
# UNIVERSIDAD TÉCNICA DE AMBATO

## FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL

## POSGRADOS

### PROGRAMA DE MAESTRÍA EN MATEMÁTICA APLICADA

### MODALIDAD DE TITULACIÓN PROYECTO DE DESARROLLO

Trabajo de titulación previo a la obtención del grado académico de Magíster en Matemática Aplicada

**Tema:** "Implementación de un sistema predictivo con redes neuronales para el control del comportamiento de la planta Festo MPS-PA"

**Autor:** Ing. Daysi Maribel Soria Mejía

**Director:** Dr. Freddy Geovanny Benalcázar Palacios, Mg.

Ambato – Ecuador

2021

# APROBACIÓN DEL TRABAJO DE TITULACIÓN

A la Unidad Académica de Titulación de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial. El Tribunal receptor de la Defensa del Trabajo de Titulación presidido por Ing. Elsa Pilar Urrutia Urrutia Mg., e integrado por los señores: Profesor Saba Rafael Infante Quirpa PhD. y la Ingeniera Clara Augusta Sánchez Benítez Mg., designados por la Unidad de Titulación de la Universidad Técnica de Ambato, para receptar el Trabajo de Titulación con el tema: "Implementación de un sistema predictivo con redes neuronales para el control del comportamiento de la planta Festo MPS-PA", elaborado y presentado por la Ing. Daysi Maribel Soria Mejía, para optar por el Grado Académico de Magíster en Matemática Aplicada; una vez escuchada la defensa oral del Trabajo de Titulación el Tribunal aprueba y remite el trabajo para uso y custodia en las bibliotecas de la Universidad Técnica de Ambato.

Firmado electrónicamente por:
ELSA PILAR URRUTIA

Ing. Elsa Pilar Urrutia Urrutia Mg.
Presidente y Miembro del Tribunal de Defensa

SABA RAFAEL INFANTE QUIRPA
Firmado digitalmente por SABA RAFAEL INFANTE QUIRPA
Fecha: 2021.07.13 10:34:39 -05'00'

Prof. Saba Rafael Infante Quirpa PhD.
Miembro del Tribunal de Defensa

Firmado electrónicamente por:
CLARA AUGUSTA SANCHEZ BENITEZ

Ing. Clara Augusta Sánchez Benítez Mg.
Miembro del Tribunal de Defensa

# AUTORÍA DEL TRABAJO DE TITULACIÓN

La responsabilidad de las opiniones, comentarios y críticas emitidas en el trabajo de titulación presentado con el tema: “Implementación de un sistema predictivo con redes neuronales para el control del comportamiento de la planta Festo MPS-PA”, le corresponde exclusivamente a la: Ing. Daysi Maribel Soria Mejía, autor bajo la dirección del: Dr. Freddy Geovanny Benalcázar Palacios, Mg., director del trabajo de investigación; y el patrimonio intelectual pertenece a la Universidad Técnica de Ambato.

**DAYSI MARIBEL SORIA MEJIA**

Nombre de reconocimiento C=EC, O=SECURITY DATA S.A. 2, OU=ENTIDAD DE CERTIFICACION DE INFORMACION, SERIALNUMBER=12022110239, CN=DAYSI MARIBEL SORIA MEJIA
Razón:
Localización:
Fecha: 2021-07-13T10:40:51.476-05:00

.....

Ing. Daysi Maribel Soria Mejía

.

Firmado electrónicamente por:
**FREDDY GEOVANNY BENALCAZAR PALACIOS**

.....

Dr. Freddy Geovanny Benalcázar Palacios, Mg.

# DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que el trabajo de titulación, sirva como un documento disponible para su lectura, consulta y procesos de investigación6n, según las normas de la Institución.

Cedo los derechos de mi trabajo de titulación, con fines de difusión pública, además apruebo la reproducción de éste, dentro de las regulaciones de la Universidad Técnica de Ambato.

**DAYSI MARIBEL SORIA MEJIA**

Nombre de reconocimiento C=EC, O=SECURITY DATA S.A. 2, OU=ENTIDAD DE CERTIFICACION DE INFORMACION, SERIALNUMBER=120221101239, CN=DAYSI MARIBEL SORIA MEJIA
Razón:
Localización:
Fecha: 2021-07-13T10:42:27.893-05:00

Ing. Daysi Maribel Soria Mejía

# Índice

# Índice de figuras

# Índice de tablas

# AGRADECIMIENTO

Agradezco en primer lugar a Dios quien me ha dado la sabiduría, fortaleza y la fuerza necesaria para culminar esta nueva etapa de mi vida.

A mi padre y madre por sus valores y consejos para seguir luchando cada día y ser una persona de bien.

A mi hermano Cristian porque con su amor y afecto me ha brindado confianza y me ha enseñado a seguir luchando para alcanzar mis objetivos.

A mi hermano Orlando a quien admiro por sus grandes enseñanzas y quien con su apoyo incondicional ha sido mi amigo, confidente y mi ejemplo para luchar sin temor y alcanzar grandes metas.

A mis docentes y amigos quienes con su apoyo y conocimiento me han permitido adquirir nuevos conocimientos.

A mi tutor quien con esfuerzo y dedicación me apoyo en el proceso de construcción del proyecto.

Ing. Daysi Maribel Soria Mejía

# DEDICATORIA

La presente investigación la dedico en primer lugar a Dios por bendecirme, protegerme y regalarme un día más de vida.

A mis padres Galo Soria y Lida Mejía por su amor incondicional, enseñanzas, paciencia, motivación, apoyo e inspiración para no desmayar en el camino de la vida.

A mis hermanos Omar Soria y Orlando Soria quienes con su comprensión, cariño, palabras de aliento han sido mi fortaleza para cumplir con mis objetivos y no decaer ante las adversidades de la vida.

Ing. Daysi Maribel Soria Mejía

**UNIVERSIDAD TÉCNICA DE AMBATO**

**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL**

**MAESTRÍA EN MATEMÁTICA APLICADA**

## INFORMACIÓN GENERAL

**TEMA**: "Implementación de un sistema predictivo con redes neuronales para el control del comportamiento de la planta Festo MPS-PA"

**AUTOR**: Ing. Daysi Maribel Soria Mejía

**DIRECTOR**: Dr. Freddy Geovanny Benalcázar Palacios, Mg.

**LÍNEA DE INVESTIGACIÓN**:

- Matemática Computacional

**FECHA**:27-04-2021

## RESUMEN EJECUTIVO

En el presente trabajo de investigación se realizó la implementación de un Controlador Predictivo de Matriz Dinámica (DMC) con Redes Neuronales para el proceso de control de nivel (líquido) de una planta Festo Compact Workstation del laboratorio de hidráulica y neumática de la Universidad Técnica de Ambato. La dinámica de la planta se encontró mediante el entrenamiento de una red neuronal feedforward; los datos de entrenamiento y testeo utilizados fueron obtenidos mediante la realización de un experimento que consiste en aplicar diferentes entradas escalón a la planta y la respuesta del sistema ante dicha entrada. El algoritmo implementado fue el de un controlador predictivo de matriz dinámica, para lo cual es necesario conocer el modelo matemático del proceso de nivel representado como función de transferencia, dicho modelo matemático fue construido mediante el método de regresión exponencial por mínimos cuadrados.

**Palabras Claves**: Control Predictivo, Función Transferencia, Matriz Dinámica, Optimización, Redes Neuronales, Sistemas Dinámicos, Redes Feedforward, Regresión Exponencial, Función de Activación, Función de Coste, Mínimos Cuadrados.

**TECHNICAL UNIVERSITY OF AMBATO**

**FACULTY OF SYSTEMS, ELECTRONIC AND INDUSTRIAL ENGINEERING**

**MASTER OF APPLIED MATHEMATICS**

## GENERAL INFORMATION

**THEME**: "Implementation of a predictive system with neural networks to control the behavior of the Festo MPS-PA plant"

**AUTHOR**: Ing. Daysi Maribel Soria Mejía

**DIRECTED BY**: Dr. Freddy Geovanny Benalcázar Palacios, Mg.

**LINE OF RESEARCH**:

- Computational Mathematics

**DATE**:27-04-2021

## EXECUTIVE SUMMARY

In the present research work, the implementation of a Dynamic Matrix Predictive Controller (DMC) with Neural Networks was carried out for the level control process (liquid) of a Festo Compact Workstation plant of the hydraulics and pneumatics laboratory of the Technical University of Ambato. The dynamics of the plant was found through the training of a feedforward neural network, the training and testing data used were obtained by conducting an experiment that consists of applying different step inputs to the plant and the response of the system to said input. The algorithm implemented was that of a dynamic matrix predictive controller, for which it is necessary to know the mathematical model of the level process represented as a transfer function, said mathematical model was built using the exponential regression method by least squares.

**Keywords**: Predictive Control, Transfer Function, Dynamic Matrix, Optimization, Neural Networks, Dynamic Systems, Feedforward Networks, Exponential Regression, Activation Function, Cost Function, Least Squares.

# CAPÍTULO I
# PROBLEMA DE INVESTIGACIÓN

## 1.1. Introducción.

En el presente trabajo de investigación se realizó la implementación de un Control Predictivo de Matriz Dinámica (DMC) con Redes Neuronales en una planta de nivel FESTO MPS-PA Compact Workstation, tanto el entrenamiento de la Red Neuronal como el algoritmo de control fueron desarrollados en el software OCTAVE y se trabajó con el microcontrolador Arduino-Uno para la lectura y escritura de datos de sensores y actuadores. El trabajo inició con el modelo matemático representado como función de transferencia del proceso de nivel de líquido, posteriormente se identifica la dinámica de la planta mediante el entrenamiento de la red neuronal feed-fordward, seguido de la identificación de la ley de control mediante la minimización de una función de coste; y finalmente se realizó la implementación del controlador en el proceso antes mencionado.

El modelo matemático del proceso de nivel de la planta de líquido denominado función de transferencia necesaria para implementar el algoritmo del controlador DMC con redes neuronales fue construido mediante un modelo de regresión exponencial de los datos capturados de la respuesta al escalón del sistema de nivel.

La dinámica del sistema de nivel de líquido se encontró mediante una red neuronal feed-forward con datos de entrenamiento obtenidos de un experimento que consiste en aplicar diferentes entradas escalón a la planta y la respuesta del sistema ante dicha entrada. La red neuronal feed-forward utilizada tiene: una capa de entrada, dos capas oculta y una capa de salida. La primera capa contiene dos entradas, las dos capas ocultas contienen 5 y 2 neuronas respectivamente con la función de activación tangente hiperbólica y finalmente la capa de salida con una sola neurona y la función de activación tangente hiperbólica, la red neuronal feed-forward fue entrenada independientemente del algoritmo de control DMC.

La ley de control del algoritmo DMC se encontró mediante la minimización de la función de coste la cual contiene: los setpoints deseados, predicciones y esfuerzos de control. Para el presente trabajo no se consideraron restricciones en la función de coste, por lo tanto la solución a la minimización de la función puede obtenerse analíticamente

calculando su derivada e igualándola a $0$.

Encontrada la dinámica de la planta de nivel de líquido mediante el entrenamiento de una red neuronal feed-fordward y utilizando la ley de control del DMC se logró implementar correctamente el algoritmo predictivo en la planta de nivel de la célula FESTO MPS-PA Compact Workstation.

El presente documento se encuentra distribuido en cinco capítulos que se detallan a continuación: Dentro del primer capítulo se da a conocer la justificación y los objetivos de la investigación; en el segundo capítulo se describe el estado del arte y la fundamentación teórica más importante para desarrollar el modelo predictivo; el tercer capítulo describe la metodología aplicada; en el cuarto capítulo se describe los resultados y discusiones realizadas en base al controlador predictivo; en el quinto capítulo se da a conocer las conclusiones y recomendaciones de la investigación.

## 1.2. Justificación.

El control predictivo es la metodología de control avanzado de procesos más ampliamente estudiada por la comunidad científica y altamente implementada en aplicaciones industriales, son utilizados en muchas áreas donde se requiere un control de alta calidad. Los algoritmos de control predictivo utilizan un modelo del proceso explícito para predecir el comportamiento futuro del proceso para un horizonte de predicción determinado. En el control predictivo basado en modelos o Model predictive Control(MPC), la señal de control se determina minimizando una función de coste en cada instante de tiempo sujeto a restricciones, de ser el caso[1]. Una de las mayores ventajas del control predictivo es que puede hacer frente de forma eficaz a una gran cantidad de restricciones tales como: la señal de control, incremento de control, salida de la planta, etc. El control predictivo basado en el modelo se puede aplicar en muchos sistemas, como por ejemplo: sistemas multivariables, de bucle abierto, inestables, no lineales o con retrasos prolongados[2]

Aunque los procesos industriales suelen contener no linealidades complejas, la mayoría de las estrategias de MPC se basan en un control convencional que utilizan modelos dinámicos lineales, pero existen también no linealidades complejas y debido a estos problemas es necesario aplicar varios procedimientos computacionales o aproximados para su solución. Para sistemas altamente no lineales se puede esperar que las técnicas de control basadas directamente en modelos no lineales proporcionen un rendimiento significativamente mejorado. En este contexto, las poderosas propiedades del aproximador de funciones de las redes neuronales las hacen útiles para representar modelos o controlado-

res no lineales[3].

Recientemente, las redes neuronales se han convertido en una herramienta atractiva en la construcción de modelos para varios tipos de sistemas complejos no lineales, así también para la implementación de sistemas de control donde se utiliza un enfoque basado en optimización numérica para representar una ley de control óptima necesaria para el correcto funcionamiento de los sistemas industriales. Cuando una red neuronal es utilizada para representar sistemas complejos no lineales y se combina con la metodología de control del MPC se ha logrado mejores resultados con respecto a sistemas de control tradicionales [4].

## 1.3. Objetivos.

### 1.3.1. Objetivo general.

- Implementar un sistema predictivo con redes neuronales para el control del comportamiento de la planta Festo MPS-PA.

### 1.3.2. Objetivos específicos.

- Identificar el tipo de estructuras de redes neuronales utilizados para la implementación de controladores predictivos.
- Analizar la información disponible obtenida en trabajos previos y determinar las variables de interés para el entrenamiento y testeo de la red neuronal.
- Construir la red neuronal utilizando la información disponible
- Entrenar la red neuronal para determinar el comportamiento del sistema de nivel de la planta Festo Compact Workstation.

# CAPÍTULO II
# ANTECEDENTES INVESTIGATIVOS

## 2.1. Antecedentes investigativos.

Durante los últimos 30 años, la teoría del control predictivo de modelos (MPC o MBPC-Model Based Predictive Control) se han desarrollado rápidamente y se han realizado gran cantidad de aplicaciones tanto industriales como académicas. Sin embargo, frente a los crecientes requisitos sobre el control de optimización restringido que surgen del rápido desarrollo de la economía y la sociedad, la teoría y tecnología actuales de MPC aún se enfrentan a grandes desafíos.

En la publicación de Yu-Geng, De-Wei y Shu [5] titulada "Model predictive control status and challenges" se revisa brevemente y analiza el desarrollo de la teoría MPC y algunas de las aplicaciones industriales; además, analiza las limitaciones de la teoría y la tecnología actuales del MPC y se señala la necesidad de fortalecer la investigación del MPC con respecto a mejorar su efectividad y utilidad en aplicaciones industriles como académicas.

Rankovic, Radulovic, Grujovic y Divac [6] en el trabajo titulado "Neural network model predictive control of nonlinear systems using genetic algorithms" consideran la síntesis del controlador predictivo no lineal. Los resultados de la simulación para diferentes setpoints o consignas muestran que los controladores predictivos se pueden aplicar con éxito para el control, donde los valores óptimos de las señales de control se obtienen mediante el algoritmo genético, el cual representa una técnica de optimización global.

Maiworm, Bäthge y Findeisen [7] en el trabajo titulado "Scenario-based model predictive control: Recursive feasibility and stability" consideraron un MPC robusto y multiescenario, en el cual adapta el enfoque clásico de MPC nominal para establecer la viabilidad y estabilidad recursiva. Para el caso de múltiples escenarios realiza una simulación para un control climático en construcciones civiles.

Smarra, Jain, Rubeis, Ambrosini, D'Innocenzo y Mangharam [8] en el trabajo titulado "Data-driven model predictive control using random forests for building energy optimization and climate control" presentan una idea para el control predictivo basada en datos de edificios históricos que aprovechan los algoritmos de aprendizaje automático, a éste enfo-

que lo denominaron Control Predictivo del Modelo basado en Datos (DPC) y lo aplicaron a tres estudios de casos diferentes para demostrar su rendimiento, escalabilidad y solidez.

Honc, Sharma, Abraham, Dušek y Pappa [9] en el trabajo titulado "Teaching and practicing model predictive control" tratan sobre las acciones de la enseñanza, práctica y aplicación de un Control Predictivo Basado en Modelo, en el cual se simularon algoritmos de control, perturbaciones y ruido en las mediciones para un sistema de primer orden a fin de controlar el nivel del agua.

Bakosová y Oravec [10] en el trabajo titulado "Robust model predictive control for heat exchanger network" investigaron la implementación del Control Robusto Predictivo de Modelo (RMPC) para el control de una red de intercambiadores de calor debido a la presencia de no linealidades del sistema, parámetros de procesos variables, perturbaciones internas y externas. Los resultados confirmaron que el uso de RMPC condujo a un menor consumo de refrigerante en comparación con el consumo logrado mediante el control cuadrático lineal óptimo.

Vasickaninová, Bakosová, Mészáros y Klemes [4] en el trabajo titulado "Neural network predictive control of a heat exchanger" demostraron que el uso de la estructura de control predictivo de redes neuronales (NNPC) para el control de los procesos térmicos puede generar ahorros de energía. La eficacia del enfoque de control descrito se verifica mediante experimentos de simulación y se elige un intercambiador de calor tubular como proceso controlado. El NNPC del intercambiador de calor se compara con el control PID clásico.

Entchev, Yang, Ghorab, Rosato y Sibilio [11] en el trabajo titulado "Energy, economic and environmental performance simulation of a hybrid renewable microgeneration system with neural network predictive control" compararon un enfoque basado en Redes Neuronales Artificiales (ANN) con un control convencional de on/off aplicado a la operación de un sistema térmico fotovoltaico de bomba de calor, que sirve a una sola casa con fines de calefacción y refrigeración. La comparación se realizó en términos de capacidad para mantener los niveles deseados de confort interior, consumo de energía primaria, costos operativos y emisiones equivalentes de dióxido de carbono durante una semana. Los resultados mostraron que el enfoque ANN es potencialmente capaz de aliviar la incomodidad térmica asociada con fenómenos de sobrecalentamiento/sobreenfriamiento.

Jung, H. Kim, J. Kim, Lee y Park [12] en el trabajo titulado "Model predictive control via output feedback neural network for improved multi-window greenhouse ventilation control" realizó un estudio de una nueva lógica de control de ventilación para invernaderos de múltiples ventanas que utiliza una predicción de una red neuronal de retroali-

mentación (OFNN); el sistema de control fue implementado en un experimento de campo durante seis días, comparando dos invernaderos uno impulsado por la lógica de control convencional y el otro por lógica de control predictivo neuronal.

Tian, Li, y Wang [13] en el trabajo titulado "Ts fuzzy neural network predictive control for burning zone temperature in rotary kiln with improved hierarchical genetic algorithm" propusieron un método de control predictivo basado en un algoritmo genético jerárquico mejorado y una red neuronal difusa para mejorar el rendimiento del control de la temperatura de la zona de combustión en el horno rotatorio de cal. Este método de control utilizó la red neuronal difusa para construir un modelo predictivo no lineal de la temperatura de la zona de combustión en un horno rotatorio. Se utilizó un algoritmo genético jerárquico mejorado para la optimización de la variable de control óptima.

Elsisi [14] en el trabajo titulado "Design of neural network predictive controller based on imperialist competitive algorithm for automatic voltage regulator" propone el controlador predictivo de red neuronal (NN), que combina las ventajas de NN y el control predictivo para el regulador automático de voltaje (AVR). El algoritmo competitivo imperialista (ICA) se presenta en este artículo como una nueva técnica de inteligencia artificial. Se compara el rendimiento del controlador predictivo NN diseñado basado en el ICA con el controlador predictivo diseñado NN basado en el algoritmo genético y el controlador convencional proporcional-integral-derivado (PID).

Heidari [15] en el trabajo titulado "Improving efficiency of photovoltaic system by using neural network mppt and predictive control of converter" propone un nuevo método para extraer la máxima energía de los sistemas fotovoltaicos. La red neuronal artificial (ANN) se utiliza para rastrear la potencia máxima según el nivel de irradiancia y la temperatura. Además, de la ANN se utiliza un controlador predictivo para maximizar la eficiencia del convertidor elevador. Los resultados de la simulación verifican el adecuado desempeño del método propuesto para maximizar la extracción de energía del sistema fotovoltaico.

Karayel [16] en el trabajo titulado "Prediction and control of surface roughness in cnc lathe using artificial neural network" presenta un enfoque de red neuronal para la predicción y el control de la rugosidad de la superficie en un torno controlado numéricamente por computadora (CNC), utilizó una red neuronal multicapa de alimentación hacia adelante y entrenó el modelo de red utilizando el algoritmo de gradiente conjugado escalado (SCGA), que es un tipo de retropropagación. Los resultados de la investigación verifican que la rugosidad se puede determinar antes de una operación de mecanizado utilizando una red neuronal y el control del algoritmo.

Shin, Jun y Kim [17] en el trabajo titulado “Dynamic control of intelligent parking guidance using neural network predictive control” proponen un enfoque de control predictivo basado en redes neuronales de modo que se pueda lograr el mejor rendimiento del sistema de guía de estacionamiento inteligente. El método propuesto puede mejorar el rendimiento de un sistema de guía de estacionamiento inteligente a través del control dinámico en la selección del mejor estacionamiento. Para evaluar el enfoque propuesto se han realizado pruebas de simulación y comparación con un modelo tradicional.

Sharma and Singh [18] en el trabajo titulado “Model predictive control and neural network predictive control of tame reactive distillation column” presentan el trabajo en tres estrategias de control diferentes: control Proporcional Integral Derivativo (PID) convencional, control predictivo del modelo y control predictivo de la red neuronal (NNPC) implementados en una columna de destilación reactiva. Todos estos controladores se comparan y se encuentra que el NNPC y MPC brindan una señal rendimiento de control no brusca y mejor que el controlador PID.

