the project, with the purpose of presenting the advantages that represents to have a process plant simulated, in comparison with an European physical plant of costly maintenance.

ÍNDICE GENERAL

RESUMEN	i
ABSTRACT	ii
ÍNDICE DE FIGURAS	v
ÍNDICE DE TABLAS	vi
GLOSARIO	vii
1. CAPÍTULO I INTRODUCCIÓN	8
1.1 Antecedentes.....	8
1.2 Objetivo.....	9
1.3 Justificación.....	9
1.4 Estado del arte.....	9
1.5 Alcance.....	12
1.6 Estructura de la tesis.....	13
2. CAPÍTULO II REDES NEURONALES	15
2.1 El modelo biológico.....	15
2.1.1 Estructura de la neurona.....	15
2.1.2 Naturaleza biológica de una red neuronal.....	16
2.2 Redes Neuronales Artificiales.....	18
2.2.1 Introducción a las Redes Neuronales Artificiales (RNA).....	19
2.2.2 Estructura del modelo neuronal artificial.....	20
2.2.3 Realización de las Redes Neuronales Artificiales.....	23
2.2.4 Características de las Redes Neuronales Artificiales.....	24
2.3 Ventajas de las Redes Neuronales Artificiales.....	42
2.4 Aplicaciones de las redes neuronales.....	45
2.5 Herramientas de software para desarrollo.....	48
3. CAPÍTULO III DESCRIPCIÓN DEL DESARROLLO DEL MODELO DE LA PLANTA	49
3.1 Diseño y desarrollo de un simulador para una planta virtual, por medio de redes neuronales.....	49
3.2 Metodología.....	49
3.2.1 Análisis de la planta PCT9.....	50
3.2.2 Diseño de las pruebas.....	51

<i>Con lo resultados de la tabla 3.2 se afirma, la no linealidad de la planta de nivel, causada por la presión hidrostática que se genera, a medida que el nivel se incrementa.</i>	53
3.2.3 <i>Análisis de resultados</i>	53
3.2.4 <i>Implementación de la red neuronal</i>	54
3.2.5 <i>Entrenamiento de la red</i>	56
3.2.6 <i>Validación de resultados</i>	57
3.2.7 <i>Generación de la red neuronal en Simulink</i>	58
3.2.8 <i>Validación de resultados de la red neuronal en Simulink</i>	58
3.3 <i>Simulación del proceso de nivel con la red neuronal</i>	59
3.4 <i>Verificación de la dinámica de la planta de nivel</i>	61
4. CAPÍTULO IV ANÁLISIS DE COSTOS	68
4.1 <i>Costos de equipo y materiales</i>	69
4.2 <i>Costos de ingeniería</i>	71
4.3 <i>Otros costos</i>	71
4.4 <i>Costo total del proyecto</i>	72
4.5 <i>Comparación de costos</i>	72
5. CAPÍTULO V CONCLUSIONES	73
BIBLIOGRAFÍA	75
A) DESCRICIÓN DE LA PLANTA PCT9	76
B) ROTÁMETROS	80
C) SERVOPOSICIONADOR	83
D) PROGRAMACION EN MATLAB	85
E) CONSTRUCCIÓN DEL MODELO DE LA PLANTA EN SIMULINK	87

ÍNDICE DE FIGURAS

Figura 2.1 Neurona Biológica.....	16
Figura 2.2 Estructura de una red neuronal artificial.	20
Figura 2.3 Función de transferencia hard lim.	22
Figura 2.4 Función de transferencia lineal.....	22
Figura 2.5 Función de transferencia sigmoid.....	23
Figura 2.6 Red monocapa.	25
Figura 2.7 Red monocapa crossbar.	26
Figura 2.8 Red con conexiones hacia adelante.	27
Figura 2.9. Red con conexión hacia adelante / atrás.....	28
Figura 3.1 Representación esquemática de la planta de nivel.....	50
Figura 3.2 Gráfica del flujo de salida vs nivel.....	53
Figura 3.3 Topología de la red neuronal.....	55
Figura 3.4 Gráfica de entrenamiento de la red.....	56
Figura 3.5 Gráfica de comparación de las curvas experimentales y las simuladas. ...	57
Figura 3.6 Representación grafica de la red neuronal.....	58
Figura 3.7 Gráfica de la entrada con respecto a la salida.....	59
Figura 3.8 Diagrama del proceso de la planta de nivel.....	59
Figura 3.9 Prueba de error en estado estacionario.	62
Figura 3.10 Gráfica de la planta de nivel, para un flujo de entrada de 3 L/min.	63
Figura 3.11 Gráfica de la planta de nivel, para un flujo de entrada de 2.2 L/min.	64
Figura 3.12 Gráfica de la planta de nivel, para un flujo de entrada de 1.5 L/min.	64
Figura 3.13 Gráfica del flujo de entrada de 3 L/min.....	65
Figura 3.14 Gráfica de la señal de la planta simulada.	65
Figura 3.15 Gráfica del flujo de entrada de 2.2 L/min.....	66
Figura 3.16 Gráfica de la señal de la planta simulada.	66
Figura 3.17 Gráfica del flujo de entrada de 1.5 L/min.....	67
Figura 3.18. Gráfica de la señal de la planta simulada.	67
Figura A-1 Circuito de entrada del fluido del proceso.....	78
Figura A-2 Circuito de salida del flujo de proceso.	79
Figura B-1 Rotámetro.	81
Figura B-2 Rotámetro Platón.	82
Figura C-1 Pulsos de control para un servoposicionador.	84
Figura D-1 Cuadro de dialogo del bloque Fi.	87
Figura D-2 Cuadro de dialogo del bloque Rotámetro.....	88
Figura D-3 Cuadro de dialogo del bloque Conversión.	88
Figura D-4 Cuadro de dialogo del bloque de Área.	89
Figura D-5 Cuadro de dialogo del bloque Integrador.	89
Figura D-6 Cuadro de dialogo del bloque de Límites de nivel de la planta.	90
Figura D-7 Cuadro de dialogo del bloque Memoria.	90

ÍNDICE DE TABLAS

Tabla 3.1 Resultados de la primera prueba.	52
Tabla 3.2 Relación de nivel – flujo.	52
Tabla 3.3 Vector de entrada.	54
Tabla 3.4 Vector salida.	55
Tabla 3.5 Generación de la red.	55
Tabla 3.6 Entrenamiento de la red.	56
Tabla 3.7 Simulación de la red.	57
Tabla 3.8 Generación de la red en Simulink.	58
Tabla 4.3 Comparativa de precios de Matlab.	69
Tabla 4.4 Costo total del proyecto.	72
Tabla 4.5 Comparación de precios entre las plantas de nivel.	72
Tabla B-1 Clasificación de los sensores de flujo.	80

GLOSARIO

Dinámica. Evolución de la respuesta temporal de un sistema, puede ser representada mediante una gráfica para tener una mejor visualización.

Matlab. Abreviatura de *Matrix Laboratory* (laboratorio de matrices), programa de análisis numérico creado por *The MathWorks*, disponible para las plataformas Unix, Windows y Mac OS X.

M-file. Archivo simple de texto en donde se ingresan comandos a Matlab, cuando se corre un archivo, Matlab lee y ejecuta los comandos escritos en él.

Neurona artificial. Unidad de procesamiento elemental de una red neuronal artificial, que modela el comportamiento de una neurona biológica.

Planta virtual. Unidad experimental de procesos, generada por medio de software y representada de la misma manera en la pantalla de una computadora.

Presión hidrostática. Presión que ejerce el peso de un fluido sobre las paredes, sobre el fondo del recipiente que lo contiene y sobre la superficie de cualquier objeto sumergido en él; en fluidos en reposo, provoca una fuerza perpendicular a las paredes del recipiente o a la superficie del objeto sumergido sin importar la orientación que adopten las caras.

Rotámetro. Medidor de caudal formado por un flotador que se mueve dentro de un tubo vertical, ligeramente cónico, con el extremo angosto hacia abajo, en donde el fluido entra por la parte inferior del tubo. El tubo es de vidrio y lleva grabada una escala lineal para conocer la posición del flotador o, directamente, el caudal.

Simulador. Elemento que reproduce el comportamiento de un sistema en determinadas condiciones, aplicado generalmente para el entrenamiento de quienes deben manejar dicho sistema.

Simulink. Paquete de Software que se ejecuta acompañando a MATLAB para modelar, simular y analizar sistemas dinámicos.

Toolbox. Caja de herramientas, por su traducción al inglés; conjunto de herramientas de software pertenecientes a una librería, útiles para generar aplicaciones de diversas índoles dentro de un programa superior (Matlab).

Válvula solenoide. Dispositivo que regula el paso de líquidos o gases en tubos o conductos, en donde una bobina responde a pulsos eléctricos logrando su apertura ó cierre.

CAPÍTULO 1

INTRODUCCIÓN

1. INTRODUCCIÓN

1.1 Antecedentes

Actualmente en la Escuela Superior de Ingeniería Mecánica y Eléctrica de la unidad Zacatenco, la carrera de Ingeniería de Control y Automatización, los laboratorios de la materia de procesos cuenta con cuatro plantas piloto, como son: la de nivel, temperatura, PH y presión. Sin embargo se han observado deficiencias en la instrumentación, y por ello en su funcionamiento. De las cuatro plantas solo funciona una de ellas, la de nivel; lo que refleja un bajo nivel de rendimiento escolar del alumnado en la materia fundamental de control de procesos.

Por lo anterior se propone el diseño e implementación de una planta de proceso virtual, que reproduzca el comportamiento de la planta física, con la cual se logre controlar y medir todas las variables que intervienen en el proceso.

La forma para la construcción de la planta virtual es efectuada por redes neuronales; en la actualidad las redes neuronales son una herramienta computacional versátil utilizada en una variedad de aplicaciones; en la ingeniería, una de las ventajas que ofrecen, es evitar la elaboración de ecuaciones diferenciales y balances de materia y energía para el desarrollo de modelos matemáticos.

1.2 Objetivo

Desarrollar una herramienta de software que permita simular el comportamiento dinámico de una planta de proceso, aplicando redes neuronales, con el fin de apoyar actividades académicas y de investigación en el área de control.

1.3 Justificación

Este proyecto es rentable, debido a los beneficios que proporciona, uno de ellos es la cuestión económica, pues para tener una sola planta y su controlador (físicos) se requiere de una inversión de \$300,000.00, a esto también se le debe sumar que en un futuro, la planta necesitará de algún tipo de mantenimiento para su funcionamiento, mientras que una planta virtual no necesita de mantenimiento y en lo económico el costo que se requiere para elaborarla, es 6 veces menor que el de una planta física, lo que significaría tener más de una. Y con esto reflejar una mejor interacción entre los estudiantes y los procesos que en un futuro enfrentarán en las industrias.

Otro punto importante, es el difundir las múltiples ventajas que ofrecen las redes neuronales, no sólo en la ingeniería, sino también en distintas disciplinas, contribuyendo con la investigación para la utilización de nuevas tecnologías o herramientas para el control de procesos.

1.4 Estado del arte

En un principio los controladores de procesos eran diseñados mediante la obtención de modelos matemáticos, para posteriormente ser implementados al proceso para el

cual fueron diseñados, pero a medida que los procesos eran más complejos, el modelo para su control también lo era, es por ello que surgió la necesidad de crear o desarrollar nuevas herramientas que permitieran reducir el tiempo de elaboración del controlador. Una de estas nuevas herramientas son las redes neuronales, las cuales permiten sustituir los modelos matemáticos por algoritmos de aprendizaje.

Las primeras explicaciones teóricas sobre el cerebro humano y el pensamiento habían sido dadas por algunos filósofos griegos, como Platón y Aristóteles. Las mismas ideas sobre el proceso mental también las mantuvieron Descartes y los filósofos empiristas del siglo XVIII.

Sin embargo fue hasta 1936, cuando Alan Turing fue el primero en estudiar el cerebro con la visualización de emplearlo en el mundo computacional; sin embargo los primeros teóricos que concibieron los fundamentos de la computación neuronal fueron Warren McCulloch, un neurofisiólogo, y Walter Pitts, un matemático, quienes en 1943 crearon una teoría acerca de la forma de trabajar con las neuronas.

En 1957, Frank Rosenblatt comenzó el desarrollo de la primera red neuronal artificial: el Perceptrón, esta neurona en la actualidad se utiliza de varias formas para la aplicación como reconocedor de patrones. Mas tarde en 1959, Bernard Widrow y Marcial Hoff, desarrollaron el modelo ADALINE (por sus siglas en ingles Elementos Lineales Adaptivos), que es la primera red neuronal aplicada a un problema real (filtros adaptivos para eliminar ecos en las líneas telefónicas).

Durante los años 60, uno de los mayores investigadores de redes neuronales, Stephen Grossberg, realiza una red llamada Avalancha, que consistía en elementos discretos con actividad que varía con el tiempo que satisface ecuaciones diferenciales

continuas, para resolver actividades tales como reconocimiento continuo del habla y aprendizaje del movimiento de los brazos de un robot.

Posteriormente en 1982 Marvin Minsky y Seymour Papert, del Instituto Tecnológico de Massachussets (MIT) publican el modelo matemático detallado del Perceptrón, así como también la difusión de las importantes limitaciones que presenta, esta neurona para resolver muchos problemas. Por su parte James Anderson, desarrolló un modelo lineal, llamado Asociador lineal, que consistía en unos elementos integradores lineales (neuronas) que sumaban sus entradas. Este modelo se basa en el principio de que las conexiones existentes entre las neuronas son reforzadas cada vez que están activadas.

En Europa y Japón, Kunihiko Fukushima desarrolló el Neocognitrón, que es un modelo de red neuronal para el reconocimiento de patrones visuales. Por otro lado Teuvo Kohonen, un ingeniero electrónico de la universidad de Helsinki, desarrolló un modelo similar al de Anderson, pero independientemente.

En 1985, se formó la INNS (Sociedad Internacional de Redes Neuronales, por sus siglas en inglés) bajo la iniciativa y dirección de Grossberg en los Estados Unidos, Kohonen en Finlandia y Amari en Japón; en menos de dos años, la INNS tenía más de 3,000 socios, es a partir de ese momento cuando surge un gran interés acerca del estudio de las redes neuronales.

Actualmente, son numerosos los trabajos que se realizan y publican cada año, las aplicaciones nuevas que surgen y las empresas que lanzan al mercado productos nuevos, tanto en hardware como en software (sobre todo para la simulación).

En relación con la información que se publica en las revistas especializadas en el área de redes neuronales, cabe destacar como las más interesantes: Neural Networks, revista oficial de las INNS; Network, Computation in Neural System; IEEE Transactions on Neural Networks, publicada por IEEE.

Dentro del entorno europeo, hay que señalar las dos citas anuales de la Sociedad Europea de Redes Neuronales (por sus siglas en inglés ENNS) y, como fuente de planificación política de investigación, el programa ESPRIT, que durante los últimos años ha financiado un gran número de proyectos en la aplicación de redes neuronales.[1]

1.5 Alcance

Este trabajo permite mostrar la metodología para la implementación de una red neuronal en el modelo de una planta de proceso.

Para este trabajo el caso de estudio es: una planta de nivel, en la cual se muestra como una red neuronal es quien genera la dinámica en la planta virtual.

Para crear la planta virtual, se desarrolla una metodología que se muestra a lo largo de la tesis.

1. **Pruebas físicas.** Consisten en el estudio de la planta para analizar su dinámica, visualizar las variables manipuladas y controladas, así como las perturbaciones que afectan al proceso.

2. **Diseño y entrenamiento de la red.** Ésta etapa está basada en la selección de una red que posea las características apropiadas para su implementación en el modelo de la planta de proceso virtual, como son: su topología, su tipo de aprendizaje, y la forma en que presenta la salida.
3. **Elaboración del modelo de proceso.** El modelo es desarrollado por medio de Matlab, utilizando una de sus herramientas (Simulink). Es construido a partir de la visualización de los datos obtenidos en la primera etapa e implementando la red neuronal.
4. **Validación de los resultados.** Ésta última etapa comprueba si los resultados obtenidos por la planta virtual se asemejan a los esperados.

1.6 Estructura de la tesis

El capítulo 1 es la introducción, en donde se describen los antecedentes del proyecto, así como su objetivo, justificación, el estado del arte y la metodología desarrollada a lo largo de la tesis.

El capítulo 2 se analizan los fundamentos de las redes, partiendo del modelo biológico de neurona hasta llegar a la descripción de la estructura de las redes neuronales artificiales, haciendo el estudio de las características que diferencian los distintos modelos de redes neuronales: su topología, mecanismo de aprendizaje y naturaleza de información. En este capítulo se hace referencia de algunos modelos existentes de redes, pero haciendo un mayor énfasis en la red Backpropagation.

El capítulo 3 desarrolla la metodología que se utiliza para generar la red neuronal a utilizar, para posteriormente implementarla en la creación de los modelos de procesos. De igual manera se presenta el caso de estudio del proceso de nivel.

El capítulo 4 presenta el grado de rentabilidad con que cuenta el proyecto por medio de un análisis de costos, así como las ventajas que se obtienen al implementarlo, en el aspecto económico.

El capítulo 5 hace énfasis en los resultados obtenidos, y si éstos cumplen con el objetivo de simular una planta de proceso.

CAPÍTULO 2

REDES NEURONALES

2. REDES NEURONALES

2.1 *El modelo biológico*

La teoría de las redes neuronales artificiales está inspirada en la estructura y funcionamiento de los sistemas nerviosos, donde la neurona es el elemento fundamental. Existen neuronas de diferentes formas, tamaños y longitudes. Estos atributos son importantes para determinar la función y utilidad de la neurona. La clasificación de estas células en tipos estándar ha sido realizada por muchos neuroanatomistas.