Aldair [19] en el trabajo titulado “Hardware implementation of the neural network predictive controller for coupled tank system” presenta el diseño de un controlador predictivo basado en una red neuronal para controlar el nivel de líquido del sistema de tanque acoplado. El controlador predictivo de red neuronal que se analiza en este documento utiliza un modelo de red neuronal de una planta no lineal para predecir el rendimiento futuro de la planta. Los resultados de la simulación se comparan con el control PID. Los resultados muestran la efectividad del uso del controlador predictivo neuronal para el sistema de tanque acoplado.

## 2.2. Fundamentación teórica.

### 2.2.1. Características de respuesta de los sistemas físicos.

Los sistemas físicos de primer orden matemáticamente están definidos por ecuaciones diferenciales lineales y no lineales; por ejemplo, un sistema eléctrico resistencia-capacitancia (RC)[20]:

$$\frac{dV_c(t)}{dt} + \frac{1}{RC}V_c(t) = \frac{1}{RC}V(t)$$

donde:

$V_c$ = Voltaje en el capacitor.
$R$ = Valor de la resistencia.
$C$ = capacitancia del condensador.

$V$ = Voltaje de fuente.

**Función de transferencia.**

La función de transferencia ($G(s)$) es un modelo matemático que a través de un cociente relaciona la respuesta de un sistema o señal de salida ($Y(s)$) con una señal de entrada o excitación ($R(s)$)[20].

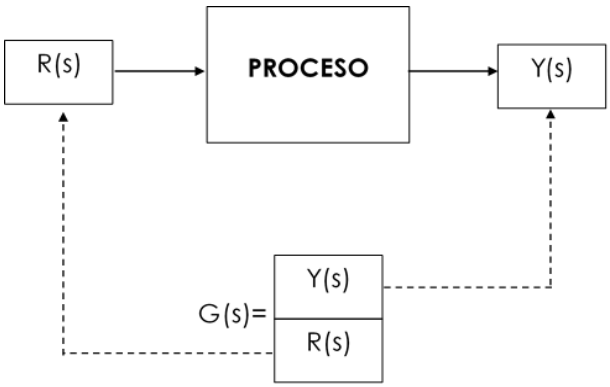


Figura 1: Función de transferencia de un sistema.

Fuente: Elaborado por la autora.

**Polos y ceros de una función de transferencia.**

Se denomina ceros a las raíces del numerador Y(s) y polos a las raíces del denominador R(s)[20].

**Sistemas físicos de primer orden**

Un sistema físico de primer orden está definido por una ecuación diferencial de primer orden en donde: la variable independiente es $t$ y las funciones incógnitas $r(t)$ e $y(t)$; además, se consideran constantes los coeficientes $a$, $b$ y $c$

$$a\frac{dy}{dt} + by = cr(t) \tag{1}$$

De la ecuación (1) se determinará la función de transferencia del sistema $G(s)$ que se encuentra en el dominio complejo $s$. Para obtener la función de transferencia $G(s)$ en términos de polos y ceros, se transforma en unitario el coeficiente con mayor derivada en la ecuación diferencial[20]:

$$(as + b)Y(s) = cR(s)$$

y

$$Y(s) = R(s)\frac{(\frac{c}{a})}{s + (\frac{b}{a})}$$

$$G(s) = \frac{Y(s)}{R(s)} = \frac{(\frac{c}{a})}{s + (\frac{b}{a})} = \frac{(\frac{c}{a})}{s + a_0} \tag{2}$$

La formula (2) se representa como:

$$G(s) = \frac{\frac{c}{a}}{s + \frac{b}{a}} \frac{\frac{a}{b}}{\frac{a}{b}} = \frac{\frac{c}{b}}{(\frac{a}{b})s + 1} = \frac{K}{\tau s + 1} \tag{3}$$

donde

$K$ = Representa el coeficiente amplificación entre salida y entrada (Ganancia del sistema).
$\tau$ = Indica en el sistema la constante de tiempo($s$).

La formula (3) se expresa en el sistema en términos de la constante de tiempo $\tau$. Conocida la función de transferencia $G(s)$, se determinará su respuesta o salida $y(t)$ (solución a la ecuación diferencial 1) cuando se emplea una entrada escalón $r(t) = Au(t)$[20].

donde:

$R(s) = \pounds\{r(t)\} = \frac{A}{s}$

$r(t)$ = Entrada escalón aplicado a un sistema.
$A$ = Valor de la entrada escalón.
$u(t)$ = Sistema de primer orden.

La respuesta del sistema se alcanza al ordenar la formula(3):

$$Y(s) = R(s)G(s) = \frac{A}{s} * \frac{\frac{c}{b}}{(\frac{a}{b})s + 1}$$

aplicando fracciones parciales se obtiene:

$$Y(s) = \frac{A(\frac{c}{b})}{s(\frac{a}{b}s + 1)} = \frac{C_1}{s} + \frac{C_2}{(\frac{a}{b}s + 1)} = \frac{A(\frac{c}{b})}{s} - \frac{A(\frac{c}{b})}{(\frac{a}{b}s + 1)}$$

aplicando la transformada inversa de laplace se tiene la solución a la ecuación diferencial de primer orden.

$$\mathcal{L}^{-1}\{Y(s)\} = y(t) = A(\frac{c}{b})[1 - e^{-(\frac{a}{b})t}] \tag{4}$$

o en términos de la ganancia del sistema $K = c/b$ y la constante de tiempo $\tau = a/b$ obtenemos:

$$y(t) = AK[1 - e^{-\frac{t}{\tau}}] \tag{5}$$

Su valor final será:

$$y(\infty) = \lim_{t \to \infty} y(t) = AK$$

si el sistema es estable.

Para obtener las características de respuesta al escalón de un sistema de primer orden se debe normalizar la formula (5) es decir $AK = 1$; tambien, se establece que $a_0 = 1/\tau$, obteniendo[20]:

$$y(t) = [1 - e^{-a_0 t}] \tag{6}$$

en el cual la respuesta $y(t)$ tiene dos términos: el factor exponencial decreciente $e^{-t}$ y un valor constante igual a la unidad. La gráfica de $y(t)$ se observa en la figura 2 [20].

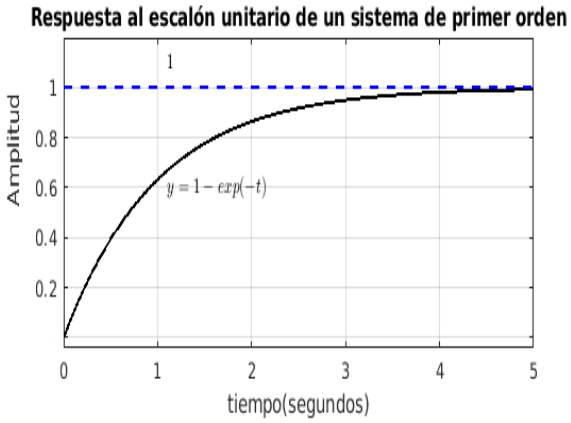


Figura 2: Respuesta al escalón de un sistema de primer orden.[20]

$$y(t) = y_{régimen\ transitorio}(t) + y_{régimen\ de\ estado\ estable}(t) \tag{7}$$

La figura 3 indica la respuesta natural y respuesta forzada de $y(t) = 1 - e^{-t}$ [20].

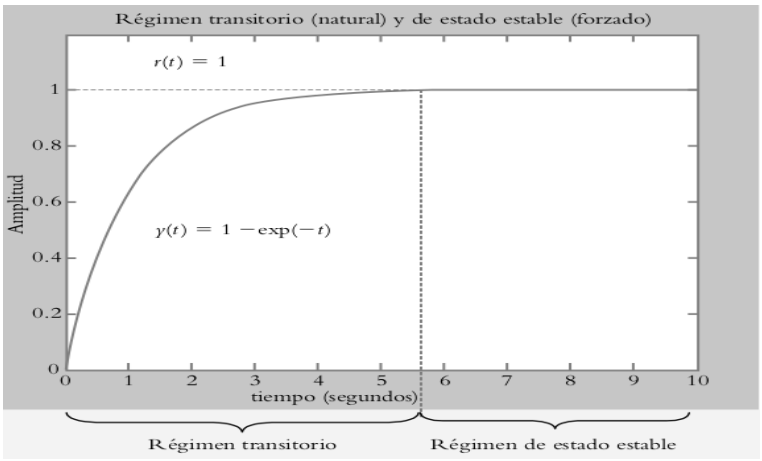


Figura 3: Respuesta de un sistema al escalón.[20]

**Constante de tiempo $\tau$.**

La formula (6) se cuantificará para varios valores de $a_0$. Para llegar a un estado estable, es necesario que $t \to \infty$; pero, se debe determinar un valor práctico de tiempo afin que el sistema llegue a su valor final.

El tiempo $t_a$ se refiere al tiempo que necesita el sistema para llegar a su valor final, y corresponde a [20]:

$$t_a = 4\tau \tag{8}$$

### 2.2.2. Regresión por mínimos cuadrados.

Con frecuencia los datos experimentales se comportan como los esquematizados en la figura 4. Una inspección visual de esos datos sugiere una posible relación entre $y$ y $x$; es decir, la tendencia general indica que valores altos de $y$ están asociados con valores altos de $x$.

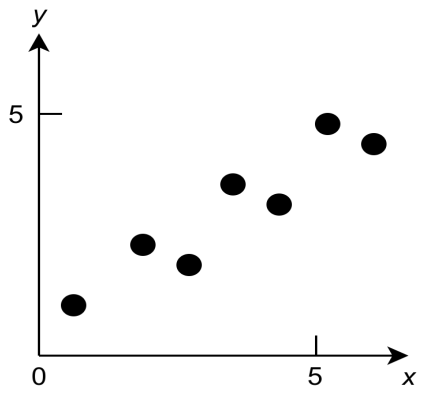


Figura 4: Datos que muestran un error significativo.[21]

Una estrategia apropiada para encontrar esa tendencia general consiste en obtener una función de aproximación que se ajuste a la forma o a la tendencia general de los datos, sin coincidir necesariamente en todos los puntos. La figura 5 ilustra cómo se utiliza una línea recta para caracterizar de manera general la tendencia de los datos sin pasar a través de algún punto específico. Una manera para determinar la recta de la figura 5 es inspeccionar en forma visual los datos graficados y después trazar una "mejor" línea a través de los puntos. Aunque tales procedimientos "a ojo" apelan al sentido común y son válidos para cálculos "superficiales", sin embargo resultan deficientes por ser arbitrarios[21].

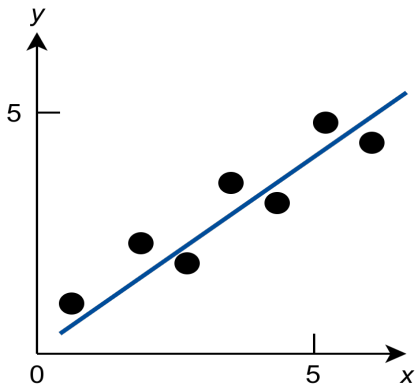


Figura 5: Resultado mediante el ajuste por mínimos cuadrados.[21]

Para dejar a un lado dicha subjetividad se debe encontrar algún criterio para establecer una base para el ajuste. Una forma de hacerlo es obtener una curva que minimice la discrepancia entre los puntos y la curva. Una técnica para lograr tal objetivo es la llamada *regresión por mínimos cuadrados* (El método de los mínimos cuadrados se utiliza para calcular la recta de regresión lineal o no lineal que minimiza los residuos, esto es, las diferencias entre los valores reales y los estimados por la recta)[21].

**Regresión Lineal.**

El ejemplo más simple de una aproximación por mínimos cuadrados es ajutar una línea recta a un conjunto de observaciones definidas por puntos: $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$. La expresión matemática para la línea recta es[21]:

$$y = a_0 + a_1 x + e \tag{9}$$

donde $a_0$ y $a_1$ son coeficientes que representan la intersección con el eje $y$ y la pendiente, respectivamente, $e$ es el error o diferencia entre el modelo y las observaciones, el cual se representa al reordenar la ecuación (9) como:

$$e = y - a_0 - a_1 x$$

Así, el *error* o *residuo* es la discrepancia entre el valor verdadero de $y$ y el valor aproximado $a_0 + a_1 x$[21].

**Criterio para un "mejor" ajuste.**

La estrategia utilizada consiste en minimizar la suma de los cuadrados de los residuos

entre la $y$ medida y la $\hat{y}$ calculada con el modelo lineal[21]

$$S_r = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_{i,medida} - \hat{y}_{i,modelo})^2 = \sum_{i=1}^{n} (y_i - a_0 - a_1 x_1)^2 \tag{10}$$

Este criterio tiene varias ventajas, entre ellas el hecho de que se obtiene una línea única para cierto conjunto de datos. Antes de analizar tales propiedades, presentaremos una técnica para determinar los valores de $a_0$ y $a_1$ que minimizan la ecuación (10).

**Ajuste de una línea recta por mínimos cuadrados.**

Para determinar los valores de $a_0$ y $a_1$ , la ecuación (10) se deriva con respecto a cada uno de los coeficientes[21]:

$$\frac{\partial S_r}{\partial a_0} = -2 \sum (y_i - a_0 - a_1 x_i)$$

$$\frac{\partial S_r}{\partial a_1} = -2 \sum [(y_i - a_0 - a_1 x_i) x_i]$$

Las sumatorias van desde $i = 1$ hasta $n$. Al igualar estas derivadas a cero, se dará como resultado un $S_r$ mínimo. Si se hace esto, las ecuaciones se expresan como

$$0 = \sum y_i - \sum a_0 - \sum a_1 x_i$$

$$0 = \sum y_i x_i - \sum a_0 x_i - \sum a_1 x_i^2$$

Se observa que $\sum a_0 = n a_0$, expresamos las ecuaciones como un conjunto de dos ecuaciones lineales simultáneas, con dos incógnitas ($a_0$ y $a_1$):

$$n a_0 + (\sum x_i) a_1 = \sum y_i \tag{11}$$

$$a_0 = \frac{\sum y_i - (\sum x_i) a_1}{n} \tag{12}$$

$$(\sum x_i) a_0 + (\sum x_i^2) a_1 = \sum x_i y_i \tag{13}$$

$$a_0 = \frac{\sum x_i y_i - (\sum x_i^2) a_1}{(\sum x_i)} \tag{14}$$

Una vez que se despeja las ecuaciones (12) y (15) se procede a igualar, obteniendo la siguiente ecuación:

$$\frac{\sum y_i - (\sum x_i) a_1}{n} = \frac{\sum x_i y_i - (\sum x_i^2) a_1}{(\sum x_i)} \tag{15}$$

Despejando $a_1$ se obtiene:

$$a_1 = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{n\sum x_i^2 - (\sum x_i)^2}$$

Este resultado se utiliza conjuntamente con la ecuación (11) para obtener:

$$a_0 = \bar{y} - a_1 \bar{x}$$

donde $\bar{y}$ y $\bar{x}$ son los valores medios de los datos de $y$ y $x$, respectivamente. [21]

### 2.2.3. Modelos de neuronas y redes neuronales artificiales.

En Inteligencia Artificial se trabaja en un modelo de neurona, en el que su valor de salida depende únicamente de la suma ponderada de pesos sinápticos y entradas [22]:

$$V_j = \sum_{i=1}^{n} x_i w_{ij}$$

donde $w_{ij}$ son los pesos de las entradas.

El modelo que utilizamos como base para construir redes neuronales artificiales es:

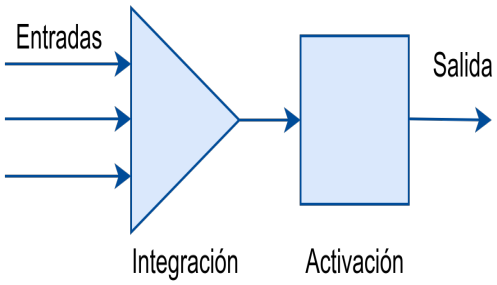


Figura 6: Modelo simplificado de neurona con integración y activación.[22]

El modelo de neuronas artificiales consta de dos etapas. En una primera etapa, las entradas de otras neuronas se combinan considerando los pesos de las sinapsis. El resultado de la primera etapa es la entrada o excitación neta de la neurona. En la segunda etapa, la entrada neta se utiliza para determinar el valor de salida de la neurona que se propagará a otras neuronas. El sesgo es un parámetro adicional de la neurona vinculado a una entrada fija con valor 1 que ejerce el papel de corrientes de fuga en una neurona biológica[22].

La mayoría de los modelos de redes neuronales artificiales utilizan un sesgo externo o *bias* ($b_j$):

$$V_j = b_j + \sum_{i=1}^{n} x_i w_{ij}$$

Los pesos sinápticos definen la fuerza de una conexión entre dos neuronas y son representadas con la letra w. Si establecemos $w_{0j} = b_j$ y $x_0 = 1$, la expresión anterior queda como[22]:

$$V_j = \sum_{i=0}^{n} x_i w_{ij}$$

En el modelo de una neurona artificial, la salida de la neurona es responsabilidad de una fase de activación independiente de la fase de integración de entradas. La activación de una neurona artificial no reproduce una entrada neta, sino que se aplica una transformación no lineal a esa entrada neta, esta transformación es no lineal para construir redes con múltiples capas si las transformaciones fuesen lineales, el resultado final del cálculo seguiría siendo una función lineal de las entradas.

Cualquier red neuronal compuesta por elementos lineales siempre podría sustituirse por una red formada por una única capa de neuronas lineales. Las neuronas lineales sólo pueden representar funciones lineales. Como la mayor parte de los problemas reales incorporan algún tipo de no linealidad, nos interesa que nuestras redes sean capaces de representar esas no linealidades[22].

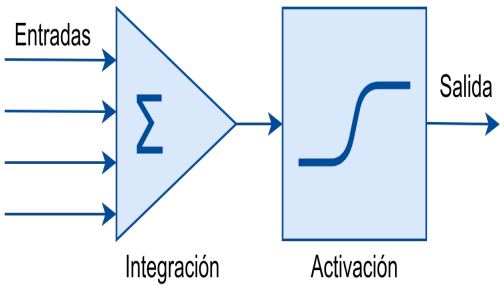


Figura 7: Modelo estándar de neurona artificial con función de activación no lineal.[22]

Las redes neuronales artificiales están formadas por conjuntos de neuronas agrupadas en capas, todas las neuronas de una misma capa comparten ciertas características. Denotaremos por $x_i$ a cada una de las $n$ entradas recibidas por una capa de neuronas formada por $m$ neuronas. La salida de cada neurona la representaremos por $y_j$, habitualmente un valor binario o real[22].

Los pesos sinápticos $w_{ij}$ los utilizaremos para modelar las conexiones de entrada a las neuronas, siendo el peso $w_{ij}$ el peso asociado a la sinapsis que conecta la entrada $i$-ésima con la neurona $j$-ésima. Los pesos tomarán valores reales positivos para modelar conexiones excitatorias (suma de pesos sinápticos y entradas) y negativos para modelar conexiones inhibitorias (resta de pesos sinapticos y entradas)[22].

En la etapa de integración de entradas, una neurona artificial combina las diferentes entradas $x_i$ con sus pesos para determinar su entrada neta $z_j$ o $net_j$:

$$z_j = net_j = \sum_{i=1} w_{ij} x_i$$

En la etapa de activación una neurona artificial utiliza el valor asociado a su entrada neta para generar una salida $y_j$; el modelo más habitual genera una salida de tipo numérico a partir de la entrada neta de la neurona[22]:

$$y_j = f(z_j) = f(net_j) = f(\sum_{i=1} w_{ij} x_i)$$

donde la función $f$ es la función de activación de la neurona(no lineal).

Cuando diseñamos una red neuronal artificial de varias capas, la notación vectorial puede simplificar varias expresiones. En lugar de definir la salida de una única neurona $y_j$, definiremos las salidas de todas las neuronas de una capa de neuronas mediante notación vectorial:[22]

$$y = f(z) = f(\mathbf{W}x + b)$$

O asumir que existe una entrada fija $x_0 = 1$ y los sesgos van ya representados en la matriz de pesos **W**.

**Arquitecturas de las redes neuronales artificiales.**

Analizado el modelo más utilizado de neuronas artificiales, empezaremos a combinar colecciones de neuronas para resolver varios problemas.

**Redes feed-forward.**

Las aplicaciones de regresión y clasificación con redes neuronales emplean múltiples capas ocultas de neuronas. Las neuronas de cada capa suelen ser independientes entre sí y operan en paralelo. Las distintas capas se conectan entre sí de tal forma que la salida de la capa $i$ se utiliza como entrada en la capa $i + 1$. Estas redes reciben el nombre de redes neuronales *feed-forward*[22].

Las capas de una red multicapa se dividen en dos categorías: capas visibles y capas ocultas. Las capas visibles son aquellas capas observables desde el exterior de la red (primera y última). Todas las capas intermedias reciben el nombre de capas ocultas, por no ser observables directamente desde el exterior. En una red neuronal de tipo *feed-forward*, encontramos varias situaciones indicadas a continuación dependiendo del número de capas ocultas utilizadas[22]:

- Redes simples con una única capa.

Las neuronas de la capa de entrada, que reciben las señales de entrada del exterior redistribuyen esas entradas a las neuronas de la capa de salida. Aunque existen dos capas, una de entrada y una de salida, este tipo de redes se considera ser unicapa[22].

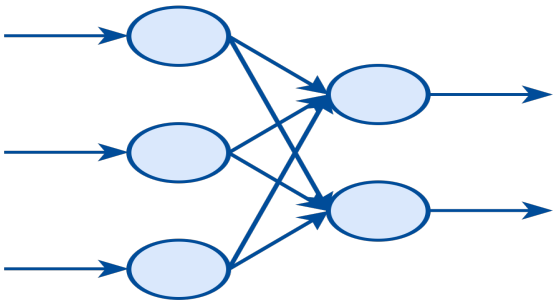

Figura 8: Red neuronal de una capa de entrada y una capa de salida.[22]

- Redes multicapa, con una capa oculta.

En una red neuronal simple la capa de entrada y la capa de salida son visibles desde el exterior de la red. Si añadimos nuevas capas intermedias, estas capas ya no serán visibles desde el exterior, lo que nos obligará a utilizar algoritmos como *backpropagation* para ajustar sus parámetros internos. La presencia de una única capa oculta ya dota a la red multicapa de su capacidad de aproximador universal (capacidad de aprender)[22].