2.1.1 *Estructura de la neurona*

Una neurona es una célula viva, y como tal, contiene los mismos elementos que forman parte de todas las células biológicas. Además contienen elementos característicos que las diferencian.

En general, una neurona consta de un cuerpo celular más o menos esférico, de 5-10 micras de diámetro, del que salen una rama principal, el axón, y varias ramas más cortas, llamadas dendritas. A su vez, el axón puede tener ramas entorno a su punto de arranque, y con frecuencia se ramifica extensamente cerca de su extremo.

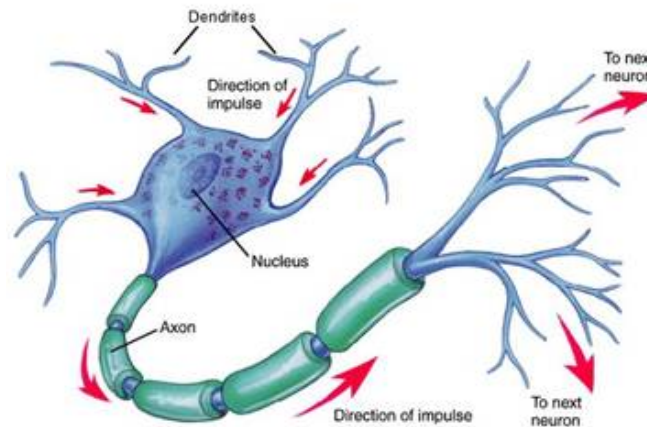


Figura 2.1 Neurona Biológica.

Una de las características que diferencian a las neuronas del resto de las células vivas, es la capacidad de comunicarse entre sí. En términos generales, las dendritas y el cuerpo celular reciben señales de entrada; el cuerpo celular las combina, e integra y emite señales de salida. El axón transporta esas señales a las terminales axónicas que se encargan de distribuir información a un nuevo conjunto de neuronas. Por lo general, una neurona recibe información de miles de otras neuronas y, a su vez, envía información a miles de neuronas más.[1]

2.1.2 Naturaleza biológica de una red neuronal

Las señales que se utilizan, son de dos tipos distintos de naturaleza: *eléctrica* y *química*, la señal generada por la neurona y transportada a lo largo del axón es un impulso eléctrico, mientras que la señal que se transmite entre las terminales axónicas de una neurona y las dendritas de las neuronas siguientes es de origen químico; concretamente, se realiza mediante moléculas de sustancias transmisoras (neurotransmisores) que fluyen a través de contactos especiales, llamados *sinapsis*,

que tienen la función de receptor y están entre las terminales axónicas y las dendritas de la neurona siguiente.

La generación de las señales eléctricas está íntimamente relacionada con la composición de la membrana celular. La neurona como todas las células, es capaz de mantener en su interior un líquido cuya composición difiere principalmente en la composición del líquido exterior. La diferencia se da en relación con la concentración de los iones *sodio* y *potasio*. El medio externo es 10 veces más rico en potasio que el medio interno, mientras que el medio interno es 10 veces más rico en sodio que el medio externo.

Esta diferencia de concentración de iones sodio y potasio en cada lado de la membrana, producen una diferencia de potencial de aproximadamente 70mV negativos en el interior de la célula. A ese potencial se le conoce con el nombre de *potencial de reposo* de la célula nerviosa, cuando la neurona permanece en ese estado, se dice que esta en estado de reposo.

La llegada de señales procedentes de otras neuronas a través de las dendritas (recepción de neurotransmisores) actúa acumulativamente, bajando ligeramente el valor del potencial de reposo. Dicho potencial modifica la permeabilidad de la membrana, de manera que cuando llega a cierto valor crítico comienza una entrada masiva de iones sodio que invierten la polaridad de la membrana.

La inversión del voltaje de la cara interior de la membrana cierra el paso a los iones sodio y abre paso a los iones potasio hasta que se restablece el equilibrio en reposo. La inversión del voltaje, es conocida como *potencial de acción*, se propaga a lo largo del axón y, a su vez, provoca la emisión de los neurotransmisores en las terminales axónicas.

Después de un pequeño periodo refractario, puede seguir un segundo impulso. El resultado de todo esto es la emisión por parte de la neurona de trenes de impulsos cuya frecuencia varía en función de la cantidad de neurotransmisores recibidos.

Existen dos tipos de sinapsis:

- a) *Sinapsis excitadoras*: cuyos neurotransmisores provocan disminuciones de potencial en la membrana de la célula post-sináptica, facilitando la generación de los impulsos a mayor velocidad.
- b) *Sinapsis inhibitoras*: cuyos neurotransmisores tienden a estabilizar el potencial de la membrana, dificultando la emisión de los impulsos.

Casi todas las neuronas reciben entradas procedentes de sinapsis excitadoras e inhibitoras; en cada instante, alguna de ellas estarán activas y en otras se hallarán en reposo; la suma de los efectos excitadores e inhibidores determina si la célula será o no estimulada; es decir, si emitirá o no un tren de impulsos y a que velocidad.[2]

2.2 Redes Neuronales Artificiales

El cerebro humano es un computador notable. Gran cantidad de investigaciones acerca de su funcionamiento, se realizan a cada momento. Es así como desde hace más de 30 años se desarrolla la teoría de las Redes Neuronales Artificiales (RNA), las cuales emulan las redes neuronales biológicas, y que han sido utilizadas para resolver diversos problemas, aprendiendo estrategias de solución, basadas en ejemplos de comportamiento típico de patrones.[3]

2.2.1 Introducción a las Redes Neuronales Artificiales (RNA)

Una Red Neuronal Artificial es un modelo de procesamiento de información que es inspirado por el modo de un sistema nervioso biológico, tal como el cerebro procesa información.

Físicamente puede ser tanto software como hardware, para el primer caso son elaborados con algún lenguaje de programación, como C++, Java, Visual Basic, etc.; el hardware por su parte, pueden ser desde simples conexiones de resistencias o arreglos con amplificadores operacionales, hasta la combinación de software-hardware como es el caso de la programación de circuitos integrados.

Las Redes Neuronales Artificiales, como la gente, aprenden con ejemplos. Una Red Neuronal Artificial es configurada para una aplicación específica, tal como el reconocimiento de patrones o clasificación de datos, a través de un proceso de aprendizaje. Aprender en sistemas biológicos implica ajustes para las conexiones sinópticas que existen entre las neuronas, de igual manera lo hace una Red Neuronal Artificial. Las Redes Neuronales Artificiales se han aplicado a un gran número de problemas reales de complejidad considerable. Su ventaja más importante está en resolver problemas que son demasiado complejos para tecnologías convencionales, problemas que no tienen un algoritmo de solución o que su algoritmo de solución es muy difícil de encontrar.

En general, a causa de su abstracción del cerebro biológico, las Redes Neuronales Artificiales son aptas para resolver problemas que la gente puede resolver, pero las computadoras no pueden. Estos problemas incluyen reconocimiento de patrones y pronósticos (los cuales requieren el reconocimiento de tendencias de datos).

El modelo de una red neuronal artificial es construido averiguando cuales son los elementos relevantes del sistema, y simplificándolos, debido a que la cantidad de información de la que se dispone puede ser excesiva o redundante. La elección adecuada de una red se basa en sus características, una estructura conveniente, el tipo de aprendizaje, etc., es el procedimiento convencional.

2.2.2 Estructura del modelo neuronal artificial

Los elementos básicos de un modelo neuronal biológico son las neuronas, que se agrupan en conjuntos compuestos por millones de ellas organizadas en capas, constituyendo un sistema con funcionalidad propia. Un conjunto de estos subsistemas da lugar a un sistema global (el sistema nervioso, en el caso biológico). En la realización de un sistema neuronal artificial puede establecerse una estructura jerárquica similar. El elemento esencial de partida es la neurona artificial, que se organiza en capas; varias capas constituirán una red neuronal; y, por ultimo, una red neuronal (o un conjunto de ellas), junto con las interfaces de entrada y salida, mas los módulos convencionales adicionales necesarios, constituirán el sistema global de proceso. En la figura 2.2 se muestra el modelo de una red neuronal, cada modelo se compone de unidades llamadas neuronas. Cada neurona recibe una serie de entradas a través de interconexiones y emite una salida. Esta salida viene dada por tres funciones:

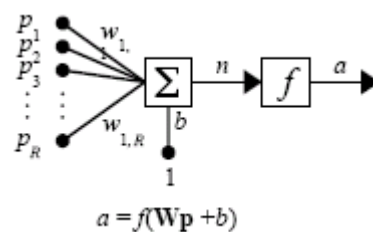


Figura 2.2 Estructura de una red neuronal artificial.

1) Entradas y salidas. Las variables de entrada p y salida n pueden ser binarias (digitales) o continuas (analógicas), dependiendo del modelo y aplicación. Así, para tareas de clasificación tiene salidas digitales, mientras que para un problema de ajuste funcional de una aplicación multivariable continua, se tendrá salidas continuas pertenecientes a un cierto intervalo.

2) Peso sináptico. Es la intensidad de interacción entre la neurona anterior y la siguiente. Dada una entrada positiva (procedente de un sensor o simplemente la salida de otra neurona), si el peso W es positivo excitará a la siguiente neurona, pero si el peso es negativo tenderá a inhibirla.

3) Regla de propagación. También conocida como función de excitación, consiste en la sumatoria de cada entrada p por el peso W de su interconexión (valor neto). Si el peso es positivo, la conexión se denomina excitatoria; si es negativo, se denomina inhibitoria.

4) Función de activación o función de transferencia f . Se aplica al valor devuelto por la función de activación. Se utiliza para acotar la salida de la neurona y generalmente viene dada por la interpretación que quiera dar a las salidas. Las funciones de transferencia más utilizadas son:

- **Función de transferencia hard-limit:** el límite de salida de la neurona, será 0, si el argumento de entrada de la red n es menor que 0, o será 1, si n es mayor o igual que cero, podemos utilizar esta función para crear neuronas que hagan clasificaciones de decisiones (ver Fig. 2.3).

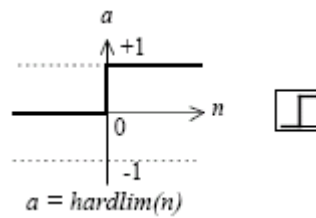


Figura 2.3 Función de transferencia hard lim.

- **Función de transferencia lineal:** las neuronas que emplean este tipo de función son usadas como aproximadores lineales en los “Filtros lineales”.

También la función lineal a tramos se puede considerar como una lineal saturada en sus extremos, es de gran sencillez computacional y resulta mas plausible desde un punto de vista biológico, ya que, las neuronas se activan mas a mayor excitación, hasta saturarse a la máxima respuesta que pueden proporcionar (ver Fig. 2.4).

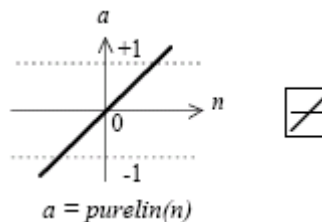


Figura 2.4 Función de transferencia lineal.

- **Función de transferencia sigmoid:** toma la entrada, la cual puede o no, tener valores entre más y menos infinito, además de comprimir la salida dentro del rango de 0 a 1, una de sus aplicaciones es cuando los algoritmos de aprendizaje requieren que la función de activación cumpla la condición de ser derivable (ver Fig. 2.5).

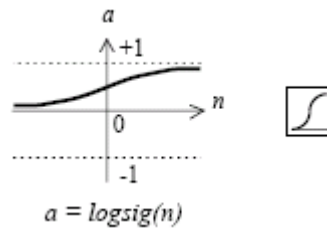


Figura 2.5 Función de transferencia sigmoide.

5) Función de salida. Esta función proporciona la salida global de la neurona en función de su estado de activación actual. Muy frecuentemente la función de salida es simplemente la identidad $F(x)=x$, de modo que el estado de activación de la neurona se considera como la propia salida.[4][2]

2.2.3 Realización de las Redes Neuronales Artificiales

La elaboración de una Red Neuronal Artificial puede ser de distintas maneras, a continuación se presentan dos de ellas, así como el objetivo de cada una.

- La realización más simple e inmediata consiste en simular la red sobre una computadora convencional mediante un software específico. Es un procedimiento rápido, poco costoso e insustituible por el momento para realizar el entrenamiento y evaluación de las redes, pero cuya mayor desventaja radica en el hecho de que se intentan simular redes con un alto grado de paralelismo sobre máquinas que ejecutan secuencialmente las operaciones.
- Una tercera aproximación radicalmente distinta es la realización de redes neuronales mediante su implementación por uno o varios circuitos

integrados específicos. Se intenta de esta manera construir un elemento o conjunto de elementos que se comporten lo más similarmente posible a cómo lo haría una red neuronal (llamados chips neuronales). Las neuronas y las conexiones se emulan con dispositivos específicos, de forma que la estructura del circuito integrado refleja la arquitectura de la red.

Se consigue de esta forma realizaciones que funcionan a alta velocidad, permitiendo en muchas ocasiones el proceso en tiempo real, pero a costa de una pérdida notable de flexibilidad.

2.2.4 Características de las Redes Neuronales Artificiales

Existen cuatro aspectos que caracterizan una red neuronal: su topología, el mecanismo de aprendizaje, tipo de asociación realizada entre la información de entrada y de salida, y por último, la forma de representación de estas informaciones.

2.2.4.1 Topología de las redes neuronales

La topología o arquitectura de las redes neuronales consiste en la organización y disposición de las neuronas en la red formando capas o agrupaciones de neuronas más o menos alejadas de la entrada y salida de la red. En este sentido, los parámetros fundamentales de la red son: *el número de capas, el número de neuronas por capa, el grado de conectividad y el tipo de conexiones entre neuronas.*

Cuando se realiza una clasificación de las redes en términos topológicos, se suele distinguir entre las redes con una sola capa o nivel de neuronas y las redes con múltiples capas.

Redes monocapa

En las redes monocapa, se establecen conexiones laterales entre las neuronas que pertenecen a la única capa que constituye la red, también pueden existir conexiones autorrecurrentes (salida de una neurona conectada a su propia entrada).

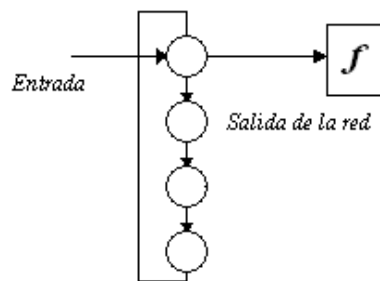


Figura 2.6 Red monocapa.

Una topología equivalente a la de las redes monocapa es la denominada topología *crossbar* (barras cruzadas). Una red de este tipo consiste en una matriz de terminales (de entrada y salida) o barras que se cruzan en unos puntos a los que se les asocia un peso. Esta representación *crossbar* suele utilizarse como etapa de transición cuando se pretende implementar físicamente una red monocapa, puesto que es relativamente sencillo desarrollar como hardware una estructura como la indicada (por ejemplo, las barras cruzadas serían cables, y los puntos de conexión, resistencias cuyos valores representarían los pesos de la red).

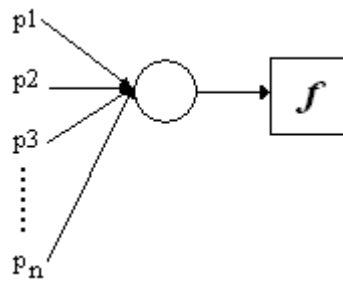


Figura 2.7 Red monocapa crossbar.

Finalmente, hay que indicar que las redes monocapa se utilizan típicamente en tareas relacionadas con lo que se conoce como autoasociación; por ejemplo, para regenerar informaciones de entradas que se presentan a la red, incompletas o distorsionadas.

Redes multicapa

Las redes multicapa son aquellas que disponen de conjuntos de neuronas agrupadas en varios niveles o capas. En estos casos, una forma para distinguir la capa a la que pertenece una neurona, consistirá en fijarse en el origen de las señales que recibe a la entrada y el destino de la señal de salida. Normalmente, todas las neuronas de una capa reciben señales de entrada de otra capa anterior, más cercana a las de entrada de la red, y envían las señales de salida a una capa posterior, más cercana a la salida de la red. A estas conexiones se les denomina conexiones hacia adelante o feedforward.

Sin embargo, en un gran número de estas redes también existe la posibilidad de conectar las salidas de las neuronas de capas posteriores a las entradas de las capas anteriores, a estas conexiones se les denomina conexiones hacia atrás o feedback.

Éstas dos posibilidades permiten distinguir entre dos tipos de redes con múltiples capas: las redes con conexiones hacia adelante o redes feedforward, y las redes que disponen de conexiones tanto hacia adelante como hacia atrás o redes feedforward – feedback.

Redes con conexiones hacia adelante

En estas redes, todas las señales neuronales se propagan hacia adelante a través de las capas de la red. No existen conexiones hacia atrás y normalmente tampoco autorrecurrentes (salida de una neurona aplicada a su propia entrada), ni laterales (salida de una neurona aplicada a la entrada de neuronas de la misma capa).

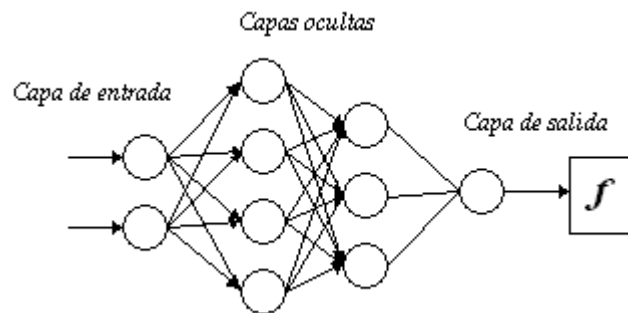


Figura 2.8 Red con conexiones hacia adelante.

Las redes con conexiones hacia adelante más conocidas son: Perceptrón, Adaline, Madaline, Memoria Lineal Adaptiva (LAM por sus siglas en inglés), Backpropagation. Todas ellas son especialmente útiles en aplicaciones de reconocimiento o clasificación de patrones.

Redes con conexiones hacia adelante y hacia atrás

En este tipo de redes circula información tanto *hacia adelante* (forward) como *hacia atrás* (backward) durante el funcionamiento de la red. Para que esto sea posible, existen *conexiones hacia adelante* (feedforward) y *conexiones hacia atrás* (feedback) entre las neuronas.

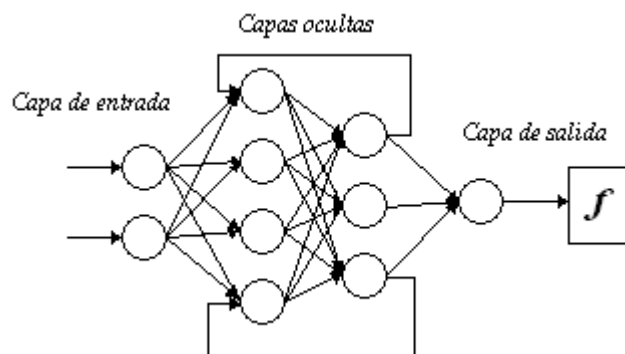


Figura 2.9. Red con conexión hacia adelante / atrás.

En general, suelen ser bicapa (dos capas), existiendo por tanto dos conjuntos de pesos: los correspondientes a las *conexiones hacia adelante* de la primera capa (capa de entrada) hacia la segunda (capa de salida) y los de las *conexiones hacia atrás* de la segunda a la primera. Los valores de los pesos de estos dos tipos de conexiones no tienen porque coincidir, siendo diferentes en la mayor parte de los casos.

Este tipo de estructura (bicapa) es particularmente adecuada para realizar una asociación de una información o patrón de entrada (en la primera capa) con otra información o patrón de salida en la segunda capa (lo cual se conoce como heteroasociación), aunque también pueden ser utilizadas para la clasificación de patrones.

Algunas redes de este tipo tienen un funcionamiento basado en lo que se conoce como resonancia, de tal forma que las informaciones en la primera y segunda capas interactúan entre sí hasta que alcanzan un estado estable. Este funcionamiento permite un mejor acceso a las informaciones almacenadas en la red.

Los dos modelos de red *hacia adelante/hacia atrás* de dos capas más conocidos son la red de Resonancia Adaptiva (ART por sus siglas en inglés) y la red de Memoria Asociativa Bidireccional (BAM por sus siglas en inglés).

También en este grupo de redes existen algunas que tienen conexiones laterales entre neuronas de la misma capa. Estas conexiones se diseñan como conexiones excitadoras (con peso positivo), permitiendo la cooperación entre neuronas, o como inhibidoras (con peso negativo), estableciéndose una competencia entre las neuronas correspondientes.

Finalmente, hay que comentar la existencia de un tipo de red *adelante/hacia atrás* multicapa muy particular, denominada *Neocognitron*, en la que las neuronas se disponen en planos superpuestos (capas bidimensionales), lo cual permite que puedan eliminarse las variaciones geométricas (tamaños, giros, desplazamientos) o distorsiones que presenten las informaciones o patrones de entrada a la red.

2.2.4.2 Mecanismo de aprendizaje

El aprendizaje es el proceso por el cual una red neuronal modifica sus pesos en respuesta a una información de entrada. Los cambios que se producen durante el proceso de aprendizaje se reducen a la destrucción, modificación y creación de conexiones entre las neuronas.

En los sistemas biológicos existe una continua creación y destrucción de conexiones. En los modelos de redes neuronales artificiales, la creación de una nueva conexión implica que el peso de la misma tenga un valor distinto de cero. De la misma forma, una conexión se destruye cuando su peso adquiere un valor de cero.

Durante el proceso de aprendizaje, los pesos de las conexiones de la red sufren modificaciones, por tanto se puede afirmar que este proceso ha terminado, cuando los valores de los pesos permanecen estables ($dw / dt = 0$).

Un aspecto importante respecto al aprendizaje en las redes neuronales es el conocer como se modifican los valores de los pesos; es decir, cuales son los criterios que se siguen para cambiar el valor asignado a las conexiones cuando se pretende que la red aprenda una nueva información.

Estos criterios determinan lo que se conoce como la regla de aprendizaje de la red. De forma general, se suelen considerar dos tipos de reglas:

- Redes neuronales con aprendizaje supervisado
- Redes neuronales con aprendizaje no supervisado

La diferencia fundamental entre ambos tipos estriba en la existencia o no de un agente externo (*supervisor*) que controle el proceso de aprendizaje de la red.

Otro criterio que se utiliza para diferenciar las reglas de aprendizaje es considerar si la red puede aprender durante su funcionamiento habitual o si el aprendizaje supone la desconexión de la red; es decir su inhabilitación hasta que el proceso termine.

En el primer caso, se trata de un aprendizaje *en línea* (on line), mientras que el segundo es lo que se conoce como aprendizaje *fuera de línea* (off line). Cuando el aprendizaje es *fuera de línea*, se distingue entre una fase de aprendizaje o entrenamiento y una fase de operación o funcionamiento, existiendo un conjunto de datos de entrenamiento y un conjunto de datos de prueba que son utilizados en la correspondiente fase.

En las redes con aprendizaje *fuera de línea*, los pesos de las conexiones permanecen fijos después que termina el entrenamiento de la red. Debido precisamente a su carácter estático, estos sistemas no presentan problemas de estabilidad en su funcionamiento.

En las redes con aprendizaje *en línea*, no se distingue entre la fase de entrenamiento y la de operación, de tal forma que los pesos varían dinámicamente siempre que se presente una nueva información al sistema. En este tipo de redes, debido al carácter dinámico de las mismas, la estabilidad suele ser un aspecto fundamental de estudio.

Redes con aprendizaje supervisado

El aprendizaje supervisado se caracteriza por que el proceso de aprendizaje se realiza mediante un entrenamiento controlado por un agente externo (supervisor, maestro) que determina la respuesta que debería generar la red a partir de una entrada determinada. El supervisor comprueba la salida de la red y en el caso de que ésta no coincida con la deseada, se procede a modificar los pesos de las conexiones, con el fin de conseguir que la salida se aproxime a la deseada.

En este tipo de aprendizaje se suelen considerar, a su vez, tres formas de llevarlo a

cabo que dan lugar a los siguientes aprendizajes supervisados:

- Aprendizaje por corrección de error
- Aprendizaje por refuerzo
- Aprendizaje estocástico

Aprendizaje por corrección de error. Consiste en ajustar los pesos de la red en función de la diferencia entre los valores deseados y los obtenidos en la salida de la red; es decir, en función del error cometido en la salida. Una regla o algoritmo simple de aprendizaje por corrección de se muestra en la ecuación 2.1

$$\Delta W_{ji} = \alpha y_i (d_j - y_j) \quad \text{Ecu. 2.1}$$

Donde:

ΔW_{ij} es la variación en el peso de la conexión entre las neuronas i y j

y_i Valor de la salida de la neurona i

d_j Valor de la salida deseado para la neurona j

y_j Valor de la salida obtenido en la neurona j

α Factor de aprendizaje

La *regla delta* o regla del mínimo error cuadrado, también conocida como regla de Widrow y Hoff, es otro tipo de algoritmo, ellos definieron una función que permita cuantificar el error global cometido en cualquier momento durante el proceso de

entrenamiento de la red, lo cual es importante, ya que cuanto más información se tenga sobre el error cometido, más rápido se puede aprender. Este error se expresa en la ecuación 2.2

$$Error_{global} = \frac{1}{2P} \sum_{k=1}^P \sum_{j=1}^N (y_j^{(k)} - d_j^{(k)}) \quad \text{Ecu. 2.2}$$

Donde:

N es el número de neuronas de salida

P es el número de informaciones que debe aprender la red

Otro algoritmo de aprendizaje por corrección de error lo constituye la denominada regla delta generalizada o algoritmo de retro-propagación del error (*error de retropropagación*). Se trata de una generalización de la regla delta, para aplicarla a redes con *conexiones hacia adelante* con capas o niveles internos u ocultos de neuronas que no tienen relación con el exterior.

Aprendizaje por refuerzo. Se trata de un aprendizaje supervisado, mas lento que el anterior, que se basa en la idea de no disponer de un ejemplo completo del comportamiento deseado; es decir, de no indicar durante el entrenamiento exactamente la salida que se desea que proporcione la red ante una determinada entrada.

En el aprendizaje por refuerzo la función del supervisor se reduce a indicar mediante una señal de refuerzo si la salida obtenida en la red se ajusta a la deseada (éxito = +1 o fracaso = -1), y en función de ello se ajustan los pesos basándose en un mecanismo de probabilidades.

Se podría decir que en este tipo de aprendizaje la función del supervisor se asemeja más a la de un crítico (que opina sobre la respuesta de la red) que a la de un maestro (que indica a la red la respuesta concreta que debe generar), como ocurría en el caso de supervisión por corrección de error.

Un ejemplo de algoritmo por refuerzo lo constituye el denominado algoritmo lineal con recompensa y penalización, presentado por Narendra y Thathacher en 1974, que se aplica en redes con conexiones hacia adelante de dos capas cuyas neuronas de salida presentan una función de activación estocástica.

Otro algoritmo conocido es el heurístico-crítico adaptivo, introducido por Barto, Sutton y Anderson en 1983, que se utiliza en redes con *conexiones hacia adelante* de tres capas especialmente diseñadas para que una parte de la red sea capaz de generar un valor interno de refuerzo que es aplicado a las neuronas de salida de la red.

Aprendizaje estocástico. Este tipo de aprendizaje consiste básicamente en realizar cambios aleatorios en los valores de los pesos de las conexiones de la red y evaluar su efecto a partir del objetivo deseado y de distribuciones de probabilidad.

En el aprendizaje estocástico se suele hacer una analogía en términos termodinámicos, asociando la red neuronal con un sólido físico que tiene cierto estado energético. En el caso de la red, la energía de la misma representaría el grado de estabilidad de la red, de tal forma que el estado de mínima energía correspondería a una situación en la que los pesos de las conexiones consiguen que su funcionamiento sea el que más se ajusta al objetivo deseado.

Según lo anterior, el aprendizaje consistiría en realizar un cambio aleatorio de los

valores de los pesos y determinar la energía de la red. Si la energía es menor después del cambio; es decir, si el comportamiento de la red se acerca al deseado, se acepta el cambio. Si, por el contrario, la energía no es menor, se aceptaría el cambio en función de una determinada y preestablecida distribución de probabilidades.

Redes con aprendizaje no supervisado

Las redes con aprendizaje no supervisado, no requieren influencia externa para ajustar los pesos de las conexiones entre sus neuronas. La red no recibe ninguna información por parte del entorno que le indique si la salida generada en respuesta a una determinada entrada es o no correcta, por ello, suele decirse que estas redes son capaces de auto-organizarse.

Estas redes deben encontrar las características, regularidades, correlaciones o categorías que se puedan establecer entre los datos que se presenten en su entrada. Existen varias posibilidades en cuanto a la interpretación de la salida de estas redes, que dependen de su estructura y del algoritmo de aprendizaje empleado. En algunos casos, la salida representa el grado de familiaridad o similitud entre la información que se le está presentando en la entrada y las informaciones que se le han mostrado hasta entonces (en el pasado).

Otro caso, es un establecimiento de categorías, indicando a la salida de la red a que categoría pertenece la información presentada a la entrada, siendo la propia red quien debe encontrar las categorías apropiadas a partir de correlaciones entre las informaciones presentadas.

En cuanto a los algoritmos de aprendizaje no supervisado, en general se suelen

considerar dos tipos que dan lugar a los siguientes aprendizajes:

- Aprendizaje hebbiano
- Aprendizaje competitivo y cooperativo

En el primer caso, normalmente se pretende medir la familiaridad o extraer características de los datos de entrada, mientras que el segundo suele orientarse hacia la clasificación de dichos datos.

Aprendizaje hebbiano

Este tipo de aprendizaje se basa en el siguiente postulado formulado por Donald O. Hebb en 1949: *Cuando un axón de una celda A está suficientemente cerca como para conseguir excitar una celda B y repetida o persistentemente toma parte en su activación, algún proceso de crecimiento o cambio metabólico tiene lugar en una o ambas celdas, de tal forma que la eficiencia de A, cuando la celda a activar es B, aumenta.* Por celda, Hebb entiende un conjunto de neuronas fuertemente conexas a través de una estructura compleja. La eficiencia podría identificarse con la intensidad o magnitud de la conexión; es decir, con el peso.

Se puede decir, por tanto, que el aprendizaje hebbiano consiste básicamente en el ajuste de los pesos de las conexiones de acuerdo con la correlación (multiplicación en el caso de valores binarios +1 y -1) de los valores de activación (salidas) de las dos neuronas conectadas.

$$\Delta W_{ji} = y_i y_j \quad \text{Ecu. 2.3}$$

La ecuación 2.3 responde a la idea de Hebb, puesto que si las dos unidades son activas (positivas), se produce un reforzamiento de la conexión. Por el contrario, cuando una es activa y la otra pasiva (negativa), se produce un debilitamiento de la conexión.

Se trata de una regla de aprendizaje no supervisado, pues la modificación de los pesos se realiza en función de los estados (salidas) de las neuronas obtenidas tras la presentación de cierto estímulo (información de entrada a la red), sin tener en cuenta si se deseaba obtener o no esos estados de activación.

Aprendizaje competitivo y cooperativo

En las redes con aprendizaje competitivo (y cooperativo), suele decirse que las neuronas compiten (y cooperan) unas con otras con el fin de llevar a cabo una tarea dada. Con este tipo de aprendizaje, se pretende que cuando se presente a la red cierta información de entrada, solo una de las neuronas de salida de la red, o una por cierto grupo de neuronas, se active (alcance su valor de respuesta máximo). Por tanto, las neuronas compiten por activarse, quedando finalmente una, o una por grupo, como neurona *vencedora*, quedando anulado el resto, que son forzadas a sus valores de respuesta mínimos.

La competencia entre neuronas se realiza en todas las capas de la red, existiendo en estas neuronas conexiones recurrentes de autoexcitación y conexiones de inhibición (signo negativo) por parte de neuronas vecinas.

Si el aprendizaje es cooperativo, estas conexiones con las vecinas serán de excitación (signo positivo).

El objetivo de este aprendizaje es categorizar los datos que se introducen en la red. De esta forma, las informaciones similares son clasificadas formando parte de la misma categoría, y por tanto deben activar la misma neurona de salida. Las clases o categorías deben ser creadas por la propia red, puesto que se trata de un aprendizaje no supervisado, a través de las correlaciones entre los datos de entrada.

Una forma de aplicar este tipo de aprendizaje fue propuesta por Rumelhart y Zisper en 1985, quienes utilizan redes multicapa dividiendo cada capa en grupos de neuronas, de tal forma que éstas disponían de conexiones inhibitorias con otras neuronas de su mismo grupo, y conexiones excitadoras con las neuronas de la siguiente capa.

En una red de este tipo, después de recibir diferentes informaciones de entrada, cada neurona en cada grupo se especializa en la respuesta a determinadas características de los datos de entrada. En este tipo de redes, cada neurona tiene asignado un peso total, suma de todos los pesos de las conexiones que tiene a su entrada.

El aprendizaje afecta sólo a las neuronas ganadoras (activas), redistribuyendo este peso total entre sus conexiones, sustrayendo una porción a los pesos de todas las conexiones que llegan a la neurona vencedora y repartiendo esta cantidad por igual entre todas las conexiones procedentes de unidades activas.

2.2.4.3 Tipo de asociación entre la información de entrada y salida

Las redes neuronales son sistemas que almacenan cierta información aprendida; esta información se registra de forma distribuida en los pesos asociados a las conexiones entre neuronas. Por tanto, puede imaginarse una red como cierto tipo de memoria que almacena datos de forma estable, y que son guardados en dicha memoria como consecuencia del aprendizaje de la red y que podrán ser leídos a la salida como respuesta a cierta información de entrada, comportándose entonces la red como lo que habitualmente se conoce por memoria asociativa; es decir, cuando se aplica un estímulo (dato de entrada) la red responde con una salida asociada a dicha información de entrada.

Existen dos formas primarias de realizar esta asociación entre entrada/salida que se corresponden con la naturaleza de la información almacenada en la red.

La primera es la denominada *heteroasociación*, que se refiere al caso en el que la red aprende parejas de datos $[(A_1, B_1), (A_2, B_2), \dots (A_N, B_N)]$, de tal forma que cuando se presente cierta información de entrada A, deberá responder generando la correspondiente salida asociada B.

La segunda se conoce como *auto-asociación*, donde la red aprende cierta información $A_1, A_2, \dots A_N$, de tal forma que cuando se le presenta una información de entrada realiza una auto correlación, respondiendo con uno de los datos almacenados, el más parecido al de entrada.