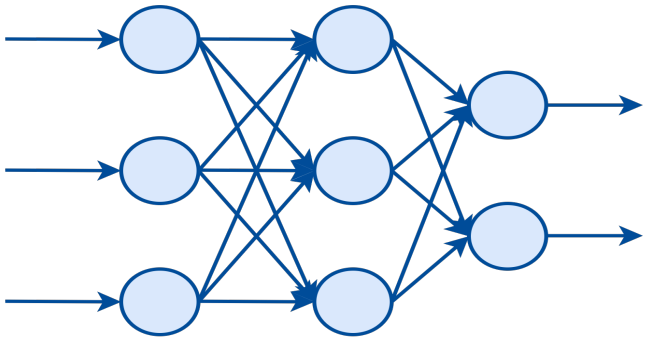

Figura 9: Red neuronal multicapa con una única capa oculta.[22]

- Redes profundas [*deep networks*] con varias capas ocultas.

Para aprovechar el potencial de las técnicas de *deep learning* a la hora de diseñar soluciones modulares para problemas complejos, las redes neuronales actuales suelen incluir múltiples capas ocultas[22].

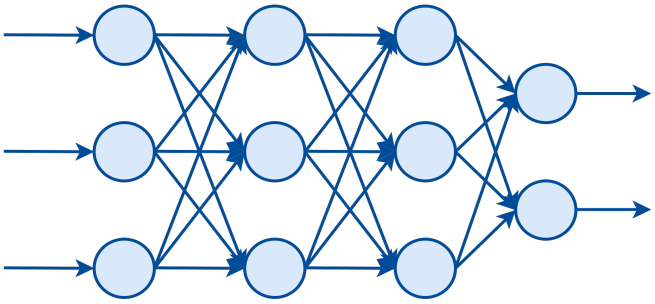

Figura 10: Red neuronal profunda con múltiples capas ocultas.[22]

En todas las redes, las conexiones pueden ser excitatorias o inhibitorias. Las funciones de activación bipolares como la tangente hiperbólica, pueden resultar útiles en situaciones en las que un cambio de signo en la salida de la neurona influye en el comportamiento de las neuronas de las siguientes capas. Si la salida bipolar de una neurona se conecta a la entrada de otra neurona con un peso sináptico positivo, la conexión será excitatoria cuando la salida sea positiva e inhibitoria cuando sea negativa [22].

### 2.2.4. Entrenamiento de Redes.

Las redes neuronales artificiales organizadas por capas, sin realimentación, constituyen la topología de red neuronal más utilizada en la práctica. Su denominación oficial es feed-forward neural networks (FFNNs). En una red multicapa de tipo feed-forward, las neuronas se organizan por capas. La arquitectura típica de una red neuronal multicapa consta de tres capas o más[22]:

- Capa de entrada.
- Capa oculta.
- Capa de salida.

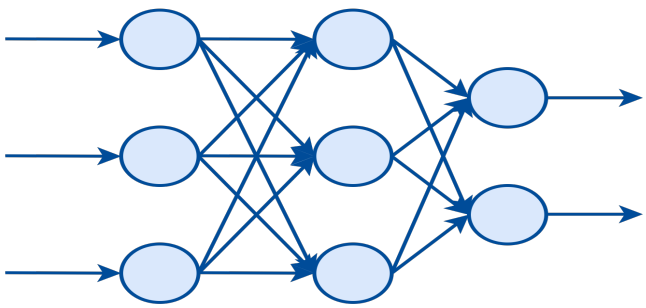

Figura 11: Red multicapa profunda de tipo feed-forward con más de una capa oculta.[22]

Cada una de las capas recibe sus entradas de la capa inmediata anterior en la red (salvo la capa de entrada, que recibe sus entradas del exterior y se limita a propagarlas a la capa oculta). Las salidas de cada capa sirven de entradas a la capa inmediata posterior en la red multicapa. Habitualmente, todas las salidas de una capa se distribuyen a todas las neuronas de la siguiente capa, formando capas completamente conectadas(*fully-connected layers*)[22] como se indica en la figura 11.

El entrenamiento de una red multicapa se suele realizar utilizando un algoritmo de propagación de errores hacia atrás denominado *backpropagation*. El término *backpropagation* se refiere únicamente a la propagación de errores hacia atrás, no al algoritmo de entrenamiento completo. La propagación de errores hacia atrás permite calcular el gradiente del error con respecto a los diferentes parámetros de la red (esto es, cómo varía el error conforme varían los parámetros de la red). Combinado con una técnica de optimización como el gradiente descendente se obtiene un algoritmo de entrenamiento para redes multicapa. El gradiente obtenido por *backpropagation* indica en qué sentido han de modificarse los parámetros de la red mientras que la técnica de optimización es la que realiza el ajuste de dichos parámetros. El gradiente se calcula para una función de error, también conocida como función de pérdida (*loss function*), y el método de optimización ajusta los pesos de la red con el objetivo de minimizar esa función de error o pérdida[22].

- Aprendizaje supervisado (gradiente descendente y *backpropagation*).

La red recibe una realimentación directa para cada uno de los datos del conjunto de entrenamiento. Esa señal, en forma de salida deseada, se puede utilizar directamente para ajustar los parámetros de la red, tal como hacen las técnicas de optimización usadas con *backpropagation*.

**Entrenamiento de redes multicapa.**

Las redes multicapa de tipo feed-forward en ocasiones se denominan *backpropagation networks* porque el algoritmo de entrenamiento que se suele utilizar está basado en la propagación del error hacia atrás; esto es, en el uso de *backpropagation*[22].

Las redes multicapa usadas en deep learning tienen una capa de entrada, múltiples capas ocultas y una capa de salida. Se puede ver como una función matemática $f$ que, a partir de un vector de entrada $x$, obtiene un vector de salida $y = f(x)$. Dado un conjunto de entrenamiento en forma de pares $(x, y)$, el objetivo del algoritmo de entrenamiento es el de ser capaz de aproximar la función $f$ de forma que para cada posible entrada $x$ se obtenga una salida $\hat{y} = f(x)$ lo más similar posible a la observada en el conjunto de entrenamiento. Para que una red neuronal sea capaz de generalizar correctamentese se tiene

que ajustar su capacidad hasta un nivel que resulte adecuado para la función que se pretende modelar. Este ajuste de la capacidad se puede realizar, o bien ajustando los parámetros que determinan la topología de la red (habitualmente denominados hiperparámetros), o bien utilizando técnicas de regularización (cualquier técnica que nos permita reducir el error sobre el conjunto de prueba sin aumentar demasiado el error sobre el conjunto de entrenamiento)[22].

**Entrenamiento de neuronas ocultas.**

Se desea entrenar redes neuronales con múltiples capas, de forma que el entrenamiento de las neuronas de sus capas ocultas les permita ser capaces de encontrar, automáticamente, características que resulten útiles en la tarea particular para la que entrenamos a la red.

Se utilizará la regla de la cadena para describir la derivada parcial del error con respecto a cada parámetro de una red neuronal multicapa como un producto de otras derivadas parciales[22]:

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial y_j}\frac{\partial y_j}{\partial z_j}\frac{\partial z_j}{\partial w_{ij}}$$

En este caso, se ha utilizado dos veces la regla de la cadena (de atrás hacia adelante):

- La derivada parcial de la entrada neta $z_j$ de la neurona $j$ con respecto a un peso particular $w_{ij}$ no es más que su entrada $x_i$ : $\frac{\partial z_j}{\partial w_{ij}} = x_i$.
- La derivada parcial de la salida $y_j$ de la neurona $j$ con respecto a su entrada neta $z_j$ no es más que la derivada de la función de activación de la neurona evaluada para su entrada neta $\frac{\partial y_i}{\partial z_j} = f'(z_j)$. Éste es el motivo por el que el algoritmo de entrenamiento de redes multicapa requieren que las funciones de activación sean derivables.

No se sabe realmente lo que debe hacer cada neurona oculta, por lo que no se puede determinar directamente cuál es el error que comete; sin embargo, se puede calcular cómo cambia el error cuando cambia su actividad; es decir, en lugar de utilizar la salida deseada para entrenar las neuronas ocultas se usa la derivada del error con respecto a sus niveles de actividad, que representa el factor que falta por calcular de la derivada del error con respecto a los niveles de actividad de las neuronas ocultas de la red: $\frac{\partial E}{\partial y_j}$ [22].

Un pequeño cambio en los pesos $w_{ij}$ de una neurona oculta $j$ generará un cambio en su salida $y_j$, que se puede aproximar como:

$$\Delta y_j \approx \sum_i \frac{\partial y_j}{\partial w_{ij}} \Delta w_{ij} = \nabla y_j \Delta w_j$$

donde $\nabla y_j$ es el gradiente o derivada parcial de la salida $y_j$ de la neurona $j$ con respecto a un peso particular $w_{ij}$.

De forma similar, si se considera el efecto que tiene un pequeño cambio en un peso particular $w_{ij}$ sobre el error cometido por la red, se puede aproximar como: [22]

$$\Delta E \approx \frac{\partial E}{\partial x_{ij}^c} \Delta x_{ij}^c$$

donde el superíndice $c$ se usa para indicar que la neurona $j$ está en la capa $c$ de la red neuronal multicapa, siendo $0$ la capa de entrada de la red y $C$ la capa de salida de la misma.

El cambio de peso ha de propagarse desde la neurona que está en la capa $c$ hasta la salida de la red, que es donde se puede evaluar directamente el error cometido. Así pues, la influencia del cambio del peso $w_{ij}$ sobre el error a través de un camino que conecta la neurona $j$ con la salida de la red vendrá dada por una expresión de la forma que se indica a continuación y se puede visualizar en la figura 12:

$$\Delta E \approx \frac{\partial E}{\partial y_m^C} \frac{\partial y_m^C}{\partial y_m^{C-1}} \frac{\partial y_m^{C-1}}{\partial y_n^{C-2}} \cdots \frac{\partial y_q^{c+1}}{\partial y_j^c} \frac{\partial y_j^c}{\partial w_{ij}^c} \Delta w_{ij}^c$$

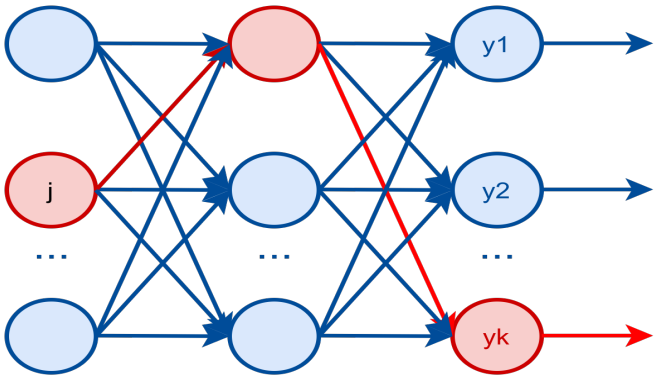


Figura 12: Uno de los caminos a través de los que un cambio en la actividad de la neurona $j$ afecta a la salida de la red.[22]

La expresión sólo recoge un camino particular desde la neurona $j$ hasta la salida de la red. Para obtener el efecto conjunto que tiene el cambio del peso $w_{ij}$ sobre el error global se debe considerar todos los caminos posibles que conectan la neurona $j$ con la salida de la red como se indica en la figura 13, de forma que

$$\Delta E \approx \sum_{mnp\ldots q} \frac{\partial E}{\partial y_m^C} \frac{\partial y_m^C}{\partial y_m^{C-1}} \frac{\partial y_m^{C-1}}{\partial y_n^{C-2}} \cdots \frac{\partial y_q^{c+1}}{\partial y_j^c} \frac{\partial y_j^c}{\partial w_{ij}^c} \Delta w_{ij}^c$$

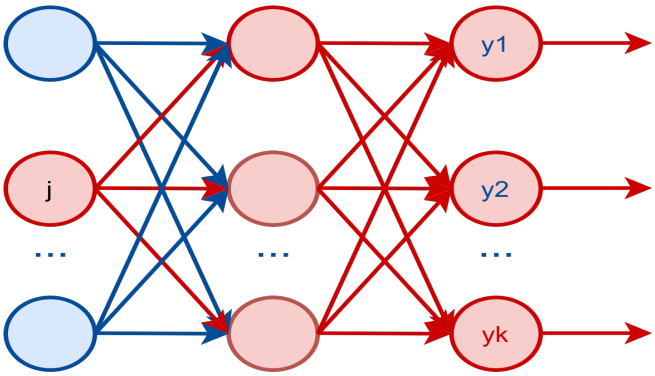


Figura 13: Todos los caminos a través de los que un cambio en la actividad de la neurona $j$ puede afectar a la salida de la red multicapa.[22]

Sustituyendo en la expresión inicial que relaciona $\Delta E$ con $\Delta w_{ij}^c$, se obtiene la derivada del error $E$ con respecto al peso $w_{ij}^c$ [22]:

$$\frac{\partial E}{\partial w_{ij}^c} = \sum_{mnp...q} \frac{\partial E}{\partial y_m^C} \frac{\partial y_m^C}{\partial y_m^{C-1}} \frac{\partial y_m^{C-1}}{\partial y_n^{C-2}} \cdots \frac{\partial y_q^{c+1}}{\partial y_j^c} \frac{\partial y_j^c}{\partial w_{ij}^c}$$

El objetivo es descubrir cómo modificar los parámetros de la red de forma que se minimice el error. En la ecuación anterior se indica como cambia el error $E$ conforme varia un parámetro individual $w_{ij}$ [22].El algoritmo de propagación de errores hacia atrás que se utiliza para ajustar los parámetros de una red multicapa calcula de forma eficiente la suma de las contribuciones de todos los caminos desde una neurona hasta la salida[22]; para ello aprovecha la topología de la red, organizada en una serie de capas con conexiones única y exclusivamente entre capas adyacentes.

**El algoritmo de propagación de errores hacia atrás.**

El término *backpropagation* se refiere únicamente al método mediante el cual se calcula el gradiente necesario para ajustar los parámetros de la red. Se pretende aplicar iterativamente un método de optimización basado en el gradiente descendente para ir ajustando los pesos de la red neuronal y conseguir entrenarla para que haga lo que se requiere[22]:

$$\Delta w = -\eta \nabla_w E$$

Donde $\nabla_w E$ es el gradiente de la función de error referente a los pesos, $\eta$ es la tasa de aprendizaje y $\Delta w$ es la modificación que se debe aplicar a los pesos de la neurona.

Para las neuronas de la capa de salida, el gradiente $\nabla_w E$ se calcula de la función de error, coste o pérdida encontrando la derivada con respecto a los pesos de las neuronas de salida para poder ajustar dichos pesos [22]. Se utiliza una función de error que cumpla dos propiedades:

- La función de error se puede especificar como un promedio de las funciones de coste asociadas al conjunto de datos de entrenamiento. Esto permitirá realizar una estimación instantánea del gradiente y calcular el gradiente para el conjunto de entrenamiento[22]:

$$E = \frac{1}{n} \sum_j E_j$$

$$\nabla_y E = \frac{1}{n} \sum_j \nabla E_j$$

- La función de coste asociada al conjunto de datos de entrenamiento se define en términos de la salida que se obtiene de la red $y_j$ y del valor que se desea obtener $t_j$. Por ejemplo, si se utiliza el error cuadrático[22]:

$$E_j = \frac{1}{2}(t_j - y_j)^2$$

$$\nabla E_j = -(t_j - y_j)$$

Para calcular el gradiente del error $\nabla E_j$, primero se tiene que calcular la salida de la red a partir del conjunto de entrenamiento. Por este motivo, el algoritmo de aprendizaje basado en *backpropagation* y gradiente descendente consta de dos fases[22]:

- Una fase de propagación hacia adelante, en la que se le suministra a la red un patrón de entrada y se calcula la salida de la red para dicho patrón.

- Una fase de propagación hacia atrás, de donde proviene el término *backpropagation*, en la que se evalúa el error cometido por la red y se propaga dicho error hacia atrás capa por capa, de forma que se pueda calcular eficientemente el gradiente del error para las neuronas ocultas de la red.

El error en la salida de una neurona puede deberse a los valores de sus pesos, a las entradas que se reciben de las capas anteriores de la red o a una combinación de ambos factores, motivo por el que se usará la entrada neta $z_j$ de la neurona como base para los cálculos, donde $z_j = w_j x_j$. Para ello, se define $\delta_j^c$ para una neurona $j$ de la capa $c$ como la derivada parcial del error con respecto a la entrada neta $z_j^c$ de la neurona[22].

$$\delta_j^c = \frac{\partial E}{\partial z_j^c}$$

El "delta" indica cómo influye una perturbación en la entrada neta de la neurona $\partial z_j^c$ en el error de la red neuronal. La perturbación en la entrada neta de la neurona, que se

debe a un cambio en alguno de sus pesos o en alguna de sus entradas, modificará la salida $y_j^c$ de la neurona.

Al pasar de las derivadas parciales con respecto a la salida $\frac{\partial E}{\partial y}$, a las derivadas parciales con respecto a la entrada neta $\frac{\partial E}{\partial z}$ se ha incorporado la función de activación $f$ de la neurona en el cálculo de los "deltas" que se reutilizará en el cálculo del gradiente del error[22]:

$$\delta_j^c = \frac{\partial E}{\partial z_j^c} = \frac{\partial E}{\partial y_j^c} f'(z_j^c)$$

donde $f'$ es la derivada de la función de activación de la neurona de salida.

Para calcular el gradiente del error con respecto a los parámetros de las neuronas ocultas de la red se tiene que considerar todos los caminos que conectan la salida de una neurona oculta con las salidas de la red neuronal. [22].

La derivada parcial del error con respecto al nivel de activación de una neurona oculta es de la forma:

$$\frac{\partial E}{\partial y_j^c} = \sum_{mnp\ldots q} \frac{\partial E}{\partial y_m^C} \frac{\partial y_m^C}{\partial y_m^{C-1}} \frac{\partial y_m^{C-1}}{\partial y_n^{C-2}} \cdots \frac{\partial y_q^{c+1}}{\partial y_j^c}$$

donde vemos que la derivada del error se calcula sumando las contribuciones de distintos caminos. Ahora bien, dado que las distintas neuronas de la capa $c$ comparten los caminos que van desde la capa $c+1$ hasta la salida de la red, esto permitirá calcular el gradiente del error para la capa $c+1$ y, a partir de ese gradiente, calcular el gradiente para las neuronas de la capa $c$ [22].

Sólo tenemos que observar cómo se relaciona el error de las neuronas de una capa oculta con el error de las neuronas de la siguiente capa. Para ello, se expresa la contribución al error de las neuronas de la capa $c$ en términos de la contribución al error de las neuronas de la capa $c+1$:

$$\frac{\partial E}{\partial y_j^c} = \sum_k \frac{\partial E}{\partial y_k^{c+1}} \frac{\partial y_k^{c+1}}{\partial y_j^c}$$

A continuación, reescribimos la expresión anterior en términos de las entradas netas de las neuronas en vez de emplear sus niveles de activación[22]:

$$\begin{aligned} \delta_j^c &= \frac{\partial E}{\partial z_j^c} \\ &= \frac{\partial E}{\partial y_j^c} f'(z_j^c) \\ &= \left(\sum_k \frac{\partial E}{\partial y_k^{c+1}} \frac{\partial y_k^{c+1}}{\partial y_j^c}\right) f'(z_j^c) \end{aligned}$$

$$= (\sum_k \delta_j^{c+1} \frac{\partial z_k^{c+1}}{\partial y_j^c}) f'(z_j^c)$$

La salida de la capa $c$ es la entrada de la capa $c+1$, $y^c = x^{c+1}$ , por lo que se puede sustituir $y^c$ por $x^{c+1}$ en la expresión anterior y calcular las derivadas parciales de la entrada neta $z_{c+1}$ con respecto a las entradas individuales $x_{c+1}$ [22]:

$$\delta_j = (\sum_k \delta_j^{c+1} \frac{\partial z_k^{c+1}}{\partial x_j^{c+1}}) f'(z_j^c)$$

$$\delta_j = (\sum_k \delta_j^{c+1} w_{jk}^{c+1}) f'(z_j^c)$$

Se ha logrado obtener una expresión que nos permite calcular los deltas de la capa $c$ a partir de los deltas de la capa $c+1$. En notación vectorial[22]:

$$\delta^c = ((W^{c+1})^T \delta^{c+1}) \odot f'(z^c)$$

donde $\odot$ es el producto elemento a elemento de dos vectores (conocido formalmente como producto Hadamard o producto Schur), de tal forma que los elementos del vector resultante de $v \odot w$ son $[v \odot w]_j = v_j w_j$

Usando el vector de "deltas" $\delta^c$ para las neuronas de la capa $c$ se puede obtener todos los valores que se necesita para completar el cálculo del gradiente del error con respecto a todos los parámetros de la red. El algoritmo de propagación de errores, *backpropagation* se puede visualizar en la figura 14 :

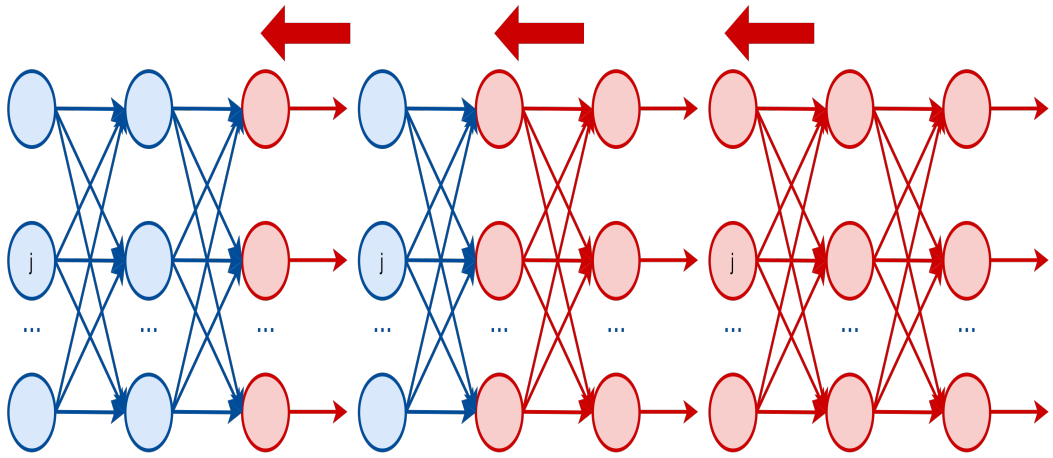


Figura 14: Cálculo del gradiente usando propagación de errores hacia atrás (backpropagation).[22]

A continuación se enumera los pasos que sigue el algoritmo de propagación de errores *backpropagation* [22] :

1. En primer lugar, para la capa de salida $C$, se calcula

$$\delta^C = \nabla_y E \odot f'(z^C)$$

2. A continuación, se calcula los "deltas" para las capas ocultas de la red, obteniendo los deltas de la capa $c$ a partir de la capa $c+1$:

$$\delta^c = ((W^{c+1})^T \delta^{c+1}) \odot f'(z^c)$$

   Combinando las dos expresiones anteriores se puede calcular el gradiente del error para todas las capas de la red multicapa empezando por la de salida. La primera expresión permite calcular el gradiente $\delta^c$ para las neuronas de la capa de salida; a partir de este resultado se utiliza la segunda expresión para calcular el gradiente $\delta^{c-1}$ de la capa $c-1$ a partir de $\delta^c$, $\delta^{c-2}$ a partir de $\delta^{c-1}$... y así sucesivamente hasta llegar al gradiente $\delta^1$ la primera capa de la red.

3. Una vez obtenido todos los "deltas" se puede calcular fácilmente cómo varía el error en función de los pesos de las neuronas. Teniendo en cuenta que la salida de la capa $c-1$ es la entrada de la capa $c(x^c = y^{c-1})$, la derivada del error con respecto a los pesos se la puede expresar como:

$$\frac{\partial E}{\partial w_{ij}^c} = \frac{\partial E}{\partial z_j^c} \frac{\partial z_j^c}{\partial w_{ij}^c} = \delta_j^c x_j^c = \delta_j^c y_j^{c-1}$$

   Si en la implementación se trata el sesgo $b_j$ como un parámetro separado del resto de los pesos de una neurona. El sesgo equivale a un peso asociado a una entrada fija $x_0 = 1$, por lo que

$$\frac{\partial E}{\partial b_j^c} = \delta_j^c$$

El algoritmo de propagación de errores proporciona un método eficiente para calcular las derivadas del error $\frac{\partial E}{\partial w_{ij}}$ para cada peso de una red multicapa. Prescindiendo de los superíndices que se utiliza para indicar la capa en la que se encuentra cada neurona, cada una de las capas de la red puede estimar el gradiente del error con respecto a su matriz de pesos como el producto exterior [*outer product*] de su vector de deltas $\delta$ y su vector de entradas $x$[22]:

$$\Delta_W E = x \otimes \delta$$

donde el producto exterior $x \otimes \delta$ de un vector $x$ de tamaño $n$ y un vector de tamaño m es una matriz $\Delta_W E$ de tamaño $n \times m$ en la que su elemento $(i, j)$ se define como $[\Delta_W E]_{ij} = x_i \delta_j$.

### 2.2.5. Procedimiento para el entrenamiento de una red neuronal.

A la hora de entrenar una red neuronal multicapa, se debe tomar decisiones con respecto tanto a los parámetros de diseño de la red como a los parámetros del algoritmo de entrenamiento utilizado. Colectivamente, estos parámetros reciben el nombre de hiperparámetros para distinguirlos de los propios parámetros de la red, los pesos de sus sinapsis y los sesgos de sus neuronas que se ajustan durante el proceso de entrenamiento de la red[22].

**Topología de la red.**

Una vez establecida la arquitectura general de la red, el diseñador debe decidir el número de capas de la red, sus patrones de interconexión y el tamaño de cada una de las capas (número de neuronas que forman parte de cada capa).

- Tamaño de la red.

En la práctica muchas veces se consiguen resultados aceptables utilizando un número pequeño de neuronas. Las redes neuronales reales no tienen que trabajar con datos arbitrarios, sino que han de tratar con los datos que se presentan en la vida real.

Dado un problema específico, determinar el número adecuado de neuronas ocultas no resulta sencillo. Habitualmente, se procura minimizar el número de neuronas utilizado para reducir la carga computacional que supone el entrenamiento y uso de neuronas adicionales, ya que cada neurona extra añade una carga computacional innecesaria.

Si el entrenamiento de la red no converge, posiblemente no se conseguirá el rendimiento deseado por lo que se deberá añadir más nodos ocultos a la red (o emplear un mejor conjunto de entrenamiento)[22].

- Profundidad de la red.

Una red neuronal con tres capas es suficiente para aproximar cualquier función siempre que se incluya un número suficiente de neuronas en la capa oculta y se disponga de un conjunto de datos de entrenamiento, sin embargo muchos investigadores se dieron cuenta de que entrenar una red con una capa oculta no siempre resultaba idóneo, con dos capas

ocultas una red se entrena mejor[22]. Las neuronas de la primera capa oculta son capaces de extraer características locales, las neuronas de la segunda capa oculta combinan esas características locales extraídas por las neuronas de la capa anterior y son capaces de identificar características globales.

**Funciones de activación.**

Para entrenar una red neuronal se utiliza el gradiente del error y en el cálculo de dicho gradiente interviene la derivada de la función de activación. Una función de activación debe ser derivable; sin embargo no es necesario que las funciones sean estrictamente derivables en todos los puntos. Suele bastar con que las funciones tengan definidas sus derivadas por la izquierda y por la derecha[22].

**Función de activación Sigmoidal**

Usualmente, interesa que la función de activación de una neurona sea, además de no lineal, estrictamente creciente, continua y derivable. Las funciones sigmoidales satisfacen todos esos requisitos lo que las hace especialmente útiles en redes neuronales que se entrenan usando *backpropagation*. Existen distintas funciones sigmoidales. Todas tienen en común su característica forma de '$S$'; algunas de ellas tienen propiedades matemáticas que las hacen especialmente interesantes para su uso en redes neuronales artificiales. Cuando existe una relación sencilla entre el valor de la función en un punto y el valor de su derivada en ese punto se puede aprovechar esa relación para reducir el coste computacional del entrenamiento de la red neuronal[22].

- Función logística.

$$y = f_{logistic}(z) = \sigma(z) = \frac{1}{1 + e^{-z}}$$

La función logística está representada en la figura 15 y su derivada en la figura 16

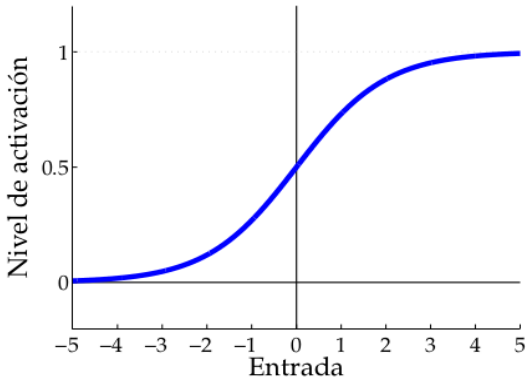


Figura 15: Función de activación sigmoidal: Función logística.[22]

La función logística es una sigmoide “binaria”, con rango [0,1], cuya derivada se calcula a continuación:

$$\frac{d\sigma(x)}{dz} = \frac{d}{dz}[\frac{1}{1+e^{-z}}]$$
$$= \frac{0(1+e^{z}) - 1(-e^{z})}{(1+e^{-z})^2}$$
$$= \frac{e^{-z}}{(1+e^{-z})^2}$$
$$= \frac{1}{1+e^{-z}}\frac{e^{-z}}{1+e^{-z}}$$

Ahora bien,

$$\frac{e^{-z}}{1+e^{-z}} = \frac{(1+e^{-z})-1}{1+e^{-z}} = \frac{1+e^{-z}}{1+e^{-z}} - \frac{1}{1+e^{-z}} = 1 - \sigma(z)$$

Por tanto:

$$\frac{d\sigma(z)}{dz} = \sigma(z)(1-\sigma(z))$$

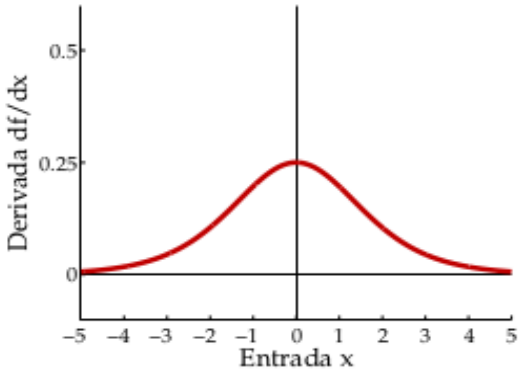


Figura 16: Derivada de la función de activación sigmoidal: Derivada de la función logística.[22]

La relación entre el valor de la función $\sigma(z)$ y su derivada permite calcular el valor de esta última con sólo dos operaciones aritméticas, una resta y una multiplicación sin necesidad de realizar llamadas a funciones trascendentes cuya evaluación es usualmente más costosa desde el punto de vista computacional[22].

- Tangente hiperbólica.

La tangente hiperbólica representada en la figura 17 se utiliza en trigonometría esférica y se define de la siguiente forma:

$$y = f_{tanh}(z) = tanh(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}} = \frac{1 - e^{-2z}}{1 + e^{-2z}}$$

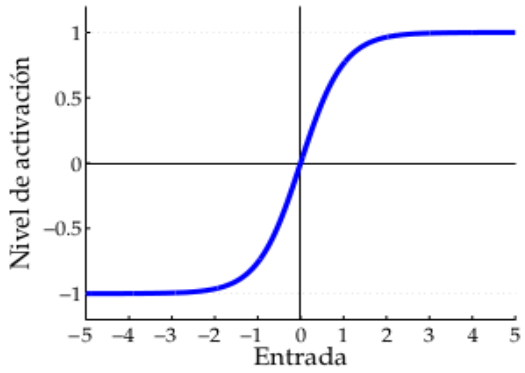


Figura 17: Función de activación sigmoidal bipolar: Tangente hiperbólica.[22]

La derivada de la tangente hiperbólica se representada en la figura 18.

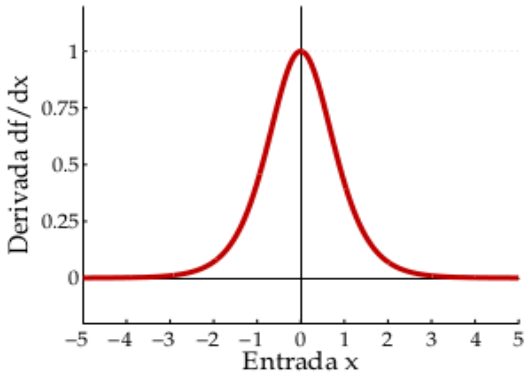


Figura 18: Derivada de la tangente hiperbólica.[22]

La tangente hiperbólica, según algunos autores, funciona mejor que la función logística al parecerse más a la función identidad para valores de $z$ cercanos a cero[22].

**El conjunto de entrenamiento.**

El rendimiento de una red neuronal depende del conjunto de datos de entrenamiento que se utilice para ajustar sus parámetros. Es fundamental que el conjunto de entrenamiento se parezca a los datos con los que realmente se encontrará la red en la práctica, una vez entrenada y puesta en marcha en un entorno real[22].

**Preprocesamiento de los datos de entrenamiento.**

Se recomienda normalizar los datos de entrada del conjunto de entrenamiento para poder comparar datos distintos. La normalización de los datos de entrada se puede realizar empleando *Z-Scores* (variables tipificadas): $Z = \frac{x-\mu}{\sigma}$, donde $x$ es el valor original de la variable, $\mu$ es la media y $\sigma$ la desviación del conjunto de entrenamiento. Cada variable del conjunto de entrenamiento pasa a tener media $0$ y varianza $1$. La normalización es una forma de compensar las diferencias que pueden existir entre las diferentes variables de

entrada[22].

Para facilitar el entrenamiento de la red, suele ser recomendable que el conjunto de entrenamiento cumpla las siguientes propiedades:

- La media de cada variable de entrada en el conjunto de entrenamiento debería ser cercana a cero.
- La escala de las variables de entrada debería ajustarse para que sus varianzas sean similares.
- Si es posible, las variables de entrada deberían estar sin correlación.

Si el rendimiento de la red neuronal sobre el conjunto de entrenamiento no es bueno se puede ampliar su capacidad añadiéndole más capas y neuronas ocultas, ajustar los hiperparámetros del algoritmo de entrenamiento o modificar la tasa de aprendizaje. Si al ampliar la capacidad de la red y ajustar sus hiperparámetros no se consigue que su rendimiento mejore, el problema puede que resida en la calidad de los datos de entrenamiento[22].

Si el rendimiento de la red neuronal es mucho peor en el conjunto de prueba que en el conjunto de entrenamiento, recopilar más datos puede ser la estrategia más adecuada.En el caso particular de las redes neuronales, añadir una pequeña fracción al número total de datos de entrenamiento puede no tener un impacto notable en el rendimiento del modelo entrenado. Normalmente se va modificando el tamaño del conjunto de entrenamiento usando una escala logarítmica.

**Tasa de aprendizaje.**

Una vez inicializados correctamente los parámetros de la red neuronal artificial, se debe ajustarlos utilizando un algoritmo de entrenamiento. Si se usa el gradiente descendente, la actualización de los pesos de la red se hará de acuerdo a una expresión de la forma[22]

$$\Delta w = -\eta \nabla E$$

Esta expresión se transforma en la fórmula de actualización de los pesos de la red

$$\Delta w_{ij} = -\eta x_i \delta_j$$

donde $\delta_j$ representa la derivada parcial del error con respecto a la entrada neta $z_j$ de la neurona $j$ de una capa con entradas $x_i$ . La magnitud del ajuste de los pesos dependerá de la magnitud del gradiente del error, de las entradas recibidas y de la tasa de aprendizaje del algoritmo $\eta$ [22]. Se elegirá un valor para la tasa de aprendizaje $\eta$ que haga que, en cada

iteración, los pesos cambien en una fracción de su valor actual. Suele ser útil comparar la magnitud de los gradientes del error con la magnitud de los parámetros que pretendemos ajustar.

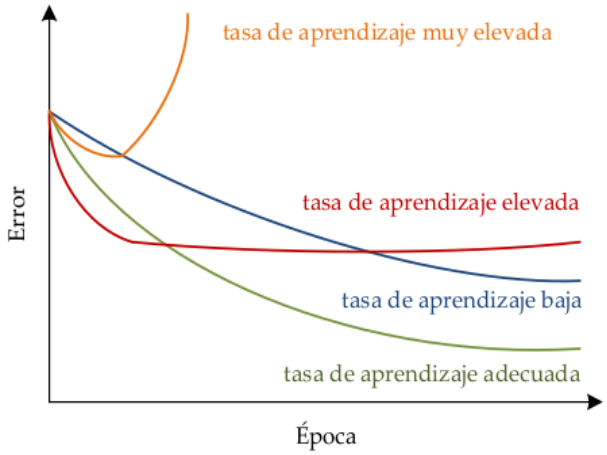


Figura 19: Representación gráfica del efecto de la tasa de aprendizaje sobre la convergencia del algoritmo de entrenamiento de una red neuronal multicapa.[22]

### 2.2.6. Control predictivo basado en el modelo (MPC)

El término Control Predictivo Basado en Modelo (MBPC o MPC) consiste en métodos de control amplios que hacen uso de un modelo de proceso para lograr alcanzar una señal de control mediante la minimización de una función objetivo. Las caracteristicas de un control predictivo son[23]:

- Utilización de un modelo matemático para predecir la salida del proceso en instantes de tiempo futuros (horizonte de predicción y control).
- Computo de una serie de control minimizando una función objetivo.
- Estrategia de retroceso para que en cada instante el horizonte se desplace hacia el futuro, lo que implica la aplicación de la primera señal de control de la secuencia calculada en cada paso.

El MPC presenta una serie de ventajas sobre otros métodos de control[23]:

- Fácil de comprender para personal con poco conocimiento sobre la teoría de control, debido a que los conceptos son muy intuitivos.
- Puede controlar una gran variedad de procesos, desde sistemas con una dinámica simple hasta los más complejos.

- El caso multivariable puede tratarse fácilmente.
- El controlador resultante es fácil de implementar.
- El tratamiento de restricciones es conceptualmente simple.

La técnica de los controladores de la familia MPC se indicada en la figura 20.

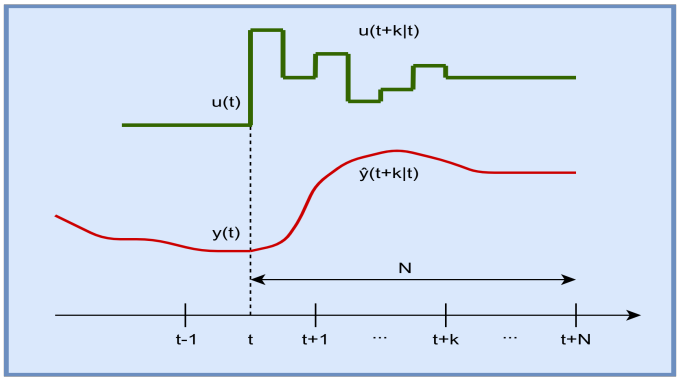


Figura 20: Técnicas utilizada por MPC.[23]

- Las salidas de un horizonte de predicción futuras para un $N$ se predicen en cada instante $t$. Las salidas pronosticadas $y(t+k|t)$ para $k = 1...N$ dependerá de los valores encontrados hasta el instante $t$ y también de las señales futuras de control $u(t+k|t), k = 0...N-1$, las mismas que son calculadas y enviadas al sistema[23].
- El grupo de señales futuras de control se calculan mejorando un criterio para conservar el proceso cerca de la trayectoria de referencia $w(t+k|t)$. Este criterio es una función cuadrática de los errores de la señal de salida y la trayectoria de referencia predicha, el esfuerzo de control se incorpora en la función objetivo.[23].
- La señal de control $u(t+k|t)$ se traslada al sistema mientras las siguientes señales de control calculadas son rechazadas.[23].

En la figura 21 se indica la estructura de la estrategia de control del MPC, para lo cual se utiliza un modelo para predecir las salidas futuras de la planta en función de valores predichos y en las acciones de control futuras óptimas.

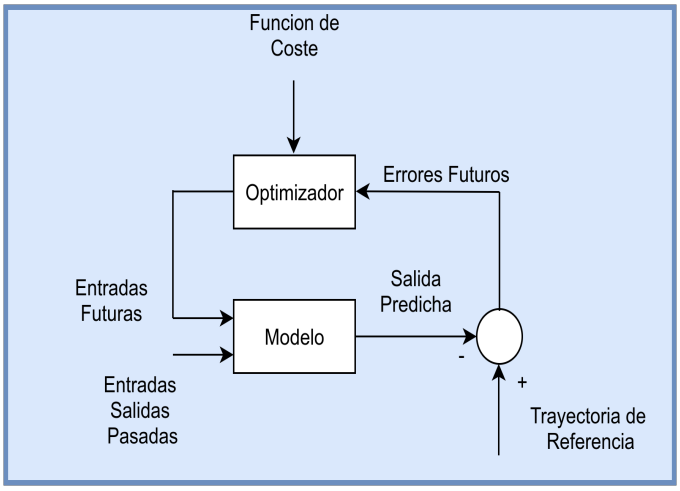


Figura 21: Estructura del MPC.[23]

El modelo del proceso es muy importante en el controlador, debe tener la capacidad de obtener la dinámica del proceso para predecir con precisión los resultados futuros. Uno de los modelos más utilizados en la industria y en la academia es el Modelo de Respuesta al Escalón, y se lo obtiene mediante la medición de la salida de la planta cuando al proceso se le aplica una entrada tipo escalón. El optimizador suministra acciones de control; si la función de coste es cuadrática su mínimo se obtiene como una función explícita, en caso de existir restricciones en la función de coste la solución se obtiene mediante algoritmos de optimización numéricos[23].

A continuación se describe los elementos del controlador basado en el modelo predictivo MPC:

- Modelo de predicción.

El modelo de predicción es la parte fundamental del MPC; un diseño completo debe tener todos los componentes para obtener el mejor modelo posible, quien se encargará de capturar la dinámica del proceso y también debe ser capaz de permitir que se calculen las predicciones. La utilización del modelo del proceso se determina calculando la salida predicha en instantes futuros $\hat{y}(t+k|t)$. Las estrategias del MPC usan varios modelos para indicar la relación entre las salidas y las entradas medibles[23].

- Modelo del proceso (Respuesta al escalón).

Este tipo de modelo es utilizado por el DMC, donde la señal de entrada o excitación es un escalón. Para sistemas estables, la respuesta truncada viene dada por:

$$y(t) = y_0 + \sum_{i=1}^{N} g_i \Delta u(t-i) \tag{16}$$

donde $g_i$ son los valores de salida muestreados para la entrada escalón y $\Delta u(t) = u(t) - u(t-1)$ como se indica en la figura 22. El valor de $y_0$ se puede tomar como $0$ debido a que el sistema parte del reposo[23]:

$$\hat{y}(t+k|t) = \sum_{i=1}^{N} g_i \Delta u(t+k-i|t) \tag{17}$$

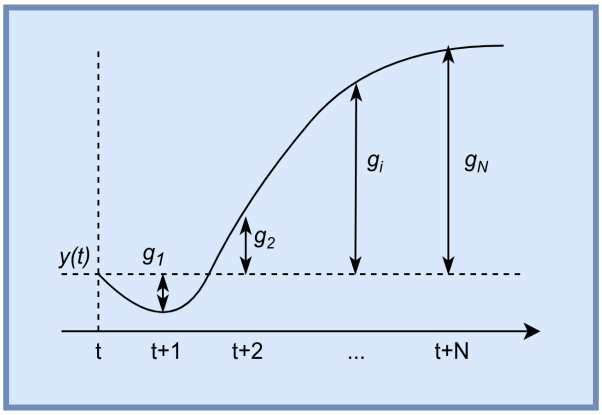


Figura 22: Respuesta escalonada.[23]

- Respuesta libre y forzada.