Estos dos mecanismos de asociación dan lugar a dos tipos de redes neuronales: las redes hetero-asociativas y las auto-asociativas. Una *red hetero-asociativa* se

considera aquella que calcula cierta función, que en la mayoría de los casos no se puede expresar analíticamente, entre un conjunto de entradas y un conjunto de salidas, correspondiendo a cada posible entrada una determinada salida. Por otra parte, una *red auto-asociativa* es una red cuya principal misión es reconstruir una determinada información de entrada que se presenta incompleta o distorsionada (le asocia el dato almacenado más parecido).

En realidad, estos dos tipos de modelos de redes no son diferentes en principio, porque una red hetero-asociativa puede siempre ser reducida a una auto-asociativa mediante la concatenación de la información de entrada y su salida (respuesta) asociada, para obtener la información de entrada de la red auto-asociativa equivalente.

También puede conseguirse que una red auto-asociativa se comporte como una hetero-asociativa, simplemente presentando, como entrada parcial de la auto-asociativa, la información de entrada para la hetero-asociativa y haciendo que la red complete la información para producir lo que sería la salida de la red hetero-asociativa equivalente.

Redes hetero-asociativas

Las redes hetero-asociativas, al asociar la información de entrada con diferentes datos de salidas, precisan al menos de 2 capas, una para captar y retener la información de entrada y otra para mantener la salida con la información asociada. Si esto no fuese así, se perdería la información inicial al obtenerse el dato asociado, lo cual no debe ocurrir, ya que en el proceso de obtención de la salida puede ser necesario acceder varias veces a esta información, por lo tanto, deberá permanecer en la capa de entrada.

En cuanto a su conectividad, existen redes hetero-asociativas con *conexiones hacia adelante* (*Perceptrón*, *Retro-propagación*, etc.), redes con *conexiones hacia atrás* (*Art*, *Barn*, etc.) y redes con conexiones laterales (*Cabam*). También hay redes hetero-asociativas con las neuronas dispuestas en capas multidimensionales (*Neocognitron*).

El aprendizaje de este tipo de redes puede ser con supervisión (*Perceptrón*, *Retro-propagación*, etc.) o sin supervisión (*Art*, *Barn*, etc.).

Las redes hetero-asociativas pueden también clasificarse según el objetivo pretendido con su utilización. Así, en algunos casos el objetivo es calcular una función general de su entrada. En otros casos el objetivo es realizar una clasificación, relacionando (mapeando) un gran número de información de entrada con un pequeño número de datos de salida, que representan los conjuntos en los que se pueden clasificar los datos de entrada.

Redes auto-asociativas

Una red auto-asociativa asocia una información de entrada con el ejemplar más parecido de los almacenados conocidos por la red. Esta red puede implementarse con una sola capa de neuronas. Esta capa comienza reteniendo la información inicial presentada a la entrada, y termina representando la información auto-asociada.

Si se quiere mantener la información de entrada y salida, se deberían añadir capas adicionales; sin embargo, la funcionalidad de la red puede conseguirse en una sola capa.

En cuanto a la conectividad en este tipo de redes, existen conexiones laterales entre las neuronas (Hopfield, etc.) y, en algunos casos, conexiones autorrecurrentes (salida de una neurona como entrada de la misma), como ocurre con las redes estado cerebral, la adaptiva de Grossberg, etc.

En relación con el tipo de aprendizaje, habitualmente el utilizado por estas redes es no supervisado (Hopfield, la adaptiva de Grossberg etc.), aunque existe alguna con aprendizaje supervisado (estado cerebral).

Su principal aplicación radica en tareas de filtrado de información para la reconstrucción de datos, eliminando distorsiones. También se utilizan para explorar relaciones entre informaciones similares, para facilitar la búsqueda por contenido en bases de datos y para resolver problemas de optimización.

2.3 Ventajas de las Redes Neuronales Artificiales

Debido a su constitución y a sus fundamentos, las redes neuronales artificiales presentan un gran número de características semejantes a las del cerebro, son capaces de aprender de la experiencia, de generalizar de casos anteriores a nuevos casos, de abstraer características esenciales a partir de entradas que representan información irrelevante, etc. Esto hace que ofrezcan numerosas ventajas y que este tipo de tecnología se aplique en múltiples áreas. Las principales ventajas de las redes neuronales son:

Aprendizaje adaptativo. La capacidad de aprendizaje adaptativo es una de las características más atractivas de las redes neuronales, ya que aprenden a llevar a cabo ciertas tareas mediante un entrenamiento. Como las redes neuronales pueden

aprender a diferenciar patrones mediante ejemplos y entrenamiento, no es necesario que se elaboren modelos a priori, ni se necesita especificar las funciones de distribución de probabilidad.

Las redes neuronales son sistemas dinámicos autoadaptativos, pues son adaptables debido a la capacidad de auto ajuste en los elementos procesadores (neuronas) que componen el sistema. Son dinámicos, pues son capaces de estar constantemente cambiando para adaptarse a las nuevas condiciones.

En el proceso de aprendizaje, los enlaces ponderados de las neuronas se ajustan de manera que se obtengan resultados específicos. Una red neuronal no necesita un algoritmo para resolver un problema, ya que ella puede generar su propia distribución de los pesos de los enlaces mediante el aprendizaje.

Auto organización. Las redes neuronales usan su capacidad de aprendizaje adaptativo para auto organizar la información que reciben durante el aprendizaje y/o la operación. Mientras que el aprendizaje es la modificación de cada elemento que procesa, la auto organización consiste en la modificación de la red neuronal completa para llevar a cabo un objetivo específico.

Esta auto organización provoca la generalización: facultad de las redes neuronales de responder apropiadamente cuando se les presentan datos o situaciones a las que no habían sido expuestas anteriormente. El sistema puede generalizar la entrada para obtener una respuesta. Esta característica es muy importante cuando se tienen que solucionar problemas en los cuales la información de entrada es poco clara; además, permite que el sistema dé una solución incluso cuando la información de entrada está especificada de forma incompleta.

Tolerancia a fallos. Las redes neuronales son los primeros métodos computacionales con la capacidad inherente de tolerancia a fallos. Comparados con los sistemas computacionales tradicionales, los cuales pierden su funcionalidad en cuanto sufren un pequeño error de memoria (cuando el sistema requiere de mayor cantidad de memoria que con la que se cuenta en el sistema), en las redes neuronales, si se produce un fallo en un pequeño número de neuronas, aunque el comportamiento del sistema se ve influenciado, no sufre una caída repentina.

Hay dos aspectos distintos respecto a la tolerancia a fallos: primero, las redes pueden aprender a reconocer patrones con ruido, distorsionados o incompletos, esta es una tolerancia a fallos respecto a los datos. Segundo, pueden seguir realizando su función (con cierta degradación) aunque se destruya parte de la red. La razón por la que las redes neuronales son tolerantes a los fallos es que tienen su información distribuida en las conexiones entre neuronas, existiendo cierto grado de redundancia en este tipo de almacenamiento.

La mayoría de los ordenadores algorítmicos y sistemas de recuperación de datos almacenan cada pieza de información en un espacio único, localizado y direccionable. Las redes neuronales almacenan información no localizada. Por tanto, la mayoría de las interconexiones entre los nodos de la red tendrán unos valores en función de los estímulos recibidos, y se generara un patrón de salida que represente la información almacenada.

Operación en tiempo real. Una de las mayores prioridades en las áreas de aplicación, es la necesidad de realizar grandes procesos con datos de forma muy rápida. Las redes neuronales se adaptan bien a esto debido a su implementación paralela. Para que la mayoría de las redes puedan operar en un entorno de tiempo

real, la necesidad de cambio en los pesos de las conexiones o entrenamiento es mínima.

Fácil inserción dentro de la tecnología existente. Una red individual puede ser entrenada para desarrollar una única y bien definida tarea, debido a que una red puede ser rápidamente entrenada, comprobada, verificada y trasladada a una implementación de hardware de bajo costo, es fácil insertar redes neuronales para aplicaciones específicas dentro de sistemas existentes. De esta manera, las redes neuronales se pueden utilizar para mejorar sistemas de forma incremental, y cada paso puede ser evaluado antes de cometer un desarrollo más amplio.

2.4 Aplicaciones de las redes neuronales

Las redes neuronales son una tecnología computacional emergente que puede utilizarse en un gran número y variedad de aplicaciones, tanto comerciales como militares, de control, entre otras. Se pueden desarrollar redes neuronales en un periodo de tiempo razonablemente corto y pueden realizar tareas concretas mejor que otras tecnologías convencionales, incluyendo los sistemas expertos.

Hay muchos tipos diferentes de redes neuronales, cada una tiene una aplicación particular apropiada. A continuación se presentan ejemplos de algunas aplicaciones comerciales separándolas en grupos según las distintas disciplinas:

Biología

- Aprender más acerca del cerebro y otros sistemas
- Obtención de modelos de la retina

Empresa

- Evaluación de probabilidad de formaciones geológicas y petrolíferas
- Identificación de candidatos para posiciones específicas
- Explotación de bases de datos
- Optimización de plazas y horarios en líneas de vuelo
- Reconocimiento de caracteres escritos

Medio ambiente

- Analizar tendencias y patrones
- Previsión del tiempo

Finanzas

- Previsión de la evolución de los precios
- Valoración del riesgo de los créditos
- Identificación de falsificaciones
- Interpretación de firmas

Manufacturación

- Robots automatizados y sistemas de control (visión artificial y sensores de presión, temperatura, gas, etc.)
- Control de producción en líneas de proceso
- Inspección de la calidad

Medicina

- Analizadores del habla para la ayuda de audición de sordos profundos
- Diagnostico y tratamiento a partir de síntomas y/o de datos analíticos (electrocardiograma, encefalograma, análisis sanguíneo, etc.)
- Monitorización en cirugía
- Predicción de reacciones adversas a los medicamentos
- Lectores de rayos X
- Entendimiento de la causa de los ataques epilépticos

Militares

- Clasificación de las señales de radar
- Creación de armas inteligentes
- Optimización del uso de recursos escasos
- Reconocimiento y seguimiento en el tiro al blanco

La mayoría de aplicaciones son hechos comunes; la mayoría de ellas consisten en realizar un reconocimiento de patrones: buscar un patrón en una serie de ejemplos, clasificar patrones, completar una señal a partir de valores parciales o reconstruir el patrón correcto partiendo de uno distorsionado. Muchos de los ejemplos tienen que trabajar con datos sensoriales y de percepción (señales visuales, auditivas y otras) y otros realizan filtrado o mapeo de señales de entrada, pero no son las únicas actividades que pueden realizar, el control de procesos es una más, entre otras.

Reconocimiento de patrones. El termino reconocimiento de patrones se refiere a la

detección de formas simples, tales como caracteres escritos a mano, mapas del tiempo y espectros del lenguaje. Sin embargo, un objetivo más ambicioso ha sido durante todo el tiempo implementar la percepción artificial; es decir, imitar las funciones de los sistemas sensoriales biológicos en su forma más completa.

2.5 Herramientas de software para desarrollo

La comercialización de productos de software ha sido y sigue siendo la forma más extendida para simular redes neuronales debido a las ventajas citadas anteriormente. La diferencia entre los distintos productos software radica en aspectos tales como el tipo y número de arquitecturas de red que soporta (*backpropagation, Hopfield, Kohonen, etc.*), velocidad de procesamiento (número de conexiones por unidad de tiempo), interfaz gráfica, exportación del código C para el desarrollo automático de aplicaciones, etc. Respecto a las herramientas de software desarrolladas en Europa, hay que citar la obtenida del proyecto Esprit, no comercializada todavía y que funciona en estaciones de trabajo con sistema operativo UNIX y con interfaz gráfica basada en X-Windows.

Actualmente existen un número bastante elevado de productos de software para el desarrollo de aplicaciones con redes neuronales. A la hora de decidirse por la utilización de una u otra, conviene tener muy claro el uso que se va a hacer de la misma. Es preferible elegir entre aquellos productos que se centran en pocas arquitecturas, pues, en muchos casos, cuando ofrecen una gran variedad de las mismas, no todas las incluidas son plenamente funcionales. Otros aspectos importantes a tener en cuenta son, por ejemplo, la forma de leer los datos durante la fase de aprendizaje (si se leen del disco en vez de la memoria, el proceso se hace largo y tedioso), la inclusión de módulos para el análisis de resultados o la posibilidad de generar código C incluyendo los pesos obtenidos.

CAPÍTULO 3

DESCRIPCIÓN DEL DESARROLLO DEL MODELO DE LA PLANTA

3. DESCRIPCIÓN DEL DESARROLLO DEL MODELO DE LA PLANTA

3.1 Diseño y desarrollo de un simulador para una planta virtual, por medio de redes neuronales

Actualmente las redes neuronales se utilizan en múltiples campos de desarrollo, para este trabajo la finalidad de utilizar una red neuronal surge debido a la necesidad de tener una planta piloto virtual que simule la dinámica del sistema en tiempo real, similar a la que se tiene físicamente en los laboratorios.

La creación de la red neuronal se obtiene mediante el uso de comandos y herramientas del software Matlab, con el análisis de la dinámica de la planta de nivel, se propone una prueba (en lazo abierto) y en base a los resultados obtenidos se implementan éstos en la red de Retro propagación que reproducirá la dinámica de la planta.

3.2 Metodología

El procedimiento para la implementación de la red neuronal, permite la obtención del modelo de la planta y que puede ser extrapolable a cualquier proceso.

3.2.1 Análisis de la planta PCT9

En la Fig. 3.1, se observa el diagrama de la planta PCT 9 cuyo objetivo es controlar el nivel del tanque B a través de la regulación del flujo proveniente del tanque A, esta línea, conocida como de proceso, está constituida por una bomba, una válvula mecánica V1 cuya función es limitar el flujo máximo, un indicador de flujo F1 que es del tipo rotámetro, una válvula solenoide normalmente abierta que tiene el objetivo de evitar el retorno por gravedad cuando la bomba esté desactivada, una válvula reguladora V3 que es el elemento final de control en un regulador retroalimentado, por otra parte en el tanque B se encuentra el sensor de nivel, que funciona bajo el principio de cambio de resistencia eléctrica (ver anexo A para una descripción mas detallada).

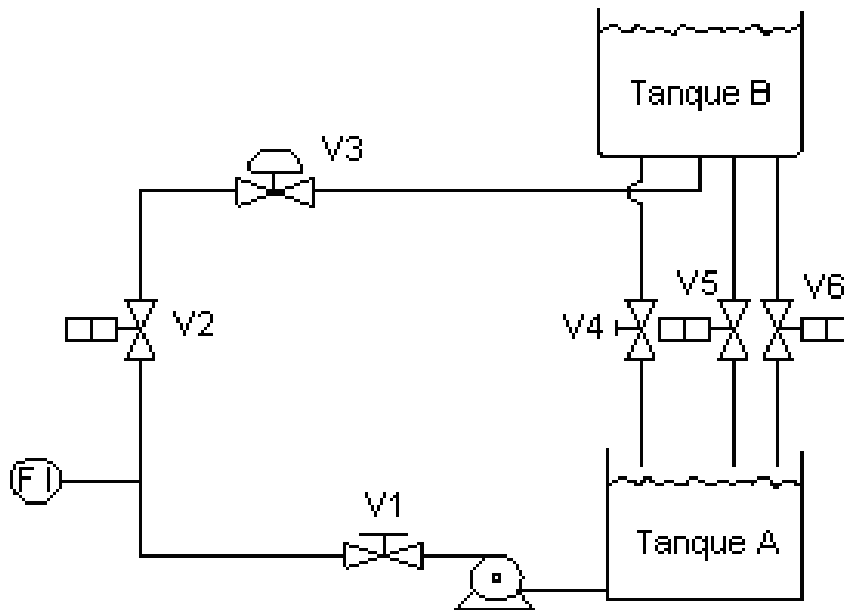


Figura 3.1 Representación esquemática de la planta de nivel.

3.2.2 *Diseño de las pruebas*

El objetivo de las pruebas consiste básicamente en dos etapas, la primera es encontrar un punto de operación, en donde tanto el flujo de entrada como el de salida sean los mismos, la segunda es evaluar el comportamiento del nivel con respecto al flujo de salida.

La relación entre el nivel y los flujos resulta ser no lineal debido a que el flujo de entrada al tanque B es por la parte inferior, esto ocasiona que el flujo de entrada no sea constante, pues éste tiene que vencer la presión hidrostática que va aumentando a medida que el nivel del tanque se va incrementando, por otra parte el mismo fenómeno también influye en el flujo de salida, pues a medida que el nivel es mayor, el flujo de salida también lo es.

Por lo anterior el procedimiento de la primera prueba consiste en suministrar un flujo de entrada constante al tanque B y al mismo tiempo activar la válvula solenoide V6. Para la ejecución de esta prueba se deben considerar dos posibles acciones:

- a) Si el nivel en el tanque incrementa, entonces disminuir el flujo de entrada.
- b) En caso contrario, si el nivel en el tanque decrementa, aumentar el flujo de entrada.

En la tabla 3.3 se muestra, el nivel que se obtiene, cuando el flujo de entrada y de salida son iguales.

Tabla 3.1 Resultados de la primera prueba.

Nivel	90mm
Flujo de salida	2.2 L/min.
Flujo de entrada	2.2 L/min.

El procedimiento para la segunda prueba consiste en llenar el tanque B a un nivel de 180 mm, para posteriormente abrir la válvula solenoide V6 e ir tomando el tiempo que tarda en bajar el nivel cada 30 mm, una vez obtenidos los valores y con el área del tanque B, se procede a realizar la conversión de mm³ a L, y aplicar la ecuación 3.1 para obtener el flujo de salida (Fo).

$$F_o = \frac{V}{t} \quad \text{Ecu. 3.1}$$

Donde:

Fo es el flujo de salida (L/min).

V es el volumen (L)

t es el tiempo (min)

Tomando flujos de salida en intervalos de 30mm, se muestran en la tabla 3.2 los siguientes resultados:

Tabla 3.2 Relación de nivel – flujo.

Nivel (mm)	Flujo de salida (L/min)
0	0
30	0.7
60	1.2
90	2.2
120	2.7
150	3
180	3.3

Con los resultados de la tabla 3.2 se afirma, la no linealidad de la planta de nivel, causada por la presión hidrostática que se genera, a medida que el nivel se incrementa.

3.2.3 *Análisis de resultados*

La figura 3.2 muestra la gráfica que se obtiene de tomar los valores de la tabla 3.2 sirve para mostrar el comportamiento del nivel del tanque B (en mm) con respecto a los flujo de salida, comprobando que el nivel y el flujo de salida son aproximadamente proporcionales, es decir ambos disminuyen con respecto al tiempo de manera proporcional.

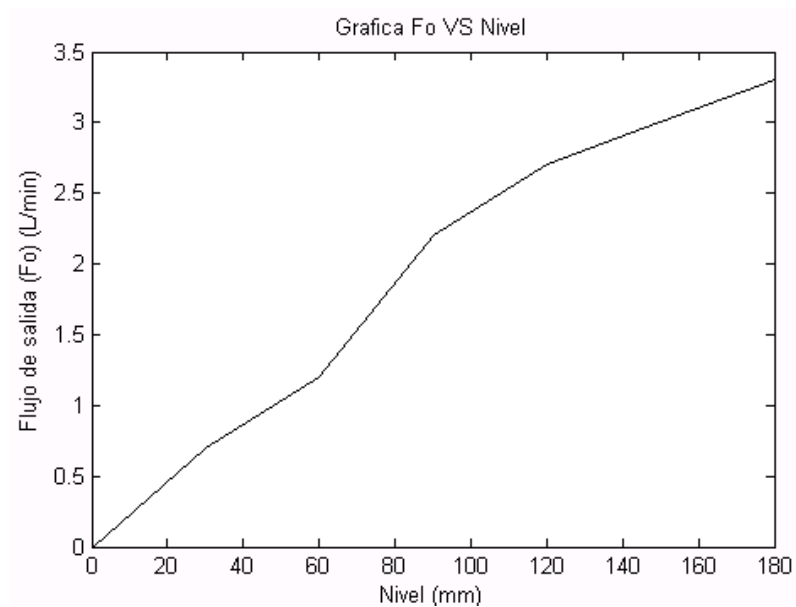


Figura 3.2 Gráfica del flujo de salida vs nivel

La razón por la que la curva de nivel se comporta de esta manera se puede encerrar básicamente en dos aspectos:

a) El flujo de entrada va disminuyendo a lo largo del tiempo, según la metodología de la prueba.

b) La presión hidrostática que se presenta, debido a que el flujo de entrada del rotámetro es por la parte inferior del tanque B.

3.2.4 Implementación de la red neuronal

La red neuronal a implementar es generada desde el M-file de Matlab, en donde se usa el código mostrado en las siguientes tablas (ver anexo D).

Primeramente, para generar la matriz de entrada es necesario definirla, por medio de un vector de 1x7 conformado por los valores 7 valores del flujo de salida de la columna 1 de la tabla 3.2, este vector es formado por el código de la tabla 3.3

Tabla 3.3 Vector de entrada.

ENTRADA	CÓDIGO EN MATLAB
Nivel	p=[0 30 60 90 120 150 180]

Una vez definida la entrada (p), se define el objetivo, que es el vector de salida conformado por los valores que se obtuvieron del flujo de salida para cada nivel (segunda columna de la tabla 3.2). El código se muestra en la tabla 3.4.

Tabla 3.4 Vector salida.

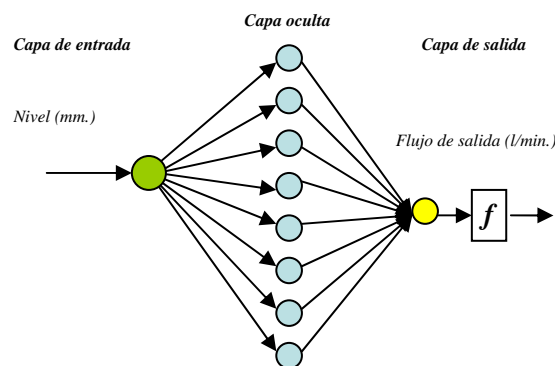
SALIDA	CÓDIGO EN MATLAB
Flujo de salida	t=[0 .7 1.2 2.2 2.7 3 3.3]

Definidas las entradas y la salida, se genera la red ya seleccionada, por medio del código mostrado en la tabla 3.5

Tabla 3.5 Generación de la red.

CÓDIGO EN MATLAB
net = newff([0 180],[10 1],{'tansig' 'purelin'},'trainlm');

Con esta función se crea una red de tres capas. En la primera capa se tiene un vector de entrada de un elemento, en la segunda se tiene una capa oculta con 8 neuronas y en la capa de salida hay una neurona (Fig. 3.3). Los valores que toma la red son los valores máximos y mínimos (180 mm – 0 mm) de la matriz de entrada P . La función de transferencia en la capa oculta es la tan-sigmoidal (tansig) y en la capa de salida la función de transferencia es lineal (purelin). El algoritmo de entrenamiento es Levenberg-Marquardt (trainlm).

**Figura 3.3** Topología de la red neuronal

3.2.5 Entrenamiento de la red

Una vez generada la red, comienza a entregar valores aleatorios a los pesos, para lograr que la red responda a los objetivos planteados, es necesario entrenarla. En la tabla 3.6 se muestra el código de entrenamiento de la red, el cual se genera en 200 épocas, y lo hace por medio de la corrección del error entre las iteraciones de .001. El entrenamiento genera una gráfica, mostrada en la Fig. 3.4, en la cual se observa el decremento del error, que va desde un valor de 10^2 hasta 10^{-3} .

Tabla 3.6 Entrenamiento de la red.

CÓDIGO EN MATLAB
<code>net = train(net,p,t);</code>

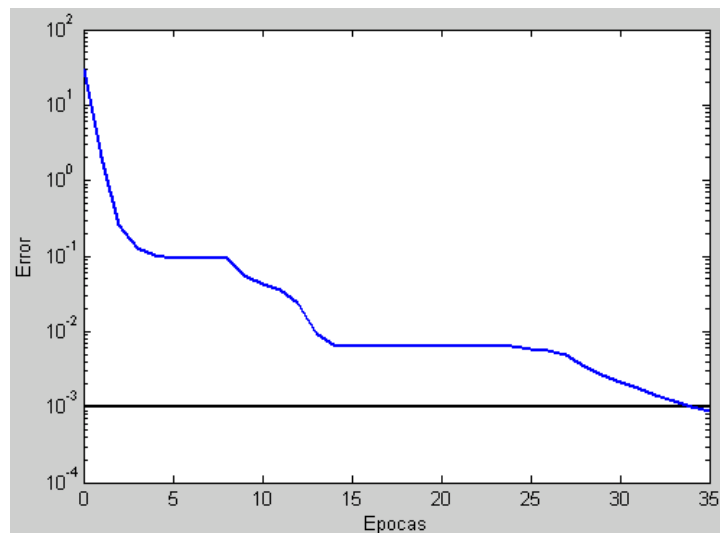


Figura 3.4 Gráfica de entrenamiento de la red.

3.2.6 Validación de resultados

Una vez entrenada la red, para verificar los resultados que se esperan obtener, se crea el siguiente código, mostrado en la tabla 3.7

Tabla 3.7 Simulación de la red.

CÓDIGO EN MATLAB
<pre>y2 = sim(net,p); plot(p,t,p,y2)</pre>

Por medio del código anterior se genera en Matlab la gráfica mostrada en la Fig. 3.5, donde se compara la respuesta de la red con la respuesta experimental.

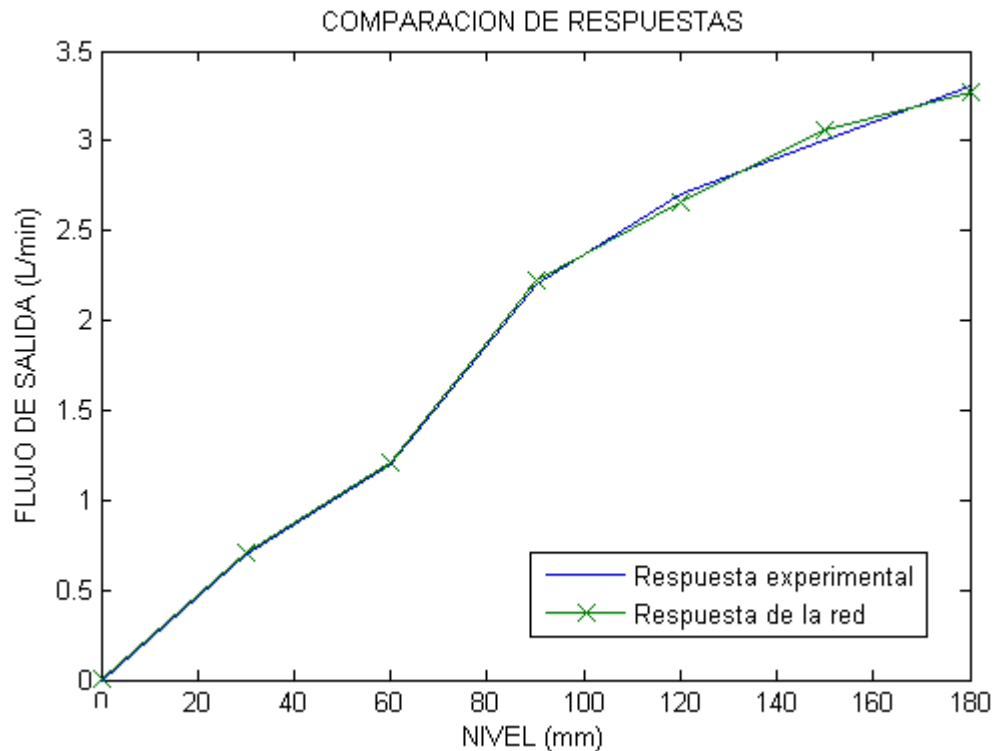


Figura 3.5 Gráfica de comparación de las curvas experimentales y las simuladas.

3.2.7 Generación de la red neuronal en Simulink

Una de las ventajas que ofrece Matlab es generar la red obtenida en Simulink esto es mediante el código mostrado en la tabla 3.8, la red neuronal formada por dicho código es mostrada en la figura 3.6

Tabla 3.8 Generación de la red en Simulink.

CÓDIGO EN MATLAB
<code>gensim(net,1)</code>

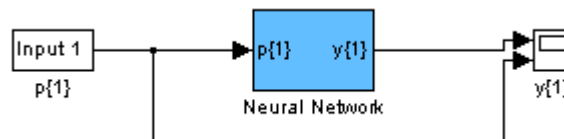


Figura 3.6 Representación grafica de la red neuronal.

3.2.8 Validación de resultados de la red neuronal en Simulink

Para la comprobación del correcto funcionamiento de la red generada con los datos de la tabla 3.2, se propone una entrada (nivel) de 120 mm, por lo que debe entregar la red como salida (flujo de salida) 2.7 L/min (de acuerdo con la tabla 3.2). La respuesta que entrega la red se aprecia en la figura 3.7

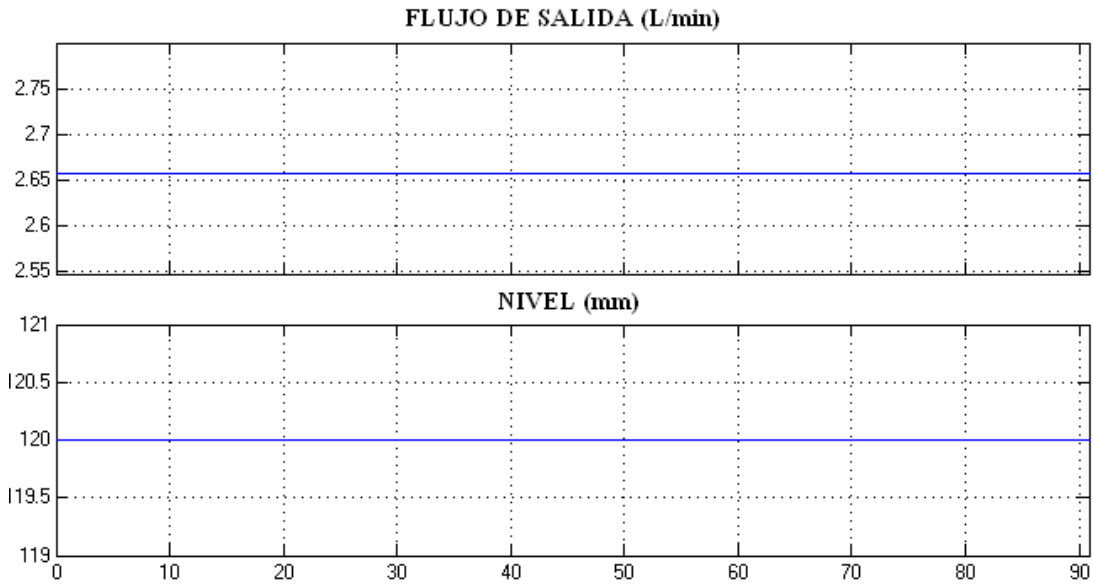


Figura 3.7 Grafica de la entrada con respecto a la salida

Se observa que la red entrega como salida 2.65 L/min, lo cual es muy próximo al valor real esperado.

3.3 Simulación del proceso de nivel con la red neuronal

Comprobada la eficiencia de la red neuronal, se elabora el diagrama en Simulink (Fig. 3.8), el cual simulara el proceso de la planta de nivel, en este diagrama se implementa también la red neuronal generada anteriormente.

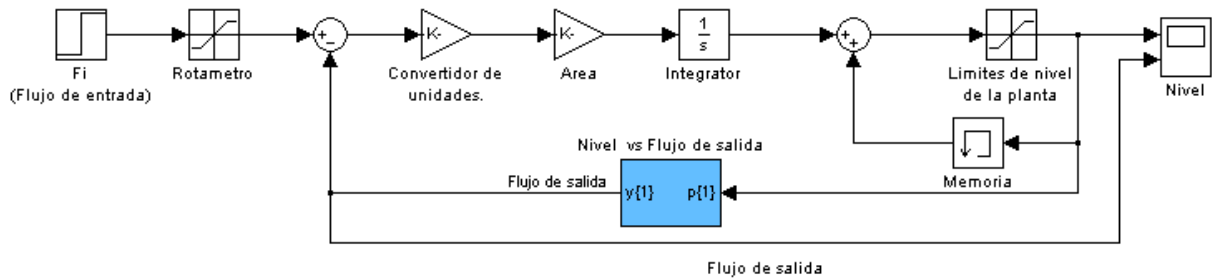


Figura 3.8 Diagrama del proceso de la planta de nivel.

La figura anterior, genera el comportamiento de la planta de nivel, la descripción del diagrama anterior es la siguiente:

- 1) Se tiene un bloque F_i , en el cual se introduce el valor del flujo de entrada (en L/min).
- 2) La salida del bloque F_i entra al bloque Rotámetro que es un saturador, manejando los rangos de 0-3 L/min.
- 3) Posteriormente la salida del Rotámetro entra a un punto suma, en el cual su valor es restado por el valor de salida (flujo de salida) que entrega el bloque de la red neuronal, que está en función del nivel de entrada que esta reciba, obteniéndose una diferencia de flujos.
- 4) El resultado de esa diferencia entra al Bloque Convertidor de unidades, donde se realiza la conversión de unidades, convirtiendo los L/min a mm^3/seg . La finalidad de esta conversión es por que la escala del tanque B esta dada en mm.
- 5) La salida del bloque conversión de unidades entra al bloque Área, el valor de esta área es 50000mm^2 y esta expresada inversamente, es decir $1/50000 \text{mm}^2$ y eliminar las unidades cúbicas del flujo de salida del tanque, teniendo en la salida solo mm/seg.
- 6) Dado que el flujo de salida es una diferencia de flujos (entrada-salida) con respecto al tiempo, es necesario integrar este flujo para eliminar las unidades de

tiempo, es por ello que la salida del bloque Área entra a un bloque integrador, obteniendo como salida el nivel del tanque (mm).

7) La salida del integrador entra a un segundo punto suma, en él se suma el valor de la salida del bloque Memoria y el valor de la salida del integrador, la finalidad de tener el bloque Memoria es establecer como condición inicial que el nivel del tanque esta en un principio en 90 mm. La salida del punto suma entra al bloque Limites de nivel de la planta, quien establece los limites con los que trabaja el tanque B (0-180mm).

8) El nivel obtenido, alimenta la entrada del bloque de la red neuronal, dando como salida, un flujo de salida.

3.4 Verificación de la dinámica de la planta de nivel

Para verificar si la simulación de la planta es correcta, primeramente se propone establecer un flujo de entrada de 2.2 L/min., y un tanque vacío (0 mm.), con el objetivo de verificar que el nivel a obtener es de 90 mm, posteriormente, se proponen los siguientes flujos de entrada: 3, 2.2 y 1.5 L/min., tomando en cuenta que las condiciones iniciales del tanque es de 90 mm. Las gráficas obtenidas se muestran en las siguientes figuras:

Para un flujo de 2.2 L/min., y un nivel de 0 mm:

Observaciones de la figura 3.9: a medida que el nivel aumenta, el flujo de salida también lo hace, de igual forma como sucede en la plata fisica, la plata simulada

también alcanza un estado estacionario con un flujo de entrada de 2.2 L/min., cuyo valor es de 90 mm.