Una de las caracteristicas de los MPC es la utilización de la respuesta $libre$ $(u_f(t))$ y $forzada$ $(u_c(t))$. Es necesario indicar la secuencia de control a través de la suma de las dos señales:

$$u(t) = u_f(t) + u_c(t)$$

La señal $u_f(t)$ pertenece a las entradas pasadas y se mantiene constante e igual al último valor de la variable manipulada en instantes de tiempo futuros; es decir:

$$u_f(t-j) = u(t-j) \qquad \text{para} \qquad j = 1, 2, ...$$

$$u_f(t+j) = u(t-j) \qquad \text{para} \qquad j = 0, 1, 2, ...$$

La señal $u_c(t)$ se iguala a cero en el pasado y se iguala a los movimientos de control futuro; es decir:

$$u_c(t-j) = 0 \qquad \text{para} \qquad j = 1, 2, ...$$

$$u_c(t+j) = u(t+j) - u(t-1) \qquad \text{para} \qquad j = 0, 1, 2, ...$$

La predicción de la serie de salida se divide como se indica en la figura 23. La respuesta libre $(y_f(t))$ pertenece al pronostico de la salida cuando la variable manipulada del proceso es igualada $u_f(t)$ y la respuesta forzada $(y_c(t))$ pertenece a la pronostico de la salida del proceso cuando la secuencia de control es igual a $u_c(t)$[23].

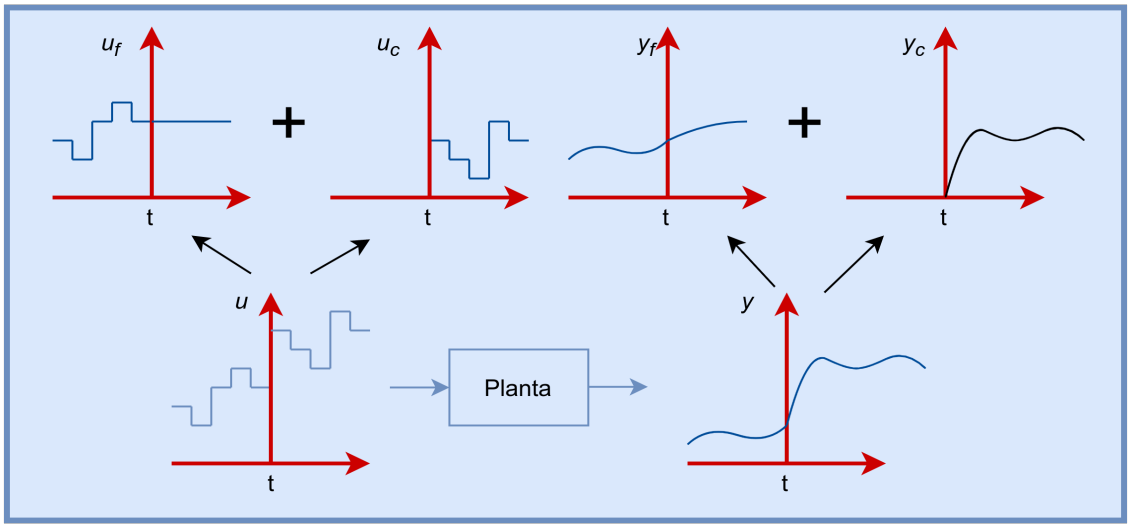


Figura 23: Respuesta libre y forzada.[23]

- Función objetivo.

Los diferentes algoritmos de control predictivo plantean varias funciones de coste para calcular la ley de control. El objetivo es mantener la señal de referencia determinada $(w)$ de la salida futura $(y)$ en un determinado horizonte y al mismo tiempo el esfuerzo de control $(\Delta u)$ necesario para hacerlo debe ser penalizado. La expresión general para tal función objetivo esta dada por:

$$J(N_1, N_2, N_u) = \sum_{j=N_1}^{N_2} \delta(j)[\hat{y}(t+j|t) - w(t+j)]^2 + \sum_{j=1}^{N_u} \lambda(j)[\Delta u(t+j-i)]^2 \quad (18)$$

La función de coste se considera:

- $N_1$ y $N_2$ representan los horizontes de coste mínimo y máximo y $N_u$ representa el horizonte de control. Al escoger un valor alto de $N_1$ los errores en los primeros instantes no tinen importancia; lo que llevara a una respuesta fluida del proceso. Las constantes $\delta(j)$ y $\lambda(j)$ consideran el comportamiento futuro[23].

- Trayectoria de referencia: Si la evolución futura de la referencia se conoce a priori, la reacción del sistema puede ser antes de que se haya realizado el cambio de manera efectiva, lo que evitaría los efectos del retraso en la respuesta del proceso. En la minimización de la ecuación 18, gran parte de los métodos por lo general utilizan una trayectoria de referencia $W(t+k)$ que no coincide con la referencia real[23].

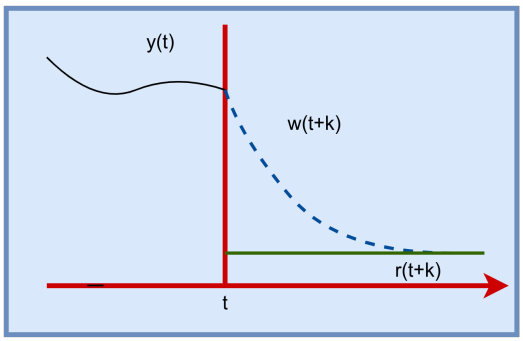


Figura 24: Trayectoria de referencia.[23]

- Obtención de la ley de control.

Para alcanzar resultados $u(t+k|t)$ es necesario minimizar $J$ de la formula (18). Para ello, los valores de las salidas predichas $\hat{y}(t+k|t)$ se calculan en función de los valores anteriores y de señales futuras de control, alcanzando una expresión en la cual la minimización conduce a los valores deseados. Se puede obtener una solución analítica para el criterio cuadrático si el modelo es lineal y no hay restricciones, de lo contrario, debería utilizarse un método iterativo de optimización[23].

**Control de Matriz Dinámica.**

- Predicción.

El modelo del proceso empleado para la predicción es la respuesta al escalón de la planta [23].

Como modelo de respuesta al escalón se emplea la ecuación:

$$y(t) = \sum_{i=1}^{\infty} g_i \Delta u(t-i)$$

los valores de predición en el horizonte serán:

$$\hat{y}(t+k|t) = \sum_{i=1}^{\infty} g_i \Delta u(t+k-i) + \hat{n}(t+k|t) =$$

$$\hat{y}(t+k|t) = \sum_{i=1}^{k} g_i \Delta u(t+k-i) + \sum_{i=k+1}^{\infty} g_i \Delta u(t+k-i) + \hat{n}(t+k|t)$$

Las perturbaciones se consideran constantes, es decir $\hat{n}(t+k|t) = \hat{n}(t|t) = y_m(t) - \hat{y}(t|t)$. Entonces:

$$\hat{y}(t+k|t) = \sum_{i=1}^{k} g_i \Delta u(t+k-i) + \sum_{i=k+1}^{\infty} g_i \Delta u(t+k-i) + y_m(t) - \sum_{i=1}^{\infty} g_i \Delta u(t-i)$$

$$\hat{y}(t+k|t) = \sum_{i=1}^{k} g_i \Delta u(t+k-i) + f(t+k)$$

donde $f(t+k)$ es la respuesta libre del sistema y es la parte de la respuesta que no depende de las futuras acciones de control, viene dada por:

$$f(t+k) = y_m(t) + \sum_{i=1}^{\infty} (g_{k+i} - g_i) \Delta u(t-i) \tag{19}$$

Si el sistema es asintóticamente estable los coeficientes $g_i$ de la respuesta al escalón tienden a un valor constante después de $N$ periodos de muestreo, por lo que se puede considerar que

$$g_{k+i} - g_i \approx 0, \qquad i > N$$

y por lo tanto la respuesta libre se puede calcular como:

$$f(t+k) = y_m(t) + \sum_{i=1}^{N} (g_{k+i} - g_i) \Delta u(t-i)$$

Los pronósticos se pueden calcular en el horizonte de predicción $(k = 1, ..., p)$, considerando $m$ acciones de control.

$$\hat{y}(t+1|t) = g_1 \Delta u(t) + f(t+1)$$

$$\hat{y}(t+2|t) = g_2 \Delta u(t) + g_1 \Delta u(t+1) + f(t+2)$$

$$\vdots$$

$$\hat{y}(t+p|t) = \sum_{i=p-m+1}^{p} g_i \Delta u(t+p-i) + f(t+p)$$

Se muestra la *matriz dinámica* del sistema $G$ como:

$$\mathbf{G} = \begin{bmatrix} g_1 & 0 & \cdots & 0 \\ g_2 & g_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_m & g_{m-1} & \cdots & g_1 \\ \vdots & \vdots & \ddots & \vdots \\ g_p & g_{p-1} & \cdots & g_{p-m+1} \end{bmatrix}$$

se puede escribir en forma matricial como:

$$\hat{\mathbf{y}} = \mathbf{Gu + f} \tag{20}$$

La matriz dinámica **G** está formado por $p$ (horizonte de predicción) filas y $m$ (horizonte de control) columnas de la respuesta al escalón del sistema, $\hat{\mathbf{y}}$ es un vector de p-dimensiones que contiene las predicciones del sistema a lo largo del horizonte, **u** representa el vector de m-dimensiones de incrementos de control y **f** de p-dimensiones es el vector de respuesta libre. Esta es la expresión que se encarga de relacionar las salidas futuras utilizando los incrementos de control, por lo que se utilizará para calcular la acción necesaria para alcanzar una conducta específica del sistema[23].

- Algoritmo de control.

En esta sección se describe el algoritmo de control a partir del caso más simple de un sistema monovariable sin restricciones.

El objetivo de un controlador DMC es conducir la salida(output) lo más cerca posible del punto de ajuste(setpoint) en un sentido de mínimos cuadrados, con la posibilidad de incluir un término de penalización en los movimientos de entrada. Por lo tanto, la salida(output) y el punto de ajuste(setpoint) se seleccionan para minimizar un objetivo cuadrático que puede considerar solo la minimización de errores futuros[23]:

$$J = \sum_{j=1}^{P} [\hat{y}(t+j|t) - w(t+j)]^2$$

o puede incluir el esfuerzo de control

$$J = \sum_{j=1}^{P} \delta[\hat{y}(t+j|t) - w(t+j)]^2 + \sum_{j=1}^{N} \lambda[\Delta u(t+j-i)]^2$$

Si no hay restricciones, la respuesta minimizada de la función de coste es $J = \mathbf{ee}^T + \lambda \mathbf{uu}^T$, donde **e** es el vector de futuros errores a lo largo del horizonte de predicción y **u** es el vector compuesto por los incrementos de control futuros $\Delta u(t), ..., \Delta u(t+m)$, se puede obtener analíticamente calculando la derivada de J e igualándola a 0, obteniendo el resultado general[23]:

$$\Delta \mathbf{u} = (\mathbf{G}^T \delta \mathbf{I} \mathbf{G} + \lambda \mathbf{I})^{-1} \mathbf{G}^T \delta \mathbf{I} (\mathbf{w} - \mathbf{f})$$

En todas las estrategias predictivas, solo el primer elemento del vector $\mathbf{u}(\Delta u(t))$ realmente se envía a la planta; no es aconsejable efectuar toda la secuencia en los siguientes $m$ intervalos, debido a que es imposible estimar las perturbaciones de manera exacta lo que imposibilita anticiparse a las perturbaciones que excitan que la salida real difiera de las predicciones que se emplean para la sucesión futura de acciones de control [23].

# CAPÍTULO III
# MARCO METODOLÓGICO

## 3.1. Ubicación.

El controlador predictivo de matriz dinámica con redes neuronales se lo implementó en un proceso de nivel de la planta FESTO MPS PA Compact Workstation que se encuentra ubicada en los laboratorios de la Facultad de Ingeniería en Sistemas Electrónica e Industrial de la Universidad Técnica de Ambato.

## 3.2. Equipos y materiales.

Software:

- GNU Octave.
- Arduino IDE.
- Mathematica.
- Rstudio.

Hardware:

- Planta FESTO MPS PA Compact Workstation.
- Materiales electrónicos (Acondicionamiento de sensor y salida PWM).
- Laptop i5.
- Microcontrolador Arduino Uno.

## 3.3. Tipo de investigación.

### 3.3.1. Investigación bibliográfica.

La investigación es bibliográfica ya que se realizó una revisión de material bibliográfico existente con respecto al tema en estudio en fuentes como libros, revistas, artículos científicos, tesis doctorales y en la web. La investigación bibliográfica es uno de los principales pasos para cualquier investigación e incluye la selección de fuentes de información.

### 3.3.2. Investigación aplicada.

La investigación es de tipo aplicativa ya que está centrada en encontrar mecanismos o estrategias que permitan lograr el objetivo concreto,es este caso, implementar un sistema de control predictivo de matriz dinámica con redes neuronales. Por consiguiente, el ámbito al que se aplica es muy específico y bien delimitado, ya que no se trata de explicar una amplia variedad de situaciones, sino que más bien se intenta abordar un problema específico.

## 3.4. Prueba de hipótesis - Pregunta científica - Idea a defender.

Es posible implementar un controlador predictivo de matriz dinámica con redes neuronales en un sistema de nivel de líquido en la planta FESTO MPS PA Compact Workstation perteneciente a los laboratorios de la Facultad de Ingeniería en Sistemas de la UTA.

## 3.5. Población o muestra.

Debido a que el tema planteado es investigativo no se requiere de población y muestra.

## 3.6. Recolección de información

La recolección de información para la investigación planteada se lo realizó a través de: artículos científicos, libros, revistas, investigaciones y tesis doctorales, mismos que dotarón de información relevante para el desarrollo de la investigación, además, se recolectaron 7912 datos experimentales aplicando diferentes valores de setpoint a la planta de nivel planta FESTO MPS PA Compact Workstation, utilizando 5448 datos para el entrenamiento y 2464 datos para el testeo de la red neuronal equivalentes al 69 % y 31 % del total de datos, respectivamente; de igual manera se utilizaron 702 datos para la identificación de la función de transferencia del sistema de la planta antes mencionada.

## 3.7. Procesamiento de la información y análisis estadístico.

Utilizando el software OCTAVE se logró:

- Capturar 7912 datos de forma conjunta del sensor y del setpoint de la planta de nivel, los cuales fueron divididos de la siguiente manera: 5448 datos para entrenamiento y 2464 datos para testeo equivalentes al 69 % y 31 % respectivamente.

- Identificar la dinámica del sistema de nivel de líquido mediante el entrenamiento de la red neuronal.
- Implementar el controlador predictivo de matriz dinámica con identificación de la dinámica de la planta utilizando redes neuronales.

Los coeficientes de la ecuación de respuesta del sistema en el dominio del tiempo ($y(t) = A * K(1 - e^{-\frac{t}{\tau}}) = 4{,}1808965 * 0{,}6718(1 - e^{-\frac{t}{44{,}05}})$) encontrados fueron $K = 0{,}6718$ y $\tau = 44{,}05$; a estos valores se aplicaron estadísticos de prueba para aceptar o rechazar la hipótesis nula;los coeficientes obtenidos fueron menores a 0,05 por lo tanto se rechaza la hipótesis nula y se acepta la hipótesis alternativa la cual sugiere que los cambios en el predictor $t$ están asociados con cambios en la respuesta $y(t)$.

## 3.8. Resultados alcanzados.

Se encontró la función de transferencia del sistema de nivel mediante una regresión exponencial con la cual se formó la matriz dinámica de predicción **G**; se utilizaron 702 datos, los cuales fueron recolectados del experimento de entrada y respuesta al escalón donde a partir de los 250 segundos se logra alcanzar la estabilidad del sistema de escalón unitario.

Se encontró la dinámica del sistema del nivel de la planta mediante el entrenamiento de una red neuronal feed-forward con 5448 datos recolectados de un experimento de multiples entradas escalón al sistema. De igual manera se realizó el testeo de la red neuronal con 2464 datos verificando así la correcta identificación de la dinámica del sistema de nivel. Finalmente se realizó el Controlador Predictivo de Matriz Dinámica (DMC) con Redes Neuronales, el cual fue implementado en el proceso de nivel de la planta FESTO MPS PA Compact Workstation y también se realizó simulaciones para diferentes setpoints deseados.

# CAPÍTULO IV
# RESULTADOS Y DISCUSIÓN

## 4.1. Análisis de resultados.

### 4.1.1. Desarrollo de la propuesta.

Durante los últimos 30 años la teoría de control predictivo de modelos (MPC) se han desarrollado rápidamente. Sin embargo, frente a los crecientes requisitos sobre el control óptimo restringido que surge del rápido desarrollo de la economía y la sociedad han demostrado que la teoría del MPC aún se enfrenta a grandes desafíos. En el presente trabajo de investigación se realizó la implementacion del controlador predictivo basado en el modelo de matriz dinámica combinado con redes neuronales aplicado a un proceso de nivel de líquido de la planta FESTO MPS PA Compact Workstation.

La planta FESTO MPS PA Compact Workstation está formada de 4 procesos industriales a baja escala en lazo cerrado. Cada uno de los procesos se pueden manejar individualmente, los cuales son: proceso de nivel de tanque, proceso de caudal, proceso de presión y proceso de temperatura.

La implementación del sistema de control fue realizado en la planta FESTO MPS PA Compact Workstation de la Facultad de Ingeniería en Sistemas de la Universidad Técnica de Ambato, la cual está representada en la figura 25.

La estructura de los sensores y actuadores de la planta Festo permite la implementación de controladores continuos y discontinuos [24]. Los sensores admiten la manipulación de sus señales eléctricas análogas basadas en normas industriales Namur NE 43 HART de valores de corriente entre 4 a 20mA y manipulación de actuadores (bomba) mediante el PWM de un microcontrolador.

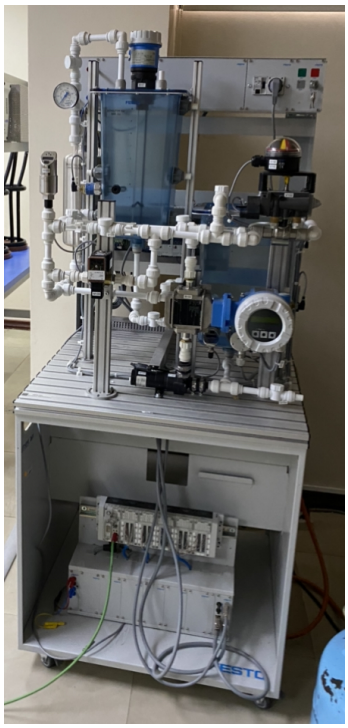

Figura 25: Planta FESTO MPS PA de la Facultad de Ingenieria en Sistemas

Los componentes de la célula FESTO® MPS-PA Compact Workstation utilizadas para la aplicación se indican en la figura 26.

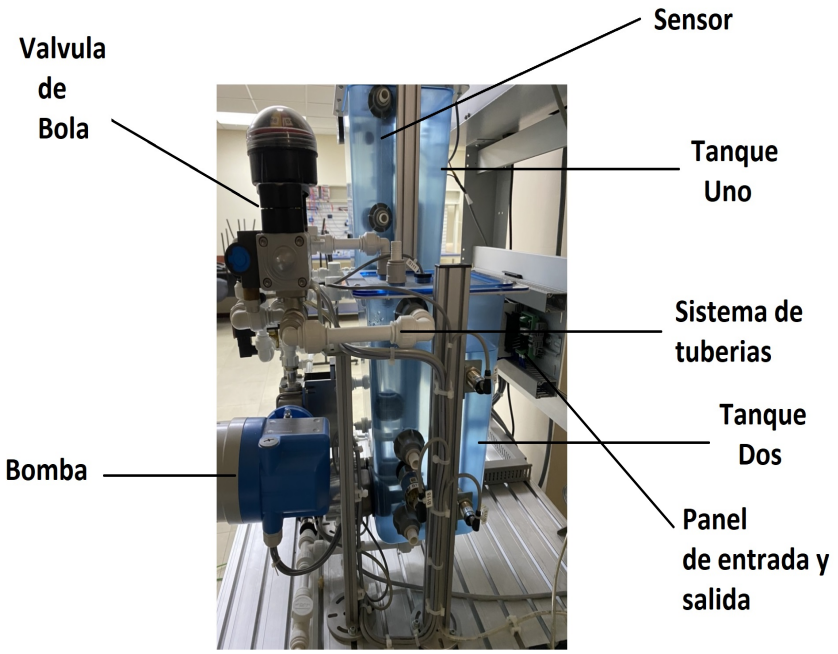


Figura 26: Planta FESTO MPS PA de la Facultad de Ingenieria en Sistemas

Fuente: Elaborado por la autora.

La estructura de los sensores y actuadores de la planta Festo permite la implementación de controladores continuos y discontinuos [24]. Los sensores admiten la manipulación de sus señales eléctricas análogas basadas en normas industriales Namur NE 43 HART de valores de corriente entre 4 a 20mA y manipulación de actuadores (bomba) mediante el PWM de un microcontrolador.

### 4.1.2. Diagrama PID del proceso de nivel.

El diagrama PID del proceso de nivel de líquido de la Planta FESTO MPS PA Compact Workstation está representado en la figura 27.

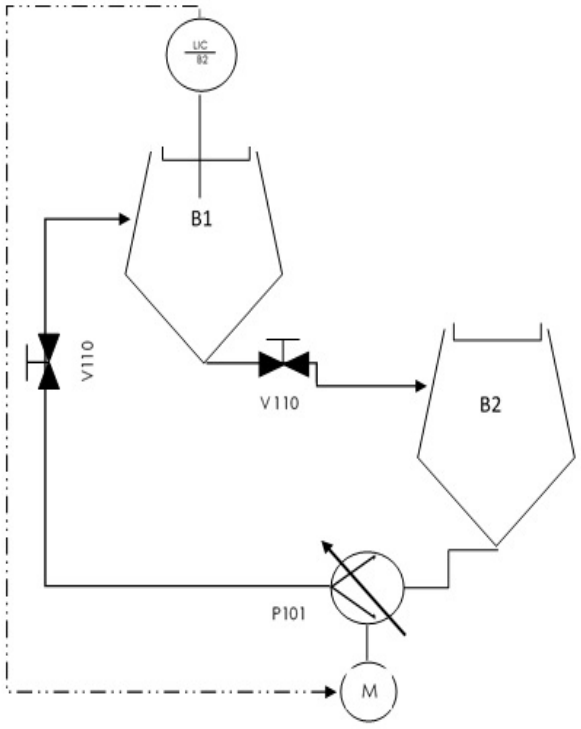


Figura 27: Diagrama PID del proceso de nivel. [24]

El actuador del sistema para este caso de estudio es una bomba P101, la cual proporciona el líquido desde un contenedor (tanque) inferior de almacenamiento B2, a otro contenedor (tanque) superior B1 mediante un sistema interconectado de tuberías. El nivel del líquido dentro del depósito superior B1 es medido mediante un sensor analógico ultrasónico B2 el cual se encuentra en el punto de medición LIC B2. El valor del nivel del líquido debe mantenerse en un valor deseado incluso cuando existan ciertos tipos de perturbaciones externas al sistema, éstas perturbaciones se pueden dar manualmente al abrir o cerrar la válvula V110.

### 4.1.3. Modelo Matemático de la planta de nivel.

Para identificar el modelo matemático representado como función de transferencia de la planta de nivel se identificaron los parámetros $K$ y $\tau$ pertenecientes a las ecuaciones (3) y (5). El método utilizado para identificar los parámetros $K$ y $\tau$ consistió en realizar un experimento de entrada escalón y capturar los datos para un posterior análisis. En la figura 28 se indica los componentes del experimento.

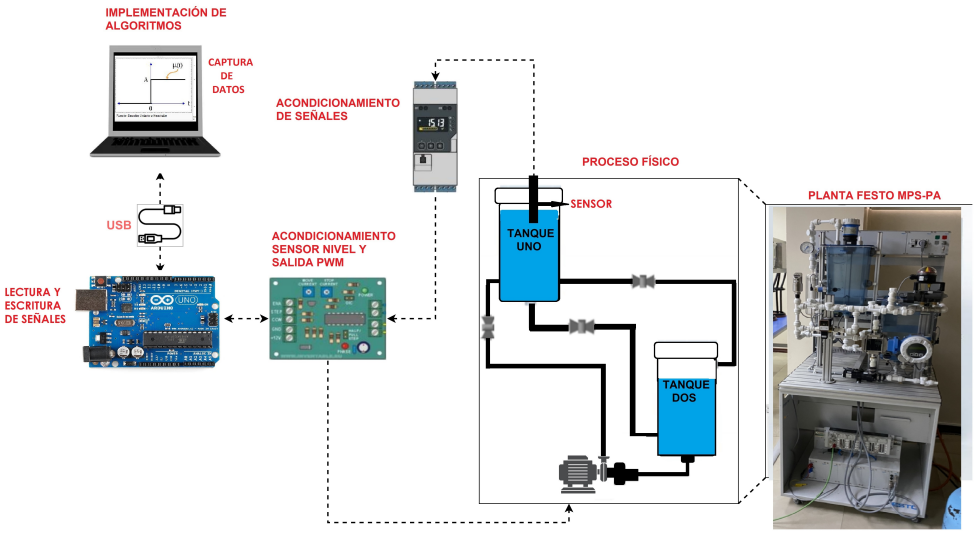


Figura 28: Componentes del experimento respuesta al escalón.

Fuente: Elaborado por la autora.

- En el computador (laptop i5) se ejecuta el código fuente en el software OCTAVE para capturar y guardar los valores del escalón y la respuesta al escalón obtenida por el sensor.
- La tarjeta Arduino-Uno se utiliza como puerta de comunicación entre el computador (laptop i5) y la tarjeta de acondicionamiento de señales.
- La tarjeta de acondicionamiento de señales adapta los rangos de voltaje del sensor estandarizados para conectarlos al Arduino-Uno; además, contiene el circuito de potencia para la manipulación de la bomba mediante el PWM.
- El transmisor de proceso con unidad de control RMA42 transforma la señal de corriente 4-20mA del sensor ultrasónico en una señal de voltaje estándar de 0 a 10V.

- En el sistema de nivel de la planta Festo MPS-PA Compact Workstation es de donde se captura los datos para la implementación del controlador.

El transmisor de proceso con unidad de control RMA42 convierte la señal de corriente de 4-20 mA del sensor de ultrasonido de nivel a una señal de voltaje de 0-10 V, razón por la cual es necesario transformar esta señal a un rango de 0-5 V que es el valor aceptado por la entrada analógica del Arduino-Uno, para lo cual se aplicó un circuito divisor de voltaje.

Finalmente se manipula una bomba de agua de 24 V mediante una señal PWM controlada desde el microcontrolador Arduino-Uno, los pines PWM de la tarjeta suministran un voltaje de 5 V a 50mA máximo a la salida, el cual es un valor no suficiente para operar la bomba, por lo tanto, se requiere implementar un circuito adicional de potencia. En la figura 29 se representa el circuito de potencia diseñado.

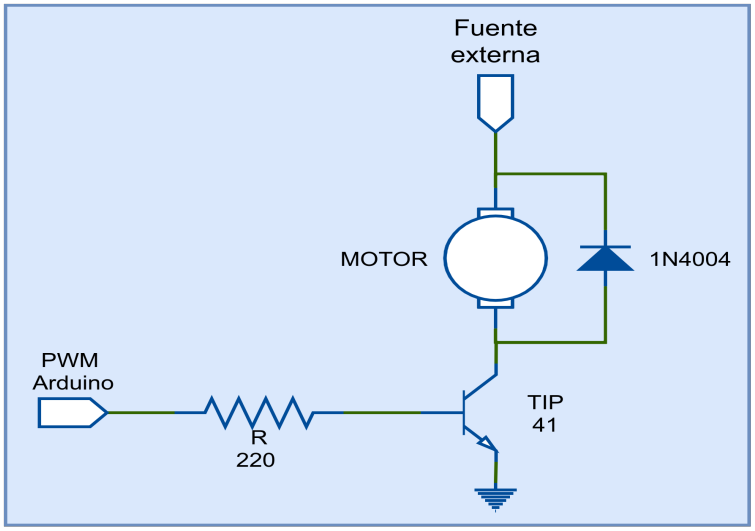


Figura 29: Circuito de potencia para el control de la bomba.
Fuente: Elaborado por la autora.

El experimento realizado consistió en aplicar una entrada escalón al sistema:

$$r(t) = Au(t)$$

Donde el valor de $A = 4{,}1808965$, el tiempo de muestreo fue de $0{,}4$ segundos y finalmente se espero que el sistema se estabilice en un valor determinado. Los datos del experimento fueron obtenidos con la ayuda de un microcontrolador Arduino-Uno y el software de programación Octave, éstos datos fueron guardados para un posterior análisis del sistema.

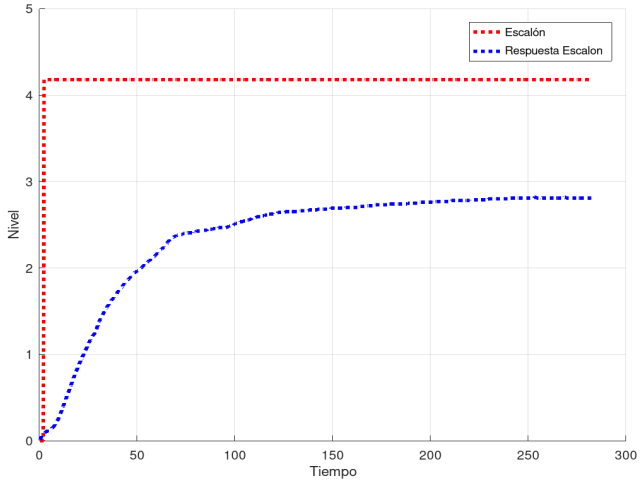


Figura 30: Respuesta del sistema al escalón unitario.

Fuente: Elaborado por la autora.

En la figura 30 se observa que el sistema alcanza su estado estable a partir de $250$ segundos, los datos capturados fueron $702$. La variable tiempo ($T$) se representa en el eje $x$ y las señales escalón y respuesta al escalón ($Nivel$) en el eje $y$, en la tabla 1 se representan dichos datos.

Tabla 1: Resultados experimentales. Tiempo(T)(segundos); Nivel(N) (litros)

| T | N | T | N | T | N | T | N | T | N | T | N |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0,000 | 15,6 | 0,609 | 31,2 | 1,409 | 46,8 | 1,899 | 62,4 | 2,222 | 78 | 2,407 |
| 0,4 | 0,012 | 16 | 0,633 | 31,6 | 1,419 | 47,2 | 1,908 | 62,8 | 2,222 | 78,4 | 2,417 |
| 0,8 | 0,03 | 16,4 | 0,660 | 32 | 1,439 | 47,6 | 1,918 | 63,2 | 2,231 | 78,8 | 2,417 |
| 1,2 | 0,049 | 16,8 | 0,684 | 32,4 | 1,458 | 48 | 1,928 | 63,6 | 2,241 | 79,2 | 2,417 |
| 1,6 | 0,064 | 17,2 | 0,709 | 32,8 | 1,478 | 48,4 | 1,938 | 64 | 2,251 | 79,6 | 2,417 |
| 2 | 0,075 | 17,6 | 0,733 | 33,2 | 1,497 | 48,8 | 1,948 | 64,4 | ,261 | 80 | 2,417 |
| 2,4 | 0,085 | 18 | 0,756 | 33,6 | 1,507 | 49,2 | 1,948 | 64,8 | 2,270 | 80,4 | 2,427 |
| 2,8 | 0,094 | 18,4 | 0,779 | 34 | 1,527 | 49,6 | 1,957 | 65,2 | 2,280 | 80,8 | 2,427 |
| 3,2 | 0,101 | 18,8 | 0,803 | 34,4 | 1,536 | 50 | 1,967 | 65,6 | 2,300 | 81,2 | 2,427 |
| 3,6 | 0,107 | 19,2 | 0,823 | 34,8 | 1,556 | 50,4 | 1,967 | 66 | 2,310 | 81,6 | 2,427 |
| 4 | 0,111 | 19,6 | 0,847 | 35,2 | 1,566 | 50,8 | 1,977 | 66,4 | 2,319 | 82 | 2,427 |
| 4,4 | 0,115 | 20 | 0,869 | 35,6 | 1,585 | 51,2 | 1,987 | 66,8 | 2,329 | 82,4 | 2,427 |
| 4,8 | 0,118 | 20,4 | 0,890 | 36 | 1,595 | 51,6 | 1,996 | 67,2 | 2,339 | 82,8 | 2,437 |
| 5,2 | 0,124 | 20,8 | 0,910 | 36,4 | 1,605 | 52 | 1,996 | 67,6 | 2,349 | 83,2 | 2,437 |
| 5,6 | 0,130 | 21,2 | 0,931 | 36,8 | 1,625 | 52,4 | 2,006 | 68 | 2,349 | 83,6 | 2,437 |
| 6 | 0,138 | 21,6 | 0,951 | 37,2 | 1,634 | 52,8 | 2,016 | 68,4 | 2,359 | 84 | 2,437 |
| 6,4 | 0,144 | 22 | 0,972 | 37,6 | 1,644 | 53,2 | 2,026 | 68,8 | 2,359 | 84,4 | 2,437 |
| 6,8 | 0,151 | 22,4 | 0,988 | 38 | 1,654 | 53,6 | 2,036 | 69. | 2,368 | 84,8 | 2,437 |
| 7,2 | 0,160 | 22,8 | 1,008 | 38,4 | 1,673 | 54 | 2,045 | 69,6 | 2,368 | 85,2 | 2,437 |
| 7,6 | 0,172 | 23,2 | 1,028 | 38,8 | 1,683 | 54,4 | 2,045 | 70 | 2,368 | 85,6 | 2,437 |
| 8 | 0,186 | 23. | 1,047 | 39,2 | 1,693 | 54,8 | 2,055 | 70,4 | 2,378 | 86 | 2,447 |
| 8,4 | 0,200 | 24 | 1,067 | 39,6 | 1,703 | 55,2 | 2,065 | 70,8 | 2,378 | 86,4 | 2,447 |
| 8,8 | 0,212 | 24,4 | 1,086 | 40 | 1,722 | 55,6 | 2,075 | 71,2 | 2,378 | 86,8 | 2,447 |
| 9,2 | 0,230 | 24,8 | 1,106 | 40,4 | 1,732 | 56 | 2,085 | 71,6 | 2,388 | 7,2 | 2,447 |
| 9,6 | 0,250 | 25,2 | 1,125 | 40,8 | 1,742 | 56,4 | 2,085 | 72 | 2,388 | 87,6 | 2,447 |
| 10 | 0,270 | 25,6 | 1,145 | 41,2 | 1,752 | 56,8 | 2,094 | 72,4 | 2,388 | 88 | 2,447 |
| 10,4 | 0,292 | 26 | 1,165 | 41,6 | 1,762 | 57,2 | 2,104 | 72,8 | 2,388 | 88,4 | 2,447 |
| 10,8 | 0,315 | 26,4 | 1,184 | 42 | 1,771 | 57,6 | 2,114 | 73,2 | 2,388 | 88,8 | 2,456 |
| 11,2 | 0,339 | 26,8 | 1,194 | 42,4 | 1,781 | 58 | 2,114 | 73,6 | 2,398 | 89,2 | 2,456 |
| 11,6 | 0,363 | 27,2 | 1,214 | 42,8 | 1,801 | 58,4 | 2,124 | 74 | 2,398 | 89,6 | 2,456 |
| 12 | 0,388 | 27,6 | 1,223 | 43,2 | 1,810 | 58,8 | 2,133 | 74,4 | 2,398 | 90 | 2,456 |
| 12,4 | 0,412 | 28 | 1,233 | 43,6 | 1,820 | 59,2 | 2,143 | 74,8 | 2,398 | 90,4 | 2,456 |
| 12,8 | 0,436 | 28,4 | 1,253 | 44 | 1,830 | 59,6 | 2,153 | 75,2 | 2,398 | 90,8 | 2,456 |
| 13,2 | 0,460 | 28,8 | 1,272 | 44,4 | 1,840 | 60 | 2,163 | 75,6 | 2,398 | 91,2 | 2,456 |
| 13,6 | 0,484 | 29. | 1,292 | 44,8 | 1,850 | 60,4 | 2,173 | 76 | 2,407 | 91,6 | 2,466 |
| 14 | 0,509 | 29,6 | 1,321 | 45,2 | 1,859 | 60,8 | 2,182 | 76,4 | 2,407 | 92 | 2,466 |
| 14,4 | 0,533 | 30 | 1,341 | 45,6 | 1,869 | 61,2 | 2,192 | 76,8 | 2,407 | 92,4 | 2,466 |
| 14,8 | 0,557 | 30,4 | 1,370 | 46 | 1,879 | 61,6 | 2,202 | 77,2 | 2,407 | 92,8 | 2,466 |
| 15,2 | 0,583 | 30,8 | 1,390 | 46,4 | 1,889 | 62 | 2,212 | 77,6 | 2,407 | 93,2 | 2,466 |

| T | N | T | N | T | N | T | N | T | N | T | N |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 93,6 | 2,466 | 109,2 | 2,574 | 124,8 | 2,642 | 140,4 | 2,672 | 156 | 2,691 | 171,6 | 2,730 |
| 94 | 2,466 | 109,6 | 2,584 | 125,2 | 2,642 | 140,8 | 2,672 | 156,4 | 2,691 | 172 | 2,730 |
| 94,4 | 2,466 | 110 | 2,584 | 125,6 | 2,642 | 141,2 | 2,672 | 156,8 | 2,701 | 172,4 | 2,730 |
| 94,8 | 2,466 | 110,4 | 2,584 | 126 | 2,652 | 141,6 | 2,672 | 157,2 | 2,701 | 172,8 | 2,730 |
| 95,2 | 2,466 | 110,8 | 2,584 | 126,4 | 2,652 | 142 | 2,672 | 157,6 | 2,701 | 173,2 | 2,730 |
| 95,6 | 2,466 | 111,2 | 2,593 | 126,8 | 2,652 | 142,4 | 2,681 | 158 | 2,701 | 173,6 | 2,730 |
| 96 | 2,476 | 111,6 | 2,593 | 127,2 | 2,652 | 142.8 | 2,681 | 158,4 | 2,701 | 174 | 2,730 |
| 96,4 | 2,476 | 112 | 2,593 | 127,6 | 2,652 | 143,2 | 2,681 | 158,8 | 2,701 | 174,4 | 2,730 |
| 96,8 | 2,476 | 112,4 | 2,593 | 128 | 2,652 | 143,6 | 2,681 | 159,2 | 2,701 | 174,8 | 2,730 |
| 97,2 | 2,486 | 112,8 | 2,593 | 128,4 | 2,652 | 144 | 2,681 | 159,6 | 2,701 | 175,2 | 2,730 |
| 97,6 | 2,486 | 113,2 | 2,593 | 128,8 | 2,652 | 144,4 | 2,681 | 160 | 2,701 | 175,6 | 2,730 |
| 98 | 2,486 | 113,6 | 2,603 | 129,2 | 2,652 | 144,8 | 2,681 | 160,4 | 2,701 | 176 | 2,730 |
| 98,4 | 2,496 | 114 | 2,603 | 129,6 | 2,652 | 145,2 | 2,681 | 160,8 | 2,701 | 176,4 | 2,730 |
| 98,8 | 2,496 | 114,4 | 2,603 | 130 | 2,652 | 145,6 | 2,681 | 161,2 | 2,701 | 176,8 | 2,730 |
| 99,2 | 2,496 | 114,8 | 2,603 | 130,4 | 2,652 | 146 | 2,681 | 161,6 | 2,701 | 177,2 | 2.730 |
| 99,6 | 2,505 | 115,2 | 2,613 | 130,8 | 2,652 | 146,4 | 2,681 | 162 | 2,701 | 177,6 | 2,730 |
| 100 | 2,505 | 115,6 | 2,613 | 131,2 | 2,652 | 146,8 | 2,681 | 162,4 | 2,701 | 178 | 2,740 |
| 100,4 | 2,515 | 116 | 2,613 | 131,6 | 2,652 | 147,2 | 2,681 | 162,8 | 2,701 | 178,4 | 2,740 |
| 100,8 | 2,515 | 116,4 | 2,613 | 132 | 2,652 | 147,6 | 2,681 | 163,2 | 2,701 | 178,8 | 2,740 |
| 101,2 | 2,515 | 116,8 | 2,613 | 132,4 | 2,652 | 148 | 2,681 | 163,6 | 2,701 | 179,2 | 2,740 |
| 101,6 | 2,525 | 117,2 | 2,623 | 132,8 | 2,662 | 148,4 | 2,681 | 164 | 2,711 | 179,6 | 2,740 |
| 102 | 2,525 | 117,6 | 2,623 | 133,2 | 2,662 | 148,8 | 2,681 | 164,4 | 2,711 | 180 | 2,740 |
| 102,4 | 2,525 | 118 | 2,623 | 133,6 | 2,662 | 149,2 | 2,691 | 164,8 | 2,711 | 180,4 | 2,740 |
| 102,8 | 2,535 | 118,4 | 2,623 | 134 | 2,662 | 149,6 | 2,691 | 165,2 | 2,711 | 180,8 | 2,740 |
| 103,2 | 2,535 | 118,8 | 2,623 | 134,4 | 2,662 | 15 | 2,691 | 165,6 | 2,711 | 181,2 | 2,740 |
| 103,6 | 2,535 | 119,2 | 2.623 | 134,8 | 2,662 | 150,4 | 2,691 | 166 | 2,711 | 181,6 | 2,740 |
| 104 | 2,535 | 119,6 | 2.623 | 135,2 | 2,662 | 150,8 | 2,691 | 166,4 | 2,711 | 182 | 2,740 |
| 104,4 | 2,544 | 120 | 2.633 | 135,6 | 2,662 | 151,2 | 2,691 | 166,8 | 2,711 | 182,4 | 2,740 |
| 104,8 | 2,544 | 120,4 | 2.633 | 136 | 2,662 | 151,6 | 2,691 | 167,2 | 2,711 | 182,8 | 2,740 |
| 105,2 | 2,544 | 120,8 | 2.633 | 136,4 | 2,662 | 152 | 2,691 | 167,6 | 2,721 | 183,2 | 2,740 |
| 105,6 | 2,544 | 121,2 | 2.633 | 136,8 | 2,662 | 152,4 | 2,691 | 168 | 2,721 | 183,6 | 2,740 |
| 106 | 2,544 | 121,6 | 2.633 | 137,2 | 2,662 | 152,8 | 2,691 | 168,4 | 2,721 | 184 | 2,740 |
| 106,4 | 2,554 | 122 | 2.642 | 137,6 | 2,662 | 153,2 | 2,691 | 168,8 | 2,721 | 184,4 | 2,740 |
| 106,8 | 2,554 | 122. | 2,642 | 138 | 2,672 | 153,6 | 2,691 | 169,2 | 2,721 | 184,8 | 2,740 |
| 107,2 | 2,554 | 122,8 | 2,642 | 138,4 | 2,672 | 154 | 2,691 | 169,6 | 2,721 | 185,2 | 2,740 |
| 107,6 | ,564 | 123,2 | 2,642 | 138,8 | 2,672 | 154,4 | 2,691 | 170 | 2,721 | 185,6 | 2,740 |
| 108 | 2,564 | 123,6 | 2,642 | 139,2 | 2,672 | 154,8 | 2,691 | 170,4 | 2,721 | 186 | 2,740 |
| 108,4 | 2,564 | 124 | 2,642 | 139,6 | 2,672 | 155,2 | 2,691 | 170,8 | 2,721 | 186,4 | 2,740 |
| 108,8 | 2,574 | 124,4 | 2,642 | 140 | 2,672 | 155,6 | 2,691 | 171,2 | 2,721 | 186,8 | 2,740 |