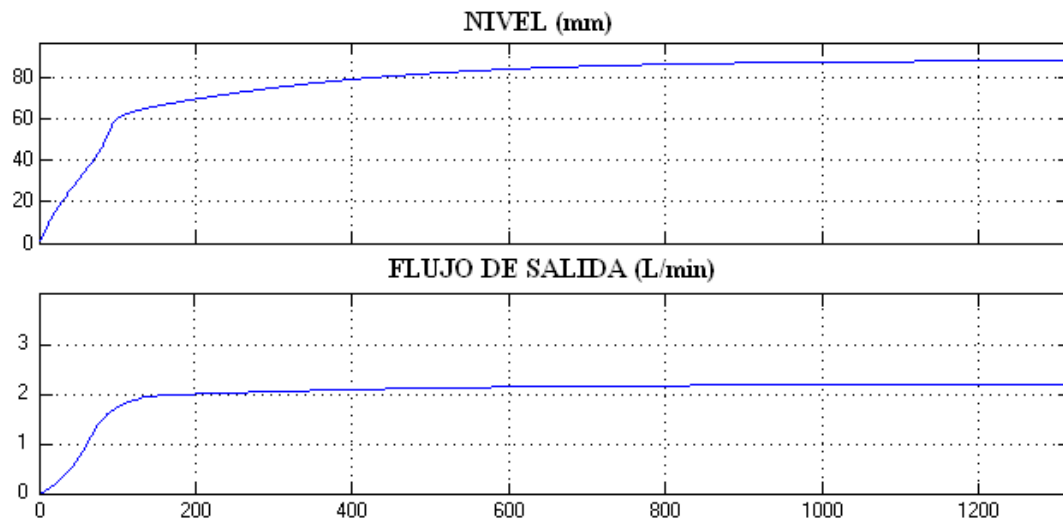


Figura 3.9 Prueba de error en estado estacionario.

Para un flujo de entrada de 3 L/min:

Observaciones: cuando tenemos un flujo de entrada de 3 L/min., y tomando en cuenta que el tanque está en un nivel de 90mm es lógico que el nivel tienda a subir, pues en un principio el flujo de salida (generada por la presión hidrostática) es menor al flujo de entrada, pero a medida que el nivel va aumentando el flujo de salida también lo hace.

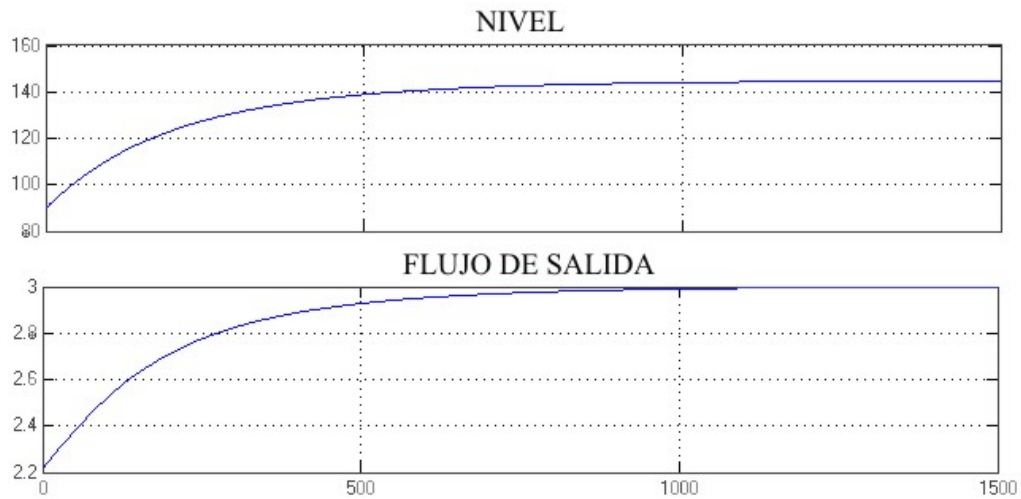


Figura 3.10 Gráfica de la planta de nivel, para un flujo de entrada de 3 L/min.

Para un flujo de entrada de 2.2 L/min.

Observaciones: en este caso los flujos de entrada y salida son del mismo valor (2.2 L/min.), debido a que cuando el nivel es de 90 mm el flujo de salida es de 2.2; por lo que la gráfica debería tender a hacer una recta, pero la deformación que se aprecia es correcta, dado a que físicamente no se puede mantener un flujo de entrada constante (al menos no en lazo abierto), es por ello que se aprecian pequeñas variaciones tanto del nivel como de flujo de salida.

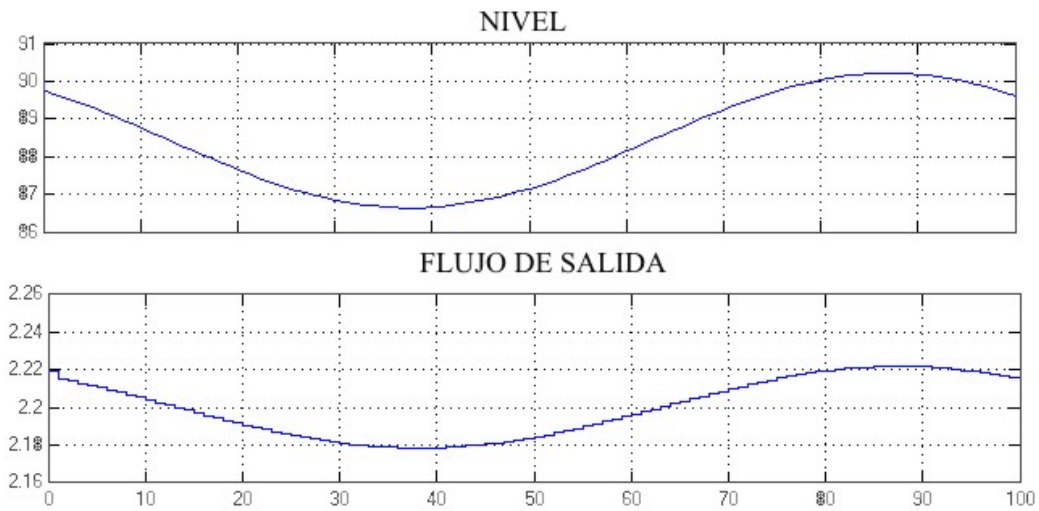


Figura 3.11 Gráfica de la planta de nivel, para un flujo de entrada de 2.2 L/min.

Para un flujo de entrada de 1.5 L/min:



Figura 3.12 Gráfica de la planta de nivel, para un flujo de entrada de 1.5 L/min.

Observaciones: cuando se tiene un flujo entrada de 1.5 L/min., y tomando en cuenta que el tanque está a un nivel de 90mm es lógico que el nivel tienda a bajar, pues el flujo de salida que se genera por la presión hidrostática es mayor.

Otras pruebas que se proponen para asegurarse que la planta funciona adecuadamente, es colocar un flujo de entrada que varíe, esto se justifica dado que en la realidad la presión hidrostática no permite tener un flujo de entrada constante, los flujo de entrada propuestos son los mismo que en las pruebas anteriores, de tal forma que las gráficas quedan de la siguiente manera:

Para un flujo de entrada de 3 L/min.:

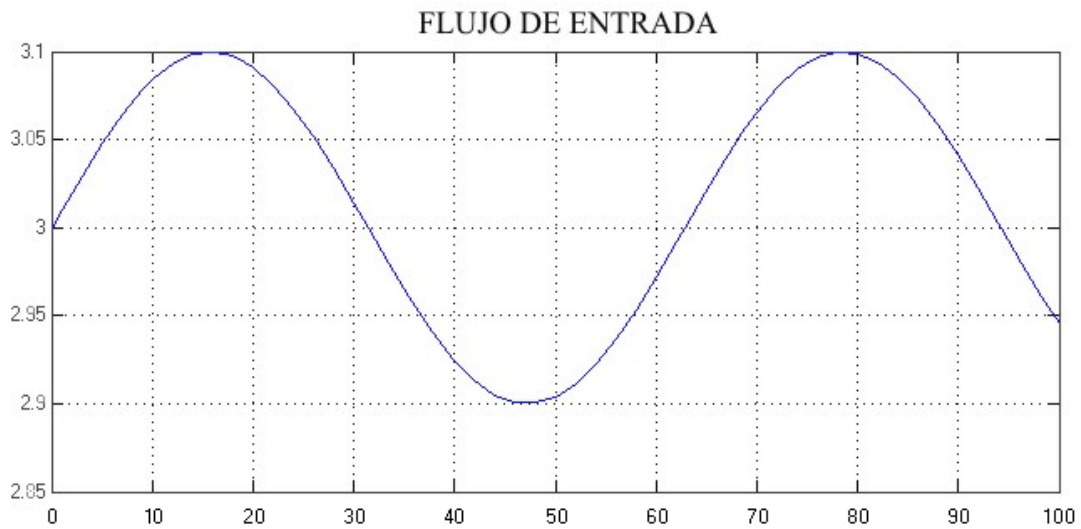


Figura 3.13 Gráfica del flujo de entrada de 3 L/min.

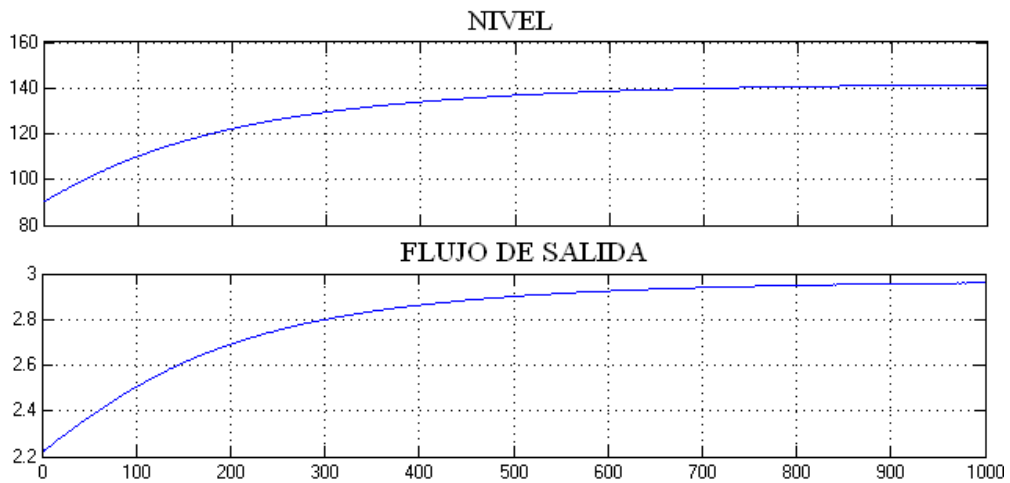


Figura 3.14 Gráfica de la señal de la planta simulada.

Para un flujo de entra de 2.2 L/min.:

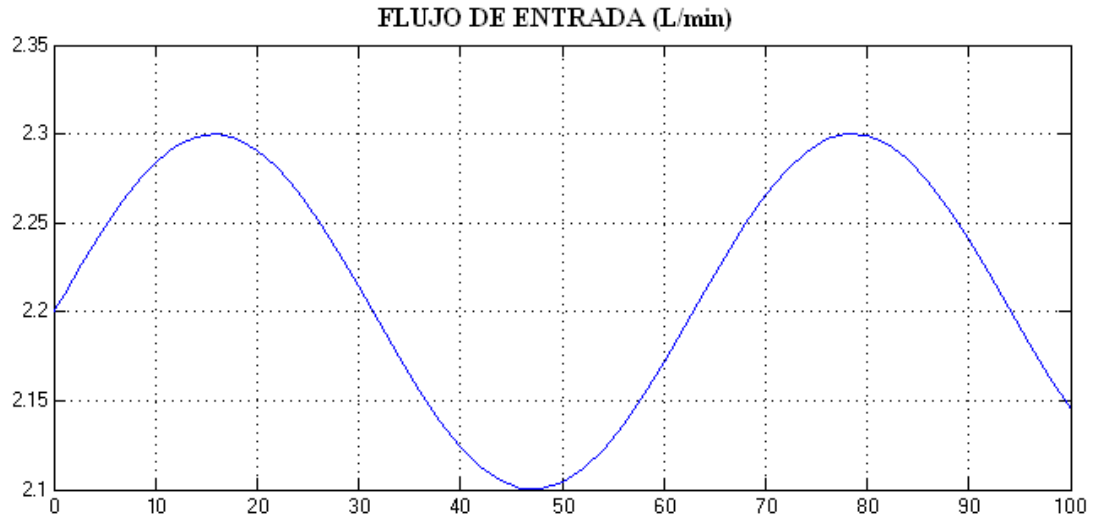


Figura 3.15 Gráfica del flujo de entrada de 2.2 L/min.

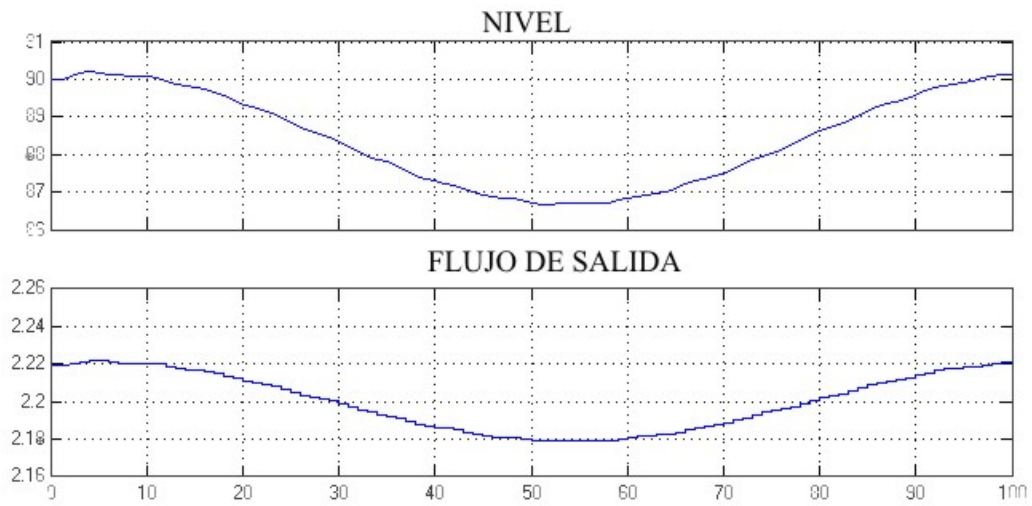


Figura 3.16 Gráfica de la señal de la planta simulada.

Para un flujo de entrada de 1.5 L/min. :

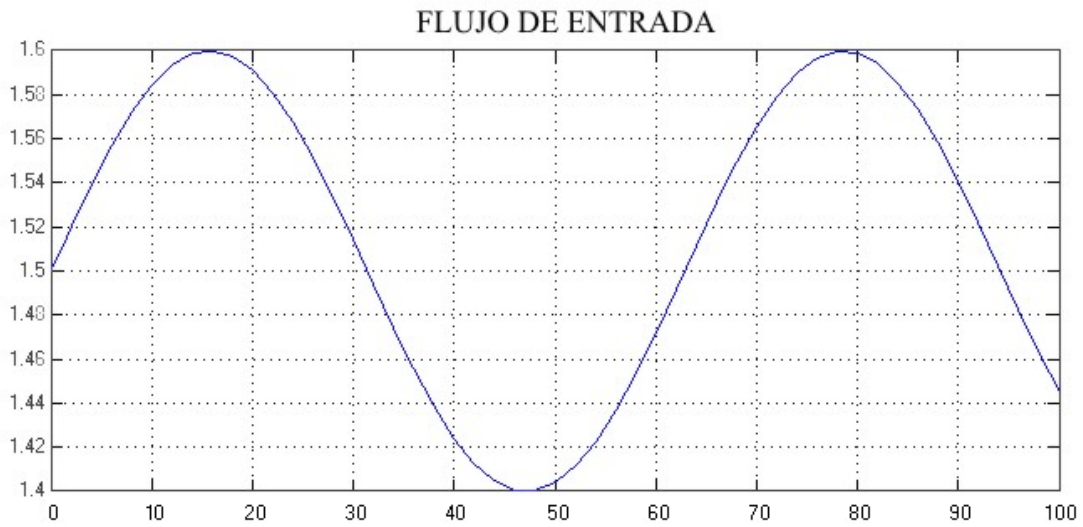


Figura 3.17 Gráfica del flujo de entrada de 1.5 L/min.