| T | N | T | N | T | N | T | N | T | N | T | N |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 187,2 | 2,740 | 202,8 | 2,770 | 218,4 | 2,779 | 234 | 2,799 | 249,6 | 2,809 | 265,2 | 2,809 |
| 187,6 | 2,740 | 203,2 | 2,770 | 218,8 | 2,779 | 234,4 | 2,799 | 250 | 2,809 | 265,6 | 2,809 |
| 188 | 2,740 | 203,6 | 2,770 | 219,2 | 2,779 | 234,8 | 2,799 | 250,4 | 2,809 | 266 | 2,809 |
| 188,4 | 2,750 | 204 | 2,770 | 219,6 | 2,779 | 235,2 | 2,799 | 250,8 | 2,809 | 266,4 | 2,809 |
| 188,8 | 2,750 | 204,4 | 2,770 | 220 | 2,779 | 235,6 | 2,799 | 251,2 | 2,809 | 266,8 | 2,809 |
| 189,2 | 2,750 | 204,8 | 2,770 | 220,4 | 2,789 | 236 | 2,799 | 251,6 | 2,809 | 267,2 | 2,809 |
| 189,6 | 2,750 | 205,2 | 2,770 | 220,8 | 2,789 | 236,4 | 2,799 | 252 | 2,809 | 267,6 | 2,809 |
| 190 | 2.750 | 205,6 | 2,770 | 221,2 | 2,789 | 236,8 | 2,799 | 252,4 | 2,809 | 268 | 2,809 |
| 190,4 | 2.750 | 206 | 2,770 | 221,6 | 2,789 | 237,2 | 2,799 | 252,8 | 2,809 | 268,4 | 2.819 |
| 190,8 | 2,750 | 206,4 | 2,770 | 222 | 2,789 | 237,6 | 2,799 | 253,2 | 2,819 | 268,8 | 2,819 |
| 191,2 | 2,750 | 206,8 | 2,770 | 222,4 | 2,789 | 238 | 2,799 | 253,6 | 2,819 | 269,2 | 2,819 |
| 191,6 | 2,750 | 207,2 | 2,770 | 222,8 | 2,789 | 238,4 | 2,799 | 254 | 2,819 | 269,6 | 2,809 |
| 192 | 2,750 | 207,6 | 2,770 | 223,2 | 2,789 | 238,8 | 2,799 | 254,4 | 2,809 | 270 | 2,809 |
| 192,4 | 2,750 | 208 | 2,770 | 223,6 | 2,789 | 239,2 | 2,799 | 254,8 | 2,809 | 270,4 | 2,809 |
| 192,8 | 2,750 | 208,4 | 2,770 | 224 | 2,789 | 239,6 | 2,809 | 255,2 | 2,809 | 270,8 | 2,809 |
| 193,2 | 2,750 | 208,8 | 2,770 | 224,4 | 2,789 | 240 | 2,809 | 255,6 | 2,809 | 271,2 | 2,809 |
| 193,6 | 2,750 | 209,2 | 2,770 | 224,8 | 2,789 | 240,4 | 2,809 | 256 | 2,809 | 271,6 | 2,819 |
| 194 | 2,750 | 209,6 | 2,770 | 225,2 | 2,789 | 240,8 | 2,809 | 256,4 | 2,819 | 272 | 2,809 |
| 194,4 | 2,760 | 210 | 2,770 | 225,6 | 2,789 | 241,2 | 2,809 | 256,8 | 2,809 | 272,4 | 2,809 |
| 194,8 | 2,760 | 210,4 | 2,770 | 226 | 2,789 | 241,6 | 2,809 | 257,2 | 2,809 | 272,8 | 2,809 |
| 195,2 | 2,760 | 210,8 | 2,779 | 226,4 | 2,789 | 242 | 2,809 | 257,6 | 2,809 | 273,2 | 2,809 |
| 195,6 | 2,760 | 211,2 | 2,779 | 226,8 | 2,789 | 242,4 | 2,809 | 258 | 2,809 | 273,6 | 2,809 |
| 196 | 2,760 | 211,6 | 2,779 | 227,2 | 2,789 | 242,8 | 2,809 | 258,4 | 2,809 | 274 | 2,809 |
| 196,4 | 2,760 | 212 | 2,779 | 227,6 | 2,789 | 243,2 | 2,809 | 258,8 | 2,809 | 274,4 | 2,809 |
| 196,8 | 2,760 | 212.4 | 2,779 | 228 | 2,789 | 243,6 | 2,809 | 259,2 | 2,809 | 274,8 | 2,809 |
| 197,2 | 2,760 | 212,8 | 2,779 | 228,4 | 2,789 | 244 | 2,809 | 259,6 | 2,809 | 275,2 | 2,809 |
| 197,6 | 2,760 | 213,2 | 2,779 | 228.8 | 2,789 | 244,4 | 2,809 | 260 | 2,809 | 275,6 | 2,809 |
| 198 | 2,760 | 213,6 | 2,779 | 229,2 | 2,789 | 244,8 | 2,809 | 260,4 | 2,809 | 276 | 2,809 |
| 198,4 | 2,760 | 214 | 2,779 | 229,6 | 2,799 | 245,2 | 2,809 | 260,8 | 2,809 | 276,4 | 2,809 |
| 198,8 | 2,760 | 214,4 | 2,779 | 230 | 2,799 | 245,6 | 2,809 | 261,2 | 2,809 | 276,8 | 2,809 |
| 199,2 | 2,760 | 214,8 | 2,779 | 230,4 | 2,799 | 246 | 2,809 | 261,6 | 2,809 | 277,2 | 2,809 |
| 199,6 | 2,760 | 215,2 | 2,779 | 230,8 | 2,799 | 246,4 | 2,809 | 262 | 2,809 | 277,6 | 2,809 |
| 200 | 2,760 | 215,6 | 2,779 | 231,2 | 2,799 | 246,8 | 2,809 | 262,4 | 2,809 | 278 | 2,809 |
| 200,4 | 2.760 | 216 | 2,779 | 231,6 | 2,799 | 247,2 | 2,809 | 262,8 | 2,809 | 278,4 | 2,809 |
| 200,8 | 2,760 | 216,4 | 2,779 | 232 | 2,799 | 247,6 | 2,809 | 263,2 | 2,809 | 278,8 | 2,809 |
| 201,2 | 2,770 | 216,8 | 2,779 | 232,4 | 2,799 | 248 | 2,809 | 263,6 | 2,809 | 279,2 | 2,809 |
| 201,6 | 2,770 | 217,2 | 2,779 | 232. | 2,799 | 248,4 | 2,809 | 264 | 2,809 | 279,6 | 2,809 |
| 202 | 2,770 | 217,6 | 2,779 | 233. | 2,799 | 248,8 | 2,809 | 264,4 | 2,809 | 280 | 2,809 |
| 202,4 | 2,770 | 218 | 2,779 | 233,6 | 2,799 | 249,2 | 2,809 | 264,8 | 2,809 | 280,4 | 2,809 |

**Gráfica de ajuste a los datos experimentales.**

El objetivo del experimento de entrada al escalón es encontrar la ecuación de la curva que mejor se ajuste a los datos experimentales de acuerdo a la ecuación (5), para ello es necesario encontrar los valores de $K$ y $\tau$ pertenecientes a la ecuación del sistema de primer orden en el tiempo y para la ecuación de la función de transferencia en el plano complejo $s$. Éstos valores se encontraron mediante el método de mínimos cuadrados.

La ecuación respuesta en el tiempo de un sistema de primer orden (5) se reemplaza en la ecuación objetivo de mínimos cuadrados (10)

$$\hat{Nivel}(t) = AK(1 - e^{-\frac{t}{\tau}})$$

obteniendo así una ecuación cuadrática que se encuentra en función de los valores experimentales recolectados y los niveles predichos por la ecuación respuesta del sistema de primer orden en el tiempo.

$$f(K, \tau) = \sum_{i=1}^{n} (Nivel_i - \hat{Nivel}_i)^2$$

posteriormente se sustituye la ecuación $\hat{Nivel}_i$ en la ecuación objetivo cuadrática, obteniendo así:

$$f(K, \tau) = \sum_{i=1}^{n} (Nivel_i - AK(1 - e^{-\frac{t_i}{\tau}}))^2$$

finalmente se reemplaza los valores del tiempo $(T_i)$, los valores del nivel del tanque alcanzados en cada tiempo de muestreo $(Nivel_i)$ y el valor del escalón aplicado al sistema correspondiente a $A = 4{,}1808965$.

$$f(K, \tau) = \sum_{i=1}^{702} (Nivel_i - 1K(1 - e^{\frac{T_i}{\tau}}))^2$$

Para encontrar los valores de $K$ y $\tau$ necesarios para la función de transferencia de primer orden se procede a minimizar la función cuadrática, para lo cual se debe derivar parcialmente la función para cada variable e igualarlas a cero, obteniendo:

$$\frac{\partial f(K, \tau)}{\partial K} = 0 \qquad \text{y} \qquad \frac{\partial f(K, \tau)}{\partial \tau} = 0$$

Las derivadas parciales de la función objetivo cuadrática fueron realizadas en el software Mathematica, encontrando un sistema de dos ecuaciones con dos incógnitas, el cual fue resuelto obteniéndose los valores de: $K$=0.6718 y $\tau$ = 44.05

Los valores de $K$ y $\tau$ hallados son reemplazados en (5) obteniendo así la ecuación de respuesta del sistema de primer orden en el tiempo.

$$y(t) = 4{,}1808965 * 0{,}6718(1 - e^{-\frac{t}{44{,}05}})$$

Del mismo modo los valores $K$ y $\tau$ son sustituidos en la ecuación (3) correspondiente a una función de transferencia de primer orden.

$$G(s) = \frac{0{,}6718}{44{,}05s + 1}$$

En la figura 31 se representa los valores de los datos experimentales y la ecuación que mejor se ajusta a los datos.

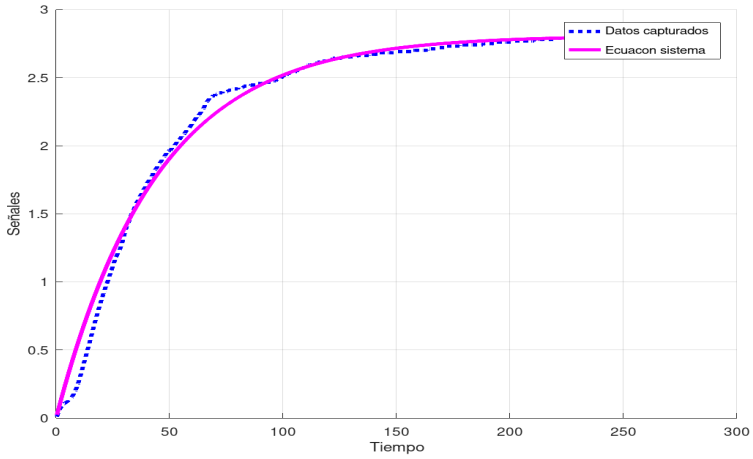


Figura 31: Ajuste a los datos experimentales.

Fuente: Elaborado por la autora.

### 4.1.4. Función de transferencia discreta.

El algoritmo empleado para la implementación del sistema de control predictivo con redes neuronales en el proceso de nivel de la planta FESTO MPS PA® Compact Workstation se lo realizó en un sistema computacional, razón por la cual se utilizaron ecuaciones en tiempo discreto. La función de transferencia del proceso de nivel en tiempo continuo fue transformada al tiempo discreto, partiendo de:

$$G(s) = \frac{0{,}6718}{44{,}05s + 1} = \frac{0{,}0152}{s + 0{,}0227}$$

mediante fracciones parciales se separa términos

$$\frac{G(s)}{s} = \frac{0{,}0152}{s(s + 0{,}0227)} = \frac{0{,}6696}{s} - \frac{0{,}6696}{s + 0{,}0227}$$

posteriormente se aplica la fórmula de la ecuación de la transformada $z$ y considerando un tiempo de muestreo de $0{,}4$ segundos se tiene:

$$\mathcal{Z}\{\frac{G(s)}{s}\} = 0{,}6696(\frac{z}{z-1}) - 0{,}6696(\frac{z}{z-e^{-0{,}0227*0{,}4}})$$

$$\mathcal{Z}\{\frac{G(s)}{s}\} = z(\frac{0{,}006073}{(z-1)(z-0{,}991)})$$

posteriormente la transformada $z$ deseada es:

$$G(z) = \frac{z-1}{z}\frac{z(0{,}006073)}{(z-1)(z-0{,}991)}$$

$$G(z) = \frac{0{,}006073}{z-0{,}991}$$

En la figura 32 se indican las respectivas respuestas al escalón de la función de transferencia encontradas en tiempo discreto; a consecuencia de un tiempo de muetreo muy pequeño la gráfica en tiempo discreto se parece a una gráfica en tiempo continuo.

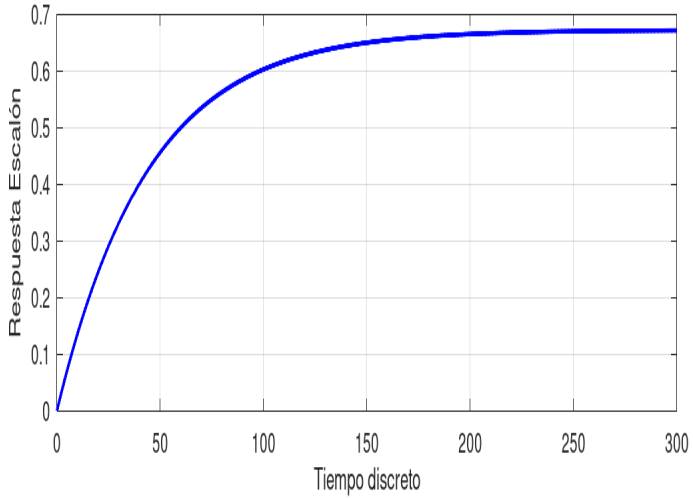


Figura 32: Respuesta al escalón en tiempo discreto.

Fuente: Elaborado por la autora.

### 4.1.5. Implementación del controlador predictivo DMC con redes neuronales.

Para encontrar la dinámica del sistema de nivel de la planta Festo se aplicaron diferentes valores de setpoint a la planta, para lo cual se capturaron 7912 datos de los cuales el 69 % fueron utilizados para el entrenamiento y el 31 % para el testeo de la red neuronal. La estructura del experimento será la misma indicada en la figura 28; los datos capturados se representan en la figura 33.

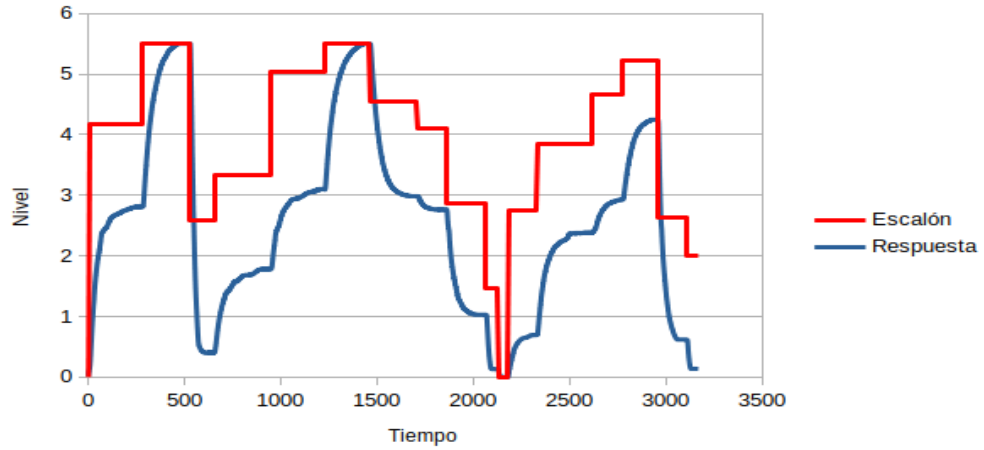


Figura 33: Datos para entrenamiento y testeo.

Fuente: Elaborado por la autora.

Para demostrar el funcionamiento del controlador predictivo de matriz dinámica con redes neuronales implementado en el proceso de nivel de la planta FESTO MPS PA Compact Workstation se trabajó con un horizontes de predicación $P = 15$ y un horizonte de control $N = 5$; además, se consideró los valores de ponderación $\lambda = 10$ y delta $\delta = 0{,}2$. A continuación se representa los pasos necesarios para la implementación del sistema de control DMC con redes neuronales.

- Salida de la planta $y(t)$ (identificación del sistema).

En general un sistema no-lineal puede ser representado en términos de entrada-salida como se indica a continuación:

$$y(k+1) = f[y(k), y(k+1), ..., y(k-n+1); u(k), u(k-1), ..., u(k-m+1)]$$

El problema de la identificación consiste en escoger un modelo apropiado y ajustar sus parámetros de acuerdo a alguna ley adaptativa, de forma que la respuesta del modelo a una señal de entrada (o bien un conjunto de señales de entrada) se aproxime a la respuesta del sistema real de esa misma entrada.

En este caso el objetivo es aproximar la función (dinámica del sistema) utilizando redes neuronales, por lo que la identificación de dicha función consiste en ajustar los parámetros de la red neuronal de manera que aproxime la función deseada. En la figura 34 se representa el diagrama de bloques para la identificación del sistema de nivel de la planta FESTO MPS PA® Compact Workstation.

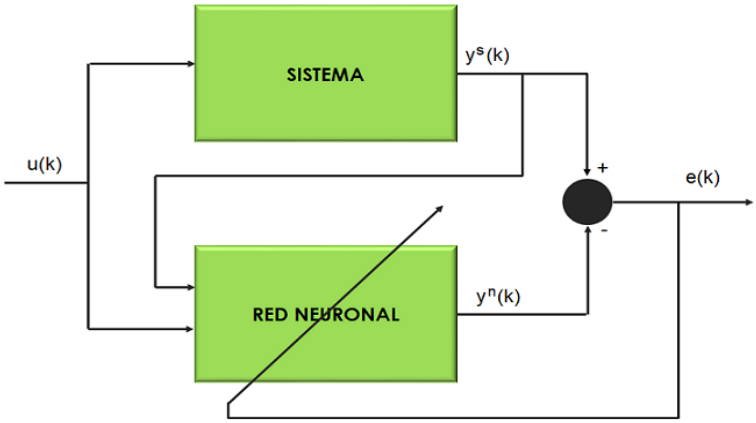


Figura 34: Esquema de identificación Serie/ Paralelo.

Fuente: Elaborado por la autora.

La primera etapa para la implementación del control predictivo es entrenar la red neuronal para representar la dinámica del sistema de nivel de líquido de la planta FESTO MPS PA Compact Workstation. La salida real de la planta (dinámica del sistema) fue aproximada mediante el entrenamiento de una red neuronal feed-forward; para el entrenamiento y testeo la red neuronal se utilizó entradas de control anteriores ($u(k-1)$) y salidas de la planta anteriores ($y^s(k-1)$) como se indica en la figura 34. La estructura de la red neuronal para la identificación del sistema dinámico se indica en la figura 35, dicha red neuronal se entrenó fuera de línea utilizando los datos recopilados de diferentes entradas escalón implementadas al sistema de nivel.

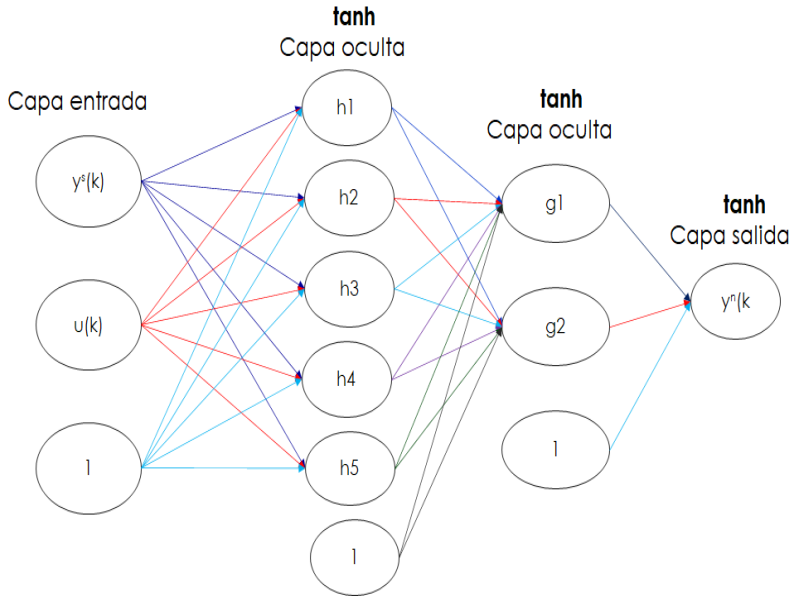


Figura 35: Red Neuronal implementada.

Fuente:Elaborado por la investigadora en base a la red neuronal planteada.

Para el entrenamiento de la red neuronal se empleó la siguiente estructura: una capa de entrada, dos capas ocultas y una capa de salida. La primera capa contiene dos entradas, las dos capas ocultas contienen 5 y 2 neuronas respectivamente con la función de activación tangente hiperbólica y finalmente la capa de salida de una sola neurona también tiene la función de activación tangente hiperbólica. En la figura 36 se observa que la red neuronal predice satisfactoriamente la dinámica del sistema.

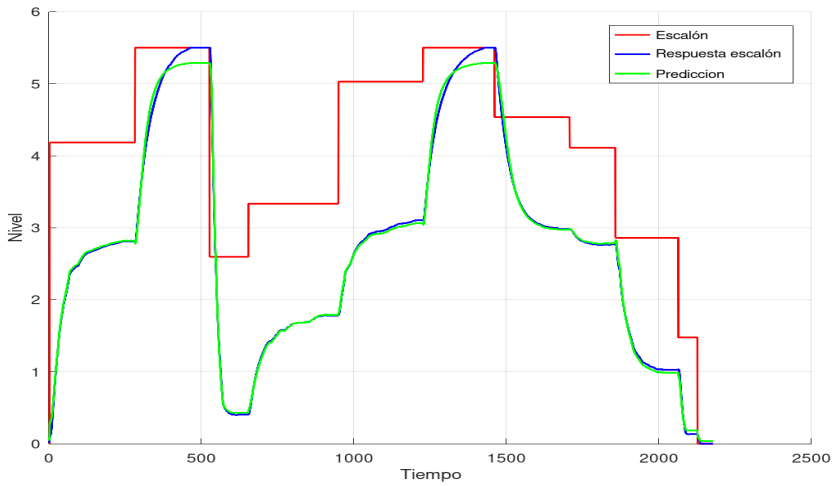


Figura 36: Entrenamiento de la Red Neuronal.

Fuente: Elaborado por la autora.

En la figura 37 se representa el valor de la función de coste utilizada para el entrenamiento de la red neuronal (Error Cuadrático Medio), el cual se reduce hasta llegar al valor mínimo en las 200 iteraciones establecidas.

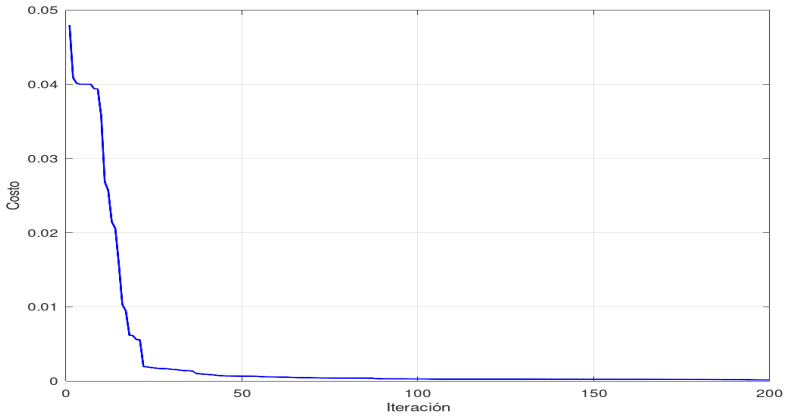


Figura 37: Función de coste.