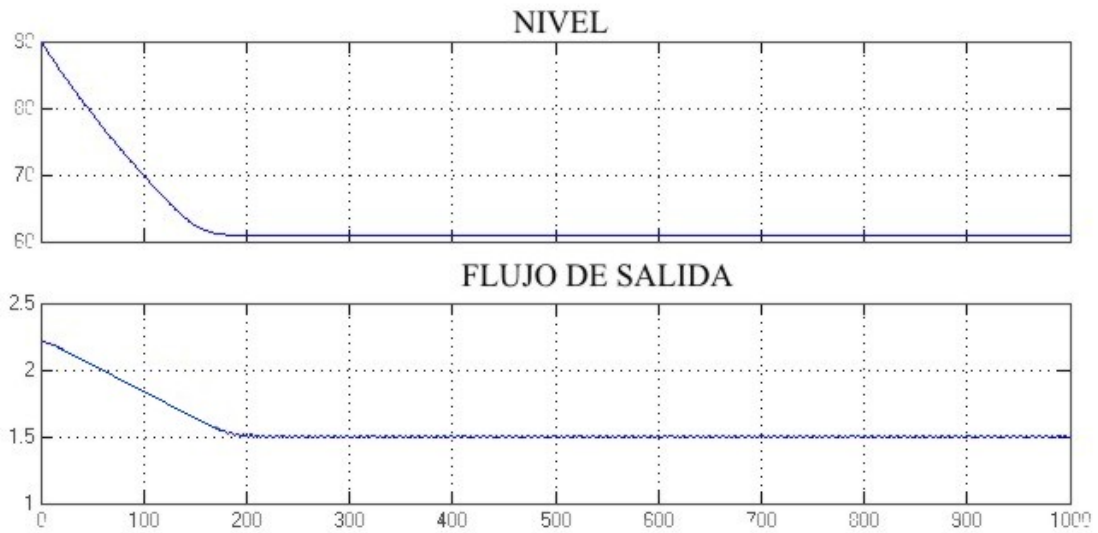


Figura 3.18. Gráfica de la señal de la planta simulada.

CAPÍTULO 4

ANÁLISIS DE COSTOS

4. ANÁLISIS DE COSTOS

Puesto que el presente proyecto es de investigación, los dos investigadores, han sido apoyados por la Comisión de Operaciones y Fomento de Actividades Académicas (COFAA), donde los ingresos por investigador son de \$1158. Por su parte la Escuela Superior de Ingeniería Mecánica y Eléctrica (ESIME) de la unidad Zacatenco, apoya con el préstamo de las instalaciones (laboratorio de pesados I), equipo (maquina de computo y planta de nivel) y material de software (Matlab) necesarios, para la elaboración del proyecto.

El desarrollo del proyecto se lleva a cabo en un periodo de 9 meses (Febrero-Junio y Agosto-Noviembre), en la tabla 4.1 se muestra el costo mensual, y la tabla 4.2 muestra el costo total.

Tabla 4.1 Costo mensual

PERSONAL	INGRESO
Investigador 1	\$1158
Investigador 2	\$1158
Total	\$2,316

Tabla 4.2 Costo total

PERSONAL	INGRESO
Investigador 1	\$10,422
Investigador 2	\$10,422
Total	\$ 20,844

4.1 Costos de equipo y materiales

Sin embargo, si se visualizara este proyecto como un negocio, se tendría que pensar en los costos del equipo, haciendo la aclaración, de que el proceso a simular sería brindado por parte de la escuela o empresa. Por lo cual los costos de equipo y materiales a considerar son:

1) **Matlab** que es un programa de cálculo numérico, orientado a matrices y vectores creado por *The MathWorks*, todo lo que se pretenda hacer con él, será mucho más rápido y efectivo, hablando en términos de matrices y vectores. Está disponible para las plataformas Unix, Windows y Mac OS X. Se pueden ampliar sus capacidades con *Toolboxes*, algunas de ellas están destinadas al procesamiento digital de señales, adquisición de datos, economía, inteligencia artificial, lógica difusa, entre otras. También cuenta con otras herramientas como Simulink, que sirve para simular sistemas. La tabla 4.3 presenta el precio los dos principales proveedores de Matlab.

Tabla 4.3 Comparativa de precios de Matlab.

Proveedor	Producto	Precio
Mathworks	MATLAB and Simulink (Student Version).	\$ 1,118
Mathworks	MATLAB and Simulink (Comercial Version).	\$ 2,825
Central del Software S.A. de C.V.	Matlab (Académico)	\$ 1,695
Central del Software S.A. de C.V.	Matlab (Comercial)	\$ 2,985

Nota: estos precios solo incluyen Matlab y Simulink, si se desea obtener los toolboxes, se deberá sumar el precio de \$610 por cada *toolbox* que se desee. También se debe considerar que todos los precios mostrados no incluyen gastos de envío.

2) Equipo de cómputo la maquina de cómputo que se propone es una DELL P4 DE 3.0 GHZ (Fig. 4.1), cuyo precio es de \$ 5,500. La computadora tiene las siguientes características:

- Procesador Genuino Intel Pentium 4 3.06 GHZ
- Disco duro de 80GB
- 1GB de memoria Ram.
- Lector de CD-Room
- 6 Puertos USB
- 3 puertos PCI
- Drive de 3.5" 1.44 MB
- Sonido y Red Integrados
- Tarjeta de Red Inalámbrica LinkSys de 2.4 GHZ
- Monitor Dell LCD Original de 15.0 Inchs
- Teclado y mouse Dell
-



Figura 4.1 Computadora DELL P4.

Nota: El precio no incluye gastos de envío ni IVA.

3) Caso de estudio este es proporcionado por la empresa o escuela que solicite el servicio. Planta piloto PCT9 Process Module, de la compañía Armfield (Fig. 4.2), tiene un costo estimado de \$115,000, mientras que para el módulo de control (Fig. 4.3) el costo es de \$185,000; haciendo un total de \$400,000.

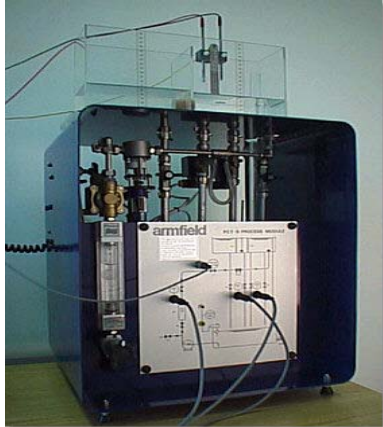


Figura 4.2 Planta piloto PCT9.



Figura 4.3 Controlador.

4.2 Costos de ingeniería

En estos costos se toma en cuenta la intervención de dos ingenieros, haciendo un estudio económico acerca del salario mínimo de un ingeniero, se encuentra que es de \$4,500, mientras que el tiempo que se estima para el desarrollo del proceso virtual es de 5 meses, teniendo un total de costo de mano de obra de \$45,000.

4.3 Otros costos

Este costo está referido al área de trabajo, cuya ubicación es en Av. Central Carlos Hank González Mz. 6 Lt. 8, Col. La Olímpica Ecatepec Edo. De México, cuyas

dimensiones son de 6 X 10 metros. El costo de la renta del área (local) de trabajo es de \$2,000 mensuales. En lo que se refiere a la papelería requerida como es el fotocopiado o impresiones de manuales se estima un costo total de de \$200.

4.4 Costo total del proyecto

La tabla 4.4 muestra el costo total del proyecto, teniendo presente que el software (Matlab) elegido es el de Mathworks, la versión comercial.

Tabla 4.4 Costo total del proyecto.

Concepto	Costo de investigación
Equipo y materiales	\$ 8,325
Mano de obra	\$ 45,000
Otros costos	\$ 10,200
Total	\$ 63,525

4.5. Comparación de costos

Esta parte hace la observación de la ventaja que se obtiene de hacer este tipo de procesos virtuales, ya que tener más de una planta piloto físicamente resulta difícil por su alto costo, además de eso, se debe sumar el costo del mantenimiento que necesita constante, mientras que una planta simulada, no requiere de mantenimiento.

Tabla 4.5 Comparación de precios entre las plantas de nivel.

Concepto	Precio
Planta física.	\$ 400,000
Planta simulada.	\$ 63,625

CAPÍTULO 5

CONCLUSIONES

5. CONCLUSIONES

Las redes neuronales son una tecnología computacional emergente que puede ser utilizada en un gran número y variedad de aplicaciones, en la ingeniería ofrecen la ventaja de poder evitar la elaboración de los modelos matemáticos de procesos. La red de Retropropagación, es una de las más utilizadas debido a las características que posee, como son: capacidad de autoadaptar los pesos de las neuronas y la generalización, que es la capacidad de dar salidas satisfactorias a entradas que el sistema no ha visto nunca en su fase de entrenamiento. Una forma de construir una red neuronal es mediante un software, pues éste procedimiento es rápido, poco costoso e insustituible al momento de realizar el entrenamiento y evaluación de las redes.

Las respuestas de la red neuronal elaborada para el proceso de la planta de nivel simulada, son satisfactorias dado que las respuestas de la planta simulada son cercanas en comparación con las respuestas obtenidas experimentalmente con la planta física, otra punto favorable de la planta simulada, es que para entradas no constantes (flujo de entrada), las respuestas son semejantes a las obtenidas con una entrada constante. Este último punto es bueno ya que en la realidad nunca se logra mantener un flujo de entrada constante, debido a la presión hidrostática que se genera a medida que el nivel del tanque aumenta. Lo anterior crea una atmósfera de confianza en el proceso de nivel, dado que cuando se coloque un controlador elaborado con otra red neuronal y que proporcione la opción de elegir las múltiples arquitecturas de control existentes, así como también los tipos de control (PI, PD, PID

y Control Difuso), permitan comparar que arquitectura y controlador es adecuado para el proceso que se tenga.

Este proyecto es factible debido a los beneficios que proporciona, uno de ellos y quizás el más importante es la cuestión económica, pues para tener una sola planta de nivel y su controlador (físicos) se requiere de una inversión de \$400,000, a esto también se le debe sumar que en un futuro, la planta necesitara de algún tipo de mantenimiento para su optimo funcionamiento, mientras que una plata simulada no necesita de un mantenimiento, y en lo económico el costo que se requiere para elaborarla es 6 veces menor que el de una planta física, lo que significaría tener mas de una planta de nivel, y con esto reflejar una mejor interacción entre estudiantes (para el caso de una escuela) o una mejor capacitación de los trabajadores (en el caso de una empresa) en los procesos industriales.

Finalmente esta tesis forma parte de una fase del proyecto de investigación número 20071510, cuyo objetivo final es visualizar el proceso de nivel animado, así como implementar algoritmos y arquitecturas de control a dicho sistema.

BIBLIOGRAFÍA

1. **HILERA**, José R, **MARTÍNEZ**, Víctor, *Redes Neuronales Artificiales*, 2^{da} edición, Editorial Alfaomega, 2000, Madrid, España.
2. **MARTÍN**, Bonifacio, **SANZ**, Alfredo, *Redes Neuronales y Sistemas Borrosos*, 2^{da} edición, Editorial RA-MA, 2002, Universidad de Zaragoza.
3. **HAYKIN**, Simon, *Neural Networks; A Comprehensive Foundation*, 2^{da} edición, Editorial Prentice Hall.
4. **DEMUTH**, Howard, **BEALE**, Mark, *Neural Network Toolbox For Use With Matlab*, 4^{ta} versión, Editorial The Math Works.
5. **SMITH**, Carlos, **CORRIPIO**, Armando, *Control Automático de Procesos Teoría y Práctica*, 1^a edición 4^{ta} reimpresión, Editorial Limusa.

ANEXO A

DESCRIPCIÓN DE LA PLANTA PCT9

A) DESCRIPCIÓN DE LA PLANTA PCT9

Para una mejor descripción del proceso de la planta PCR9 en lazo abierto es dividido en dos circuitos:

El circuito de entrada del fluido de proceso se aprecia en la figura A-1. Para este caso el fluido de servicio es agua a temperatura ambiente, que se encuentra en un tanque de almacenamiento (1), la entrada del fluido al proceso es por medio de una bomba (2), de manera que a través de una tubería flexible (3) el fluido pueda entrar con una buena presión al rotámetro (F1), el cual regulara el flujo de entrada del fluido entre los rangos de 0 a 3 L/min. De la salida del rotámetro en fluido de servicio entra por una válvula solenoide (Sol 1) normalmente abierta, cuyo objetivo es evitar el retorno por gravedad del fluido de proceso, cuando la bomba esté desactivada; posteriormente a la salida del la válvula Sol 1 el fluido de proceso entra a una válvula reguladora (4) del tipo servoposicionador, que es el elemento final de control en un regulador retroalimentado, de la salida de la válvula 4 el fluido de proceso continua su viaje a través de un by-pass (5) para finalmente entrar por la parte inferior del tanque superior (6), el cual esta seccionado en dos.

El circuito de salida del flujo de proceso se aprecia en la figura A-2. El retorno del fluido de proceso contenido dentro del tanque superior (6) al tanque de almacenamiento (1) puede darse de varias formas, la elección de cada una de ellas

dependerá del tipo de actividad que se este realizando con la planta. Una forma es a través de la apertura de las válvulas manuales (V4 y V5), otra es manera es con la apretura individual de las válvulas solenoides

(Sol 2 y Sol 3) cuyo Cv es de .4 y .6 respectivamente; finalmente la ultima forma de retorno del fluido al tanque de almacenamiento es a través de un tubo vertical (7) que se encuentra dentro del tanque superior cuya altura es 180 mm con respecto a la escala (8) de nivel (en mm) colocada en la parte frontal del tanque superior, la finalidad de este tuvo, es establecer el nivel máximo al cual el fluido podrá llegar; es decir cuando el nivel del fluido rebase los 180 mm comenzara a salir del tanque superior mediante el tuvo vertical. Para la limpieza del tanque de almacenamiento, el fluido es expulsado a través de la válvula manual (V6).

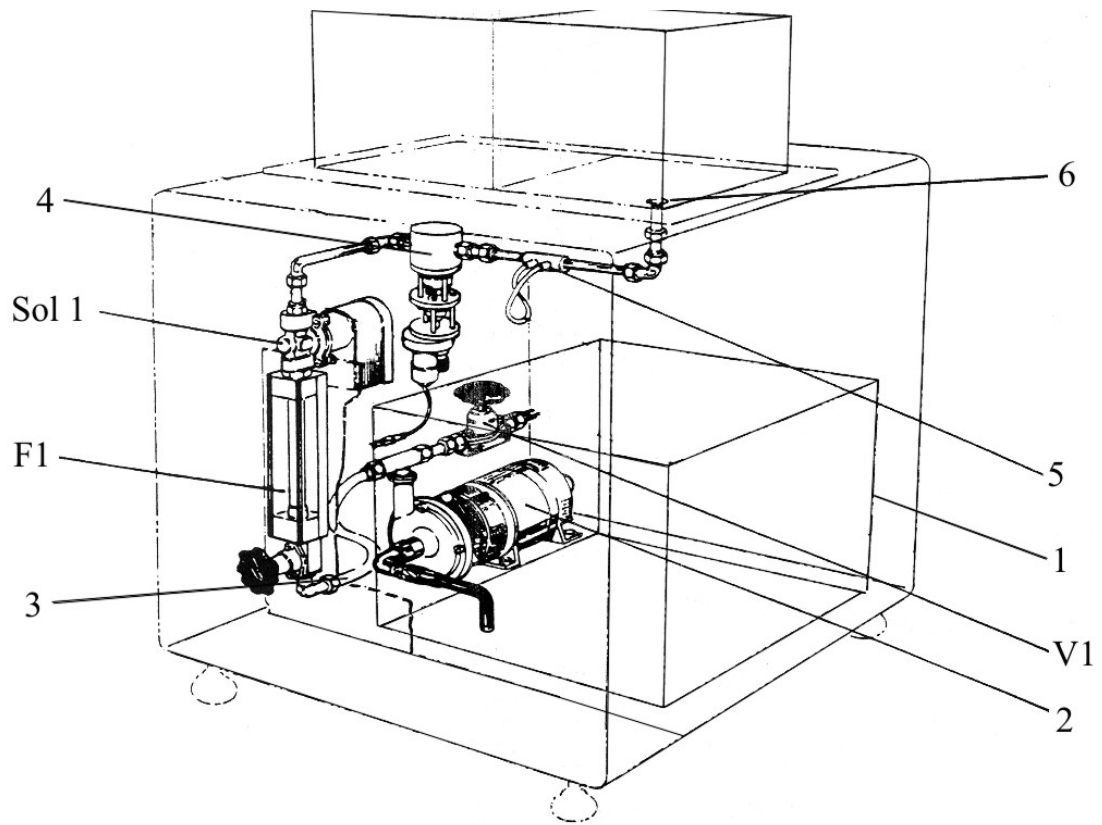


Figura A-1 Circuito de entrada del fluido del proceso.

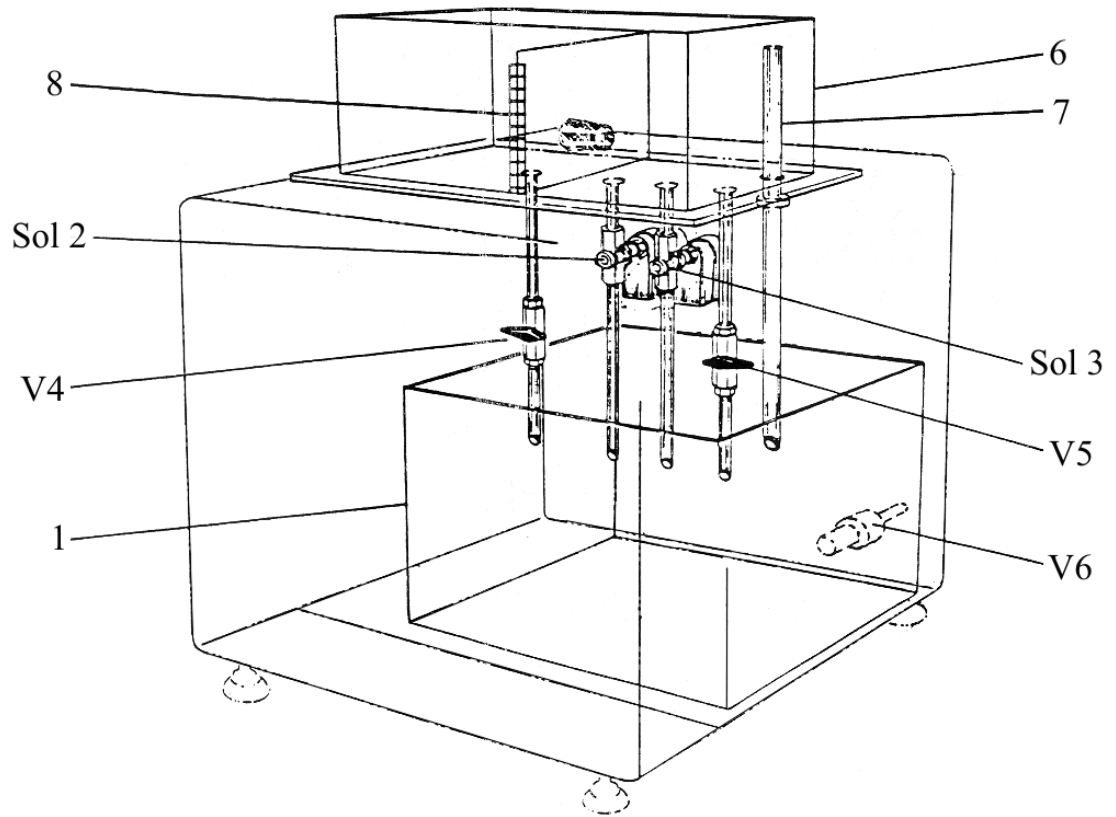


Figura A-2 Circuito de salida del flujo de proceso.

ANEXO B

ROTÁMETROS

B) ROTÁMETROS

El flujo, es una de las muchas variables mas son medidas en la industria; en consecuencia, se han desarrollado diversos dispositivos, entre los cuales se encuentran en la siguiente clasificación.

Tabla B-1 Clasificación de los sensores de flujo.

Medidores de presión diferencial. <ul style="list-style-type: none">❖ Placa orificio❖ Tubo Venturi❖ Tubo Pitot❖ Medidores de impacto
Medidores máscicos. <ul style="list-style-type: none">❖ Medidor máscico térmico❖ Medidor de Coriolis
Medidores volumétricos <ul style="list-style-type: none">❖ Medidor de desplazamiento positivo
Medidores de velocidad <ul style="list-style-type: none">❖ Medidor de turbina❖ Medidor electromagnético❖ Medidor Vortex❖ Rotámetro❖ Medidor de ultra sonidos

El rotámetro es un medidor de sección transversal variable, que consiste en un conducto transparente y un flotador (que en la realidad es más pesado que el líquido), el cual se desplaza hacia arriba debido al flujo también hacia arriba a través del conducto, el tubo gradúa para leer el gasto directamente, en la figura B-1 se muestran las partes principales de las que consta el rotámetro.

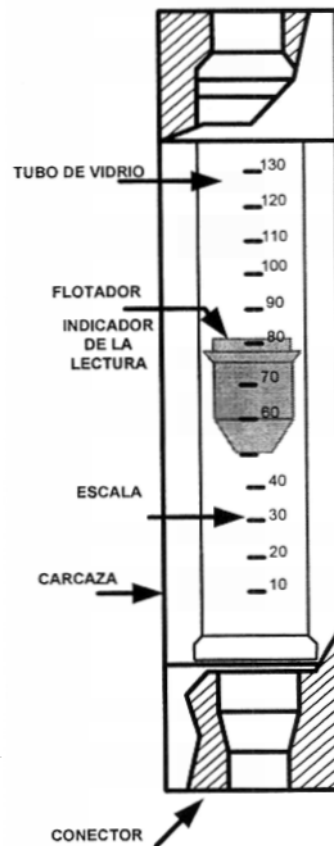


Figura B-1 Rotámetro.

El rotámetro que está instalado en la planta PCT9, es de la marca Platón, el cual tiene las siguientes características:

- Tubo: Cristal de borosilicato.
- Flotador: Acero inoxidable
- Escala: 0 – 3 L/min
- Exactitud: ± 1.25 % de la escala
- Conexiones: $\frac{1}{4}$ " hembra
- Cubierta: Policarbonato

En la figura B-2 se muestran las dimensiones del rotámetro.

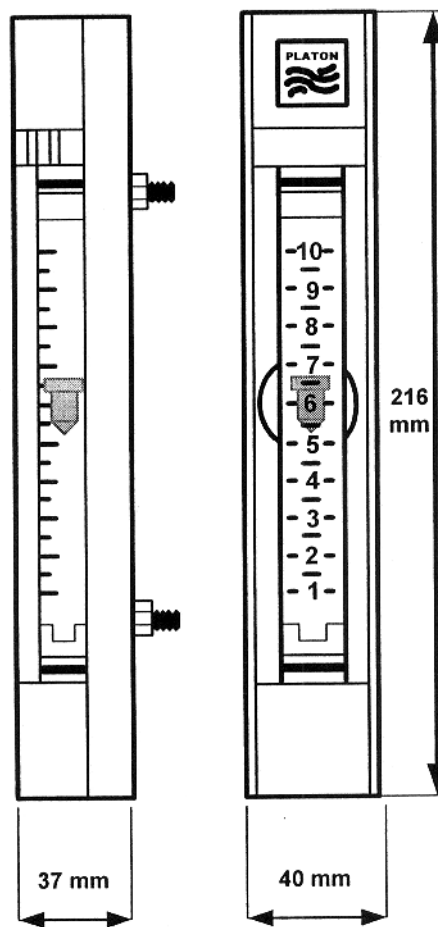


Figura B-2 Rotámetro Platón.

ANEXO C

SERVOPOSICIONADOR

C) SERVOPOSICIONADOR

Los servos son un tipo especial de motor que se caracterizan por su capacidad para posicionarse de forma inmediata en cualquier posición dentro de su rango de operación. Para ello, el servo espera un tren de pulsos que corresponden con el movimiento a realizar. Están generalmente formados por un amplificador, un motor, la reducción de engranaje y la realimentación

El control de un servo se limita a indicar en que posición se debe situar. Estas "ordenes" consisten en una serie de pulsos. La duración del pulso indica el Angulo de giro del motor. Cada servo tiene sus márgenes de operación, que corresponden con el ancho del pulso máximo y mínimo que el servo entiende. Los valores más generales corresponden entre los valores de 1 ms y 2 ms, que dejarían al motor en ambos extremos. El valor 1,5 ms indicaría la posición central, mientras que otros valores del pulso lo dejan en posiciones intermedias. Estos valores suelen ser los recomendados, sin embargo, es posible emplear pulsos menores de 1 ms o mayores de 2 ms, pudiéndose conseguir ángulos mayores de 180°. Si se sobrepasan los límites de movimiento del servo, este comenzara a emitir un zumbido, indicando que se debe cambiar la longitud del pulso.

El periodo entre pulso y pulso no es crítico, e incluso puede ser distinto entre uno y otro pulso. Se suelen emplear valores entre 10 ms y 30 ms. Si el intervalo entre pulso

y pulso es inferior al mínimo, puede interferir con la temporización interna del servo, causando un zumbido, y la vibración del brazo de salida. Si es mayor que el máximo, entonces el servo pasara a estado dormido, entre pulsos. Esto provoca que se mueva con intervalos pequeños.

Es importante destacar que para que un servo se mantenga en la misma posición durante un cierto tiempo, es necesario enviarle continuamente el pulso correspondiente. De este modo, si existe alguna fuerza que le obligue a abandonar esta posición, intentara resistirse. Si se deja de enviar pulsos (o el intervalo entre pulsos es mayor del máximo) entonces el servo perderá fuerza y dejara de intentar mantener su posición, de modo que cualquier fuerza externa podría desplazarlo.[5]

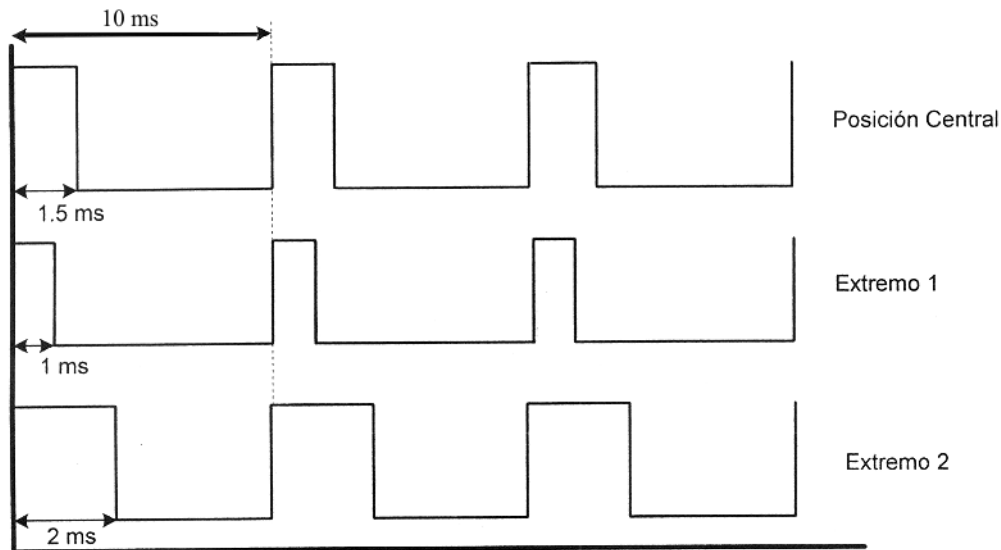


Figura C-1 Pulsos de control para un servoposicionador.

ANEXO D

PROGRAMACIÓN EN MATLAB

D) PROGRAMACION EN MATLAB

```
% Archivo .m para la generación y entrenamiento de la Red neuronal
% que simulara el comportamiento de la planta PCT9.

% Datos obtenidos de la segunda prueba experimental de la planta PCT
x=[0 30 60 90 120 150 180]; % Valores del nivel

y=[0 .7 1.2 2.2 2.7 3 3.3]; % Valores del flujo de salida.

plot(x,y) % Comando de Matlab que genera la figura
          % 3.2 con los valores de x & y

% Instrucciones para la generación y entrenamiento de la Red Neuronal

p=[0 30 60 90 120 150 180]; % Vector entrada, el cual esta constituido
          % por los valores del nivel.

t=[0 .7 1.2 2.2 2.7 3 3.3]; % Vector de salida, el cual contiene los
          % valores del flujo de salida.

% El comando newff genera e indica las capas y el tipo de entrenamiento
% con el cual se forma la Red Neuronal. Donde:

% Los valores que toma la Red Neuronal son los valores mínimos y máximos
% (180 – 0 mm) de la matriz p, mientras que la función de transferencia
% en la capa oculta es la tan-sigmoidal (tansig) y en la capa de salida
% la función de transferencia es lineal (purelin), por ultimo el algoritmo
% de entrenamiento es Levenberg-Marquardt (trainlm).

net = newff([0 180],[10 1],{'tansig' 'purelin'},'trainlm');

y1 = sim(net,p); % Este comando guarda todos los datos generados, para
          % el uso futuro de ellos en Simulink.
```

```
net.trainParam.epochs = 200; % Indica el numero de capas con la que se
                             % genera la red.

net.trainParam.goal = 0.001; % Indica el valor de corrección de error
                             % que tendra la red.

net = train(net,p,t); % El comando train, entrena a la Red Neuronal
                    % con los datos proporcionados anteriormente.

y2 = sim(net,p); % Guarda los nuevos valores de la Red Neuronal después
                % del entrenamiento en Simulink.

plot(p,t,p,y2) % Genera la grafica de la Figura 3.5

gensim(net,1) % Comando que genera la Red Neuronal en Simulink.
```


ANEXO E

CONSTRUCCION DEL MODELO DE LA PLANTA EN SIMULINK.

E) CONSTRUCCIÓN DEL MODELO DE LA PLANTA EN SIMULINK

En las siguientes figuras se muestran los cuadros de diálogos de los bloques que forman el modelo de la plata virtual, generada en Simulink, así como también los respectivos valores ingresados en los mismos.

Step

Output a step.

Parameters

Step time:
.1

Initial value:
0

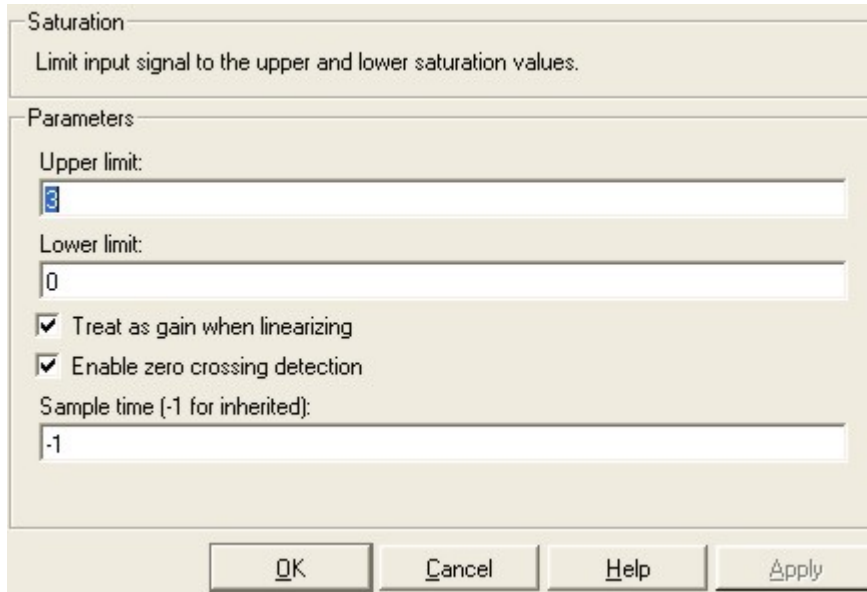
Final value:
2.2

Sample time:
0

Interpret vector parameters as 1-D

Enable zero crossing detection

Figura D-1 Cuadro de dialogo del bloque Fi.



Saturation
Limit input signal to the upper and lower saturation values.

Parameters

Upper limit:
3

Lower limit:
0

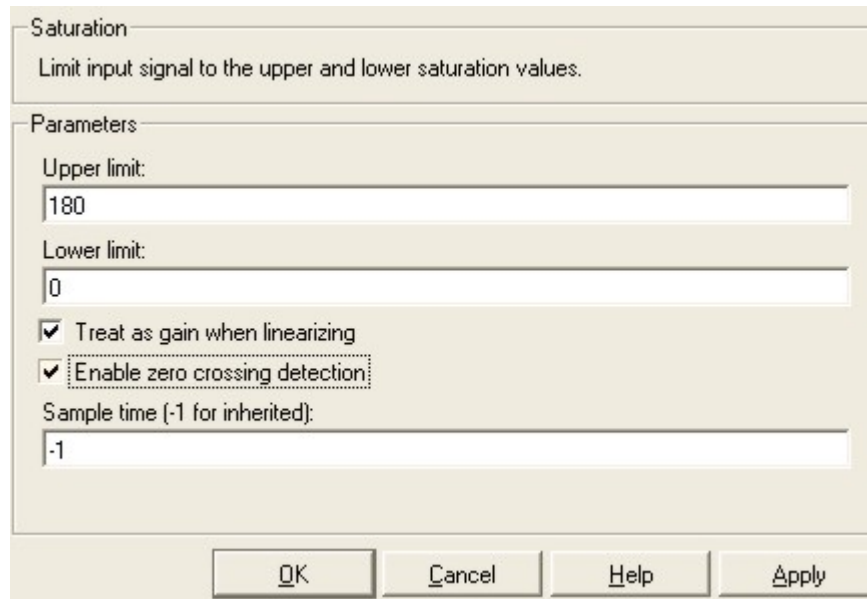
Treat as gain when linearizing

Enable zero crossing detection

Sample time (-1 for inherited):
-1

OK Cancel Help Apply

Figura D-2 Cuadro de dialogo del bloque Rotámetro.



Saturation
Limit input signal to the upper and lower saturation values.

Parameters

Upper limit:
180

Lower limit:
0

Treat as gain when linearizing

Enable zero crossing detection

Sample time (-1 for inherited):
-1

OK Cancel Help Apply

Figura D-3 Cuadro de dialogo del bloque Conversión.

Gain

Element-wise gain ($y = K.*u$) or matrix gain ($y = K*u$ or $y = u*K$).

Main | Signal data types | Parameter data types

Gain:

1/50000

Multiplication: Element-wise($K.*u$)

Sample time (-1 for inherited):

-1

OK Cancel Help Apply

Figura D-4 Cuadro de dialogo del bloque de Área.

Integrator

Continuous-time integration of the input signal.

Parameters

External reset: none

Initial condition source: internal

Initial condition:

0

Limit output

Upper saturation limit:

inf

Lower saturation limit:

-inf

Show saturation port

Show state port

Absolute tolerance:

auto

Enable zero crossing detection

OK Cancel Help Apply

Figura D-5 Cuadro de dialogo del bloque Integrador.

Saturation

Limit input signal to the upper and lower saturation values.

Parameters

Upper limit:
180

Lower limit:
0

Treat as gain when linearizing

Enable zero crossing detection

Sample time (-1 for inherited):
-1

OK Cancel Help Apply

Figura D-6 Cuadro de dialogo del bloque de Límites de nivel de la planta.

Memory

Apply a one integration step delay. The output is the previous input value.

Main | State properties

Initial condition:
90

Inherit sample time

Direct feedthrough of input during linearization

OK Cancel Help Apply

Figura D-7 Cuadro de dialogo del bloque Memoria.