Fuente: Elaborado por la autora.

Los pesos de la red neuronal encontrados en el entrenamiento con la estructura indicada anteriormente se representan a continuación:

- Primera capa $\mathbf{W}^{(1)}$

| (1) | $y^s(k)$ | $u(k)$ |
|---|---|---|
| $-0{,}057463$ | $-0{,}12522$ | $-0{,}15229$ |
| $0{,}22236$ | $-0{,}015489$ | $-0{,}51529$ |
| $-0{,}022445$ | $-0{,}26283$ | $-0{,}37709$ |
| $-0{,}59987$ | $-0{,}1417$ | $0{,}75315$ |
| $1{,}3125$ | $0{,}25358$ | $-1{,}4914$ |

- Segunda capa $\mathbf{W}^{(2)}$

| (1) | h1 | h2 | h3 | h4 | h5 |
|---|---|---|---|---|---|
| $0{,}42412$ | $0{,}040999$ | $0{,}69167$ | $0{,}1782$ | $-1{,}1735$ | $1{,}8725$ |
| $-0{,}31849$ | $0{,}13824$ | $0{,}32327$ | $0{,}29777$ | $-0{,}35755$ | $0{,}27625$ |

- Capa de salida $\mathbf{W}^{(3)}$

| (1) | g1 | g2 |
|---|---|---|
| $1{,}6367$ | -1,4177 | -1,4246 |

Posterior al entrenamiento y obteniendo los pesos de cada una de las capas de la red neuronal se procede con datos diferentes a los utilizados para el entrenamiento a verificar las predicciones de la red neuronal (testeo). En la figura 38 se representa los resultados del testeo de la red neuronal y se visualiza una predicción aceptable en el comportamiento dinámico del proceso de nivel de la planta FESTO.

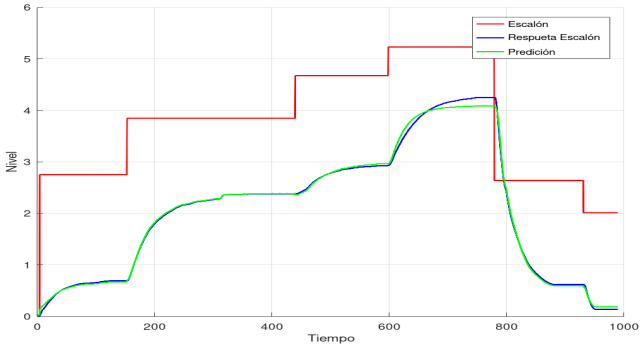


Figura 38: Testeo de la red neuronal.

Fuente: Elaborado por la autora.

- Respuesta libre, coeficientes al escalón y predicción.

Como se indicó en la fundamentación teórica la respuesta libre $f(t+k)$ viene dada por:

$$f(t+k) = y_m(t) + \sum_{i=1}^{\infty}(g_{k+i} - g_i)\Delta u(t-i)$$

A partir de esta expresión formaremos el vector libre **f** para cada una de las predicciones hasta el horizonte de predicción $P = 15$ establecido.

Los coeficientes $g_i$ se pueden obtener directamente de la respuesta al escalón de la función de transferencia en tiempo discreto. El sistema se estabiliza luego de $250$ segundos y para el caso de estudio se capturaron 702 muestras; la salida del sistema esta dado por:

$$y(t) = \sum_{i=1}^{702} g_i \Delta u(t-i)$$

los coeficientes $g_i$ se representan en las figuras 44,45,46,47.

La respuesta que se muestra en la figura 39 corresponde a un sistema con una función de transferencia dada por:

$$G(z) = \frac{0{,}0076}{z - 0{,}9817}$$

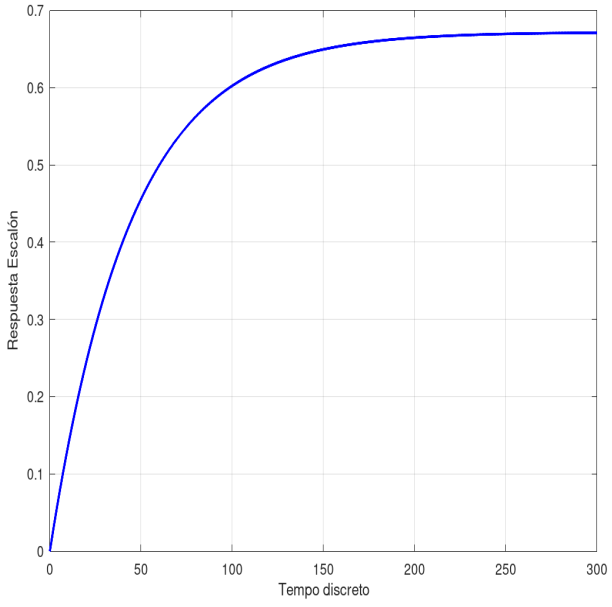


Figura 39: Respuesta al escalón en tiempo discreto.

Fuente: Elaborado por la autora.

Considerando el horizonte de predicción $P = 15$ y el horizonte de control $N = 5$, la matriz dinámica $G$ se obtiene a partir de los coeficientes de la respuesta al escalón y está dada por:

$$\mathbf{G} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0{,}0060727 & 0 & 0 & 0 & 0 \\ 0{,}012091 & 0{,}0060727 & 0 & 0 & 0 \\ 0{,}018054 & 0{,}012091 & 0{,}0060727 & 0 & 0 \\ 0{,}023964 & 0{,}018054 & 0{,}012091 & 0{,}0060727 & 0 \\ 0{,}02982 & 0{,}023964 & 0{,}018054 & 0{,}012091 & 0{,}0060727 \\ 0{,}035623 & 0{,}02982 & 0{,}023964 & 0{,}018054 & 0{,}012091 \\ 0{,}041374 & 0{,}035623 & 0{,}02982 & 0{,}023964 & 0{,}018054 \\ 0{,}047072 & 0{,}041374 & 0{,}035623 & 0{,}02982 & 0{,}023964 \\ 0{,}052719 & 0{,}047072 & 0{,}041374 & 0{,}035623 & 0{,}02982 \\ 0{,}058316 & 0{,}052719 & 0{,}047072 & 0{,}041374 & 0{,}035623 \\ 0{,}063861 & 0{,}058316 & 0{,}052719 & 0{,}047072 & 0{,}041374 \\ 0{,}069357 & 0{,}063861 & 0{,}058316 & 0{,}052719 & 0{,}047072 \\ 0{,}074802 & 0{,}069357 & 0{,}063861 & 0{,}058316 & 0{,}052719 \\ 0{,}080199 & 0{,}074802 & 0{,}069357 & 0{,}063861 & 0{,}058316 \end{bmatrix}$$

Las predicciones están dadas por:

$$\hat{\mathbf{y}} = \mathbf{G}\mathbf{u} + \mathbf{f}$$

$$\begin{bmatrix} \hat{y}(t+1|t) \\ \hat{y}(t+2|t) \\ \hat{y}(t+3|t) \\ \hat{y}(t+4|t) \\ \hat{y}(t+5|t) \\ \hat{y}(t+6|t) \\ \hat{y}(t+7|t) \\ \hat{y}(t+8|t) \\ \hat{y}(t+9|t) \\ \hat{y}(t+10|t) \\ \hat{y}(t+11|t) \\ \hat{y}(t+12|t) \\ \hat{y}(t+13|t) \\ \hat{y}(t+14|t) \\ \hat{y}(t+15|t) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0{,}006072 & 0 & 0 & 0 & 0 \\ 0{,}012091 & 0{,}006072 & 0 & 0 & 0 \\ 0{,}018054 & 0{,}012091 & 0{,}006072 & 0 & 0 \\ 0{,}023964 & 0{,}018054 & 0{,}012091 & 0{,}006072 & 0 \\ 0{,}02982 & 0{,}023964 & 0{,}018054 & 0{,}012091 & 0{,}006072 \\ 0{,}035623 & 0{,}02982 & 0{,}023964 & 0{,}018054 & 0{,}012091 \\ 0{,}041374 & 0{,}035623 & 0{,}02982 & 0{,}023964 & 0{,}018054 \\ 0{,}047072 & 0{,}041374 & 0{,}035623 & 0{,}02982 & 0{,}023964 \\ 0{,}052719 & 0{,}047072 & 0{,}041374 & 0{,}035623 & 0{,}02982 \\ 0{,}058316 & 0{,}052719 & 0{,}047072 & 0{,}041374 & 0{,}035623 \\ 0{,}063861 & 0{,}058316 & 0{,}052719 & 0{,}047072 & 0{,}041374 \\ 0{,}069357 & 0{,}063861 & 0{,}058316 & 0{,}052719 & 0{,}047072 \\ 0{,}074802 & 0{,}069357 & 0{,}063861 & 0{,}058316 & 0{,}052719 \\ 0{,}080199 & 0{,}074802 & 0{,}069357 & 0{,}063861 & 0{,}058316 \end{bmatrix} \begin{bmatrix} \Delta u(t) \\ \Delta u(t+1) \\ \Delta u(t+2) \\ \Delta u(t+3) \\ \Delta u(t+4) \end{bmatrix} +$$

$$
+ \begin{bmatrix} f(t+1) \\ f(t+2) \\ f(t+3) \\ f(t+4) \\ f(t+5) \\ f(t+6) \\ f(t+7) \\ f(t+8) \\ f(t+9) \\ f(t+10) \\ f(t+11) \\ f(t+12) \\ f(t+13) \\ f(t+14) \\ f(t+15) \end{bmatrix}
$$

- Incremento de control $\Delta u(t)$ (ley de control).

De acuerdo a la ley control del DMC establecida en la fundamentación teórica se tiene que $\Delta u = \mathbf{K}(\mathbf{w}-\mathbf{f})$, donde $\mathbf{K}$ representa la primera fila de la matriz $(\mathbf{G}^T\delta\mathbf{I}\mathbf{G}+\lambda\mathbf{I})^{-1}\mathbf{G}^T\delta\mathbf{I}$, $\mathbf{w}$ es el vector de referencia deseado y $\mathbf{f}$ es el vector de respuesta libre, además se ha considerado un $\delta = 0{,}2$ y un $\lambda = 10$ debido a que estos valores permiten tener un mejor rendimiento en el contolador.

$$
(\mathbf{G}^T\delta\mathbf{I}\mathbf{G} + \lambda\mathbf{I})^{-1}\mathbf{G}^T\delta\mathbf{I} =
$$

$$
\begin{bmatrix}
0{,}0000 & 0{,}0001 & 0{,}0002 & 0{,}0004 & 0{,}0005 & 0{,}0006 & 0{,}0007 & 0{,}0008 & 0{,}0009 & 0{,}0011 & 0{,}0012 & 0{,}0013 & 0{,}0014 & 0{,}0015 & 0{,}0016 \\
0{,}0000 & 0{,}0000 & 0{,}0001 & 0{,}0002 & 0{,}0004 & 0{,}0005 & 0{,}0006 & 0{,}0007 & 0{,}0008 & 0{,}0009 & 0{,}0011 & 0{,}0012 & 0{,}0013 & 0{,}0014 & 0{,}0015 \\
0{,}0000 & 0{,}0000 & 0{,}0000 & 0{,}0001 & 0{,}0002 & 0{,}0004 & 0{,}0005 & 0{,}0006 & 0{,}0007 & 0{,}0008 & 0{,}0009 & 0{,}0011 & 0{,}0012 & 0{,}0013 & 0{,}0014 \\
0{,}0000 & 0{,}0000 & 0{,}0000 & 0{,}0000 & 0{,}0001 & 0{,}0002 & 0{,}0004 & 0{,}0005 & 0{,}0006 & 0{,}0007 & 0{,}0008 & 0{,}0009 & 0{,}0011 & 0{,}0012 & 0{,}0013 \\
0{,}0000 & 0{,}0000 & 0{,}0000 & 0{,}0000 & 0{,}0000 & 0{,}0001 & 0{,}0002 & 0{,}0004 & 0{,}0005 & 0{,}0006 & 0{,}0007 & 0{,}0008 & 0{,}0009 & 0{,}0011 & 0{,}0012
\end{bmatrix}
$$

$$
\mathbf{K} = \begin{bmatrix} 0{,}0000 & 0{,}0001 & 0{,}0002 & 0{,}0004 & 0{,}0005 & 0{,}0006 & 0{,}0007 & 0{,}0008 & 0{,}0009 & 0{,}0011 & 0{,}0012 & 0{,}0013 & 0{,}0014 & 0{,}0015 & 0{,}0016 \end{bmatrix}
$$

De acuerdo a la ecuación de ley de control se tiene:

$$
\Delta u(t) = \mathbf{K}(\mathbf{w} - \mathbf{f})
$$

$$
\Delta u(t) = \begin{bmatrix} 0{,}0000 & 0{,}0001 & 0{,}0002 & 0{,}0004 & 0{,}0005 & 0{,}0006 & 0{,}0007 & 0{,}0008 & 0{,}0009 & 0{,}0011 & 0{,}0012 & 0{,}0013 & 0{,}0014 & 0{,}0015 & 0{,}0016 \end{bmatrix}
$$

$$\begin{bmatrix} w(t+1) - f(t+1) \\ w(t+2) - f(t+2) \\ w(t+3) - f(t+3) \\ w(t+4) - f(t+4) \\ w(t+5) - f(t+5) \\ w(t+6) - f(t+6) \\ w(t+7) - f(t+7) \\ w(t+8) - f(t+8) \\ w(t+9) - f(t+9) \\ w(t+10) - f(t+10) \\ w(t+11) - f(t+11) \\ w(t+12) - f(t+12) \\ w(t+13) - f(t+13) \\ w(t+14) - f(t+14) \\ w(t+15) - f(t+15) \end{bmatrix}$$

donde $w(t+i)$ es la trayectoria de referencia que puede considerarse constante e igual al setpoint actual y $f(t+i)$ es la respuesta libre hasta el horizonte $P = 15$. Para ejemplificar de mejor manera se eligió un instante $k = 100$ en la simulación del algoritmo de control, donde se obtuvieron los valores de los vectores **f** y **w**.

$$\mathbf{f} = \begin{bmatrix} 2{,}8162 \\ 2{,}8262 \\ 2{,}8361 \\ 2{,}8459 \\ 2{,}8556 \\ 2{,}8653 \\ 2{,}8748 \\ 2{,}8843 \\ 2{,}8937 \\ 2{,}903 \\ 2{,}9122 \\ 2{,}9214 \\ 2{,}9304 \\ 2{,}9394 \\ 2{,}9483 \end{bmatrix} \quad \text{y} \quad \mathbf{w} = \begin{bmatrix} 4{,}0000 \\ 4{,}0000 \\ 4{,}0000 \\ 4{,}0000 \\ 4{,}0000 \\ 4{,}0000 \\ 4{,}0000 \\ 4{,}0000 \\ 4{,}0000 \\ 4{,}0000 \\ 4{,}0000 \\ 4{,}0000 \\ 4{,}0000 \\ 4{,}0000 \\ 4{,}0000 \end{bmatrix}$$

$$\Delta u(t) = \mathbf{K}(\mathbf{w} - \mathbf{f})$$

Se obtiene un incremento de control para un instante $k$:

$$\Delta u(t) = 0{,}013364$$

- Salida de control $u(t)$.

La señal de control $u(t)$ es simplemente la suma de la señal de control anterior más el incremento de control $\Delta u(t)$ calculada en el anterior paso.

$$u(t) = u(t-1) + \Delta u(t)$$

### 4.1.6. Conexión física del sistema de nivel implementado.

La distribución física para la implementación del controlador predictivo para el sistema de nivel de la planta FESTO se representa en la figura 40.

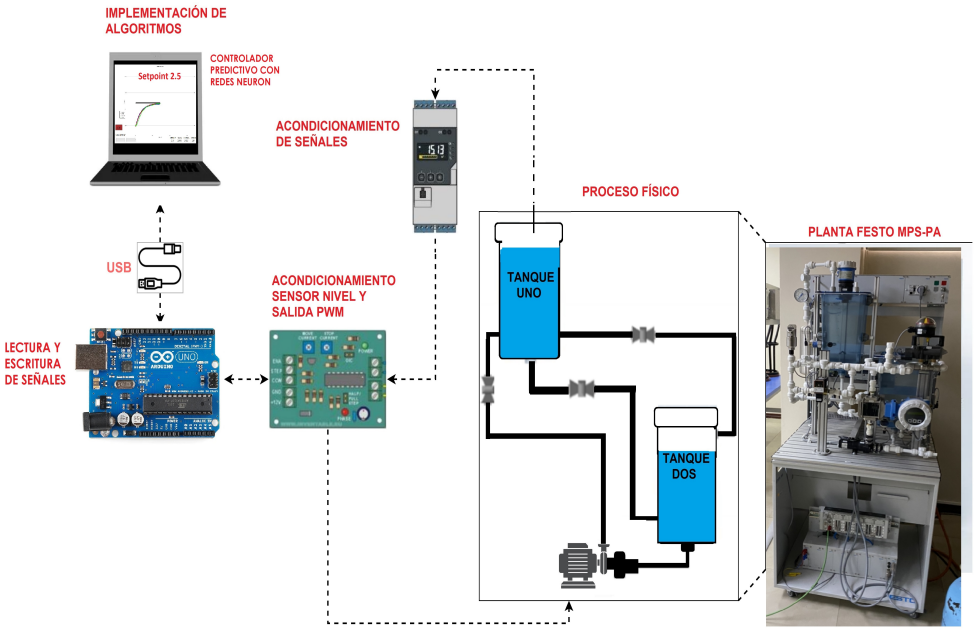


Figura 40: Esquema físico del sistema implementado.

Fuente: Elaborado por la autora.

El sistema está constituido por una computadora (laptop i5), una tarjeta de adquisición de datos Arduino Uno, una tarjeta de acondicionamiento de señales, un transmisor de proceso con unidad de control RMA42 y el sistema de nivel de la planta, la implementación tiene una conexión usb.

- En la computadora (laptop i5) se ejecuta el código fuente del algoritmo del controlador predictivo con redes neuronales en el software Octave.
- La tarjeta Arduino-Uno se utiliza como puerta de comunicación entre la computadora y la tarjeta de acondicionamiento de señales.
- La tarjeta de acondicionamiento de señales adapta los rangos de voltaje estandarizados para conectarlos al Arduino-Uno. Además, contiene el circuito de potencia para la manipulación de la bomba mediante PWM.
- El transmisor de proceso con unidad de control RMA42 transforma la señal de corriente 4-20mA del sensor ultrasónico en una señal de 0 a 10V.
- El sistema de nivel de la planta Festo MPS-PA Compact Workstation es donde se implementará el controlador predictivo.

### 4.1.7. Prueba de funcionamiento

Para comprobar el correcto funcionamiento del controlador predictivo de matriz dinámica con redes neuronales se asignó un setpoint de 2.75 litros. Se observa que la señal de salida ($y(t)$) alcanza un valor muy cercano al valor deseado al igual que la señal capturada por el sensor; además, se visualiza que la señal de control ($u(t)$) enviada a la bomba no presenta un incremento alto debido al valor de lambda asignado $\lambda = 10$.

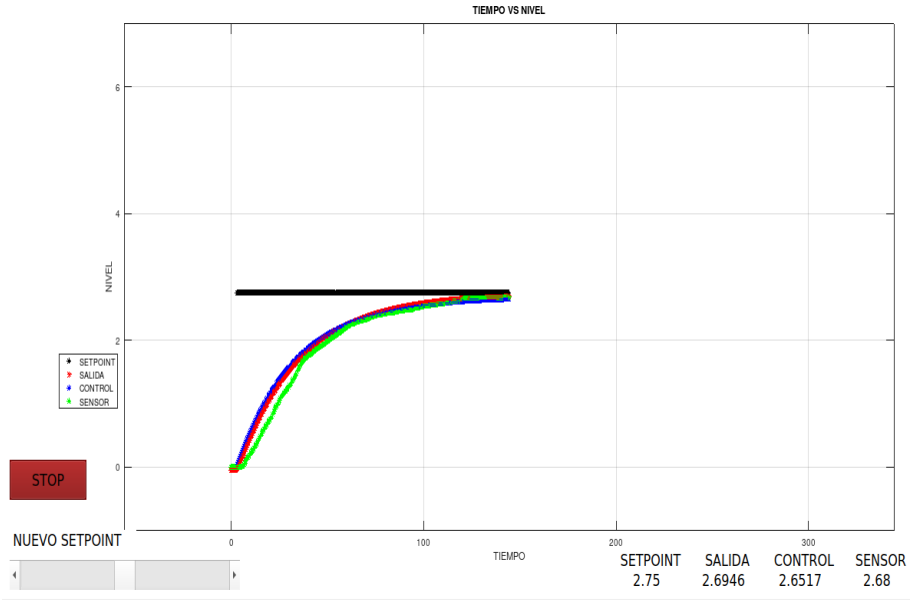


Figura 41: Prueba de funcionamiento 1.

Fuente: Elaborado por la autora.

## 4.2. Discusión.

El análisis de regresión exponencial perteneciente a la respuesta del sistema en el tiempo genera la siguiente ecuación:

$$y(t) = 4{,}1808965 * 0{,}6718(1 - e^{-\frac{t}{44{,}05}})$$

La misma que describe la relación estadística entre la variable predictora ($tiempo$) y la variable de respuesta ($y(t)$).

Los valores de:

$$K = 0{,}6718$$

$$y$$

$$\tau = 44{,}05$$

que fueron encontrados en la ecuación de respuesta del sistema en el tiempo e interpretados en la función de transferencia,se presenta a continuación:

El valor $Pr$ de cada término $K = 0{,}6718$ y $\tau = 44{,}05$ evalúa la hipótesis nula la cual indica que los coeficientes $K$ o $\tau$ sean igual a cero. Los valores obtenidos de $Pr$ fueron menor que $0{,}05$ lo que indica que se rechaza la hipótesis nula y se acepta la hipótesis alternativa, lo que significa que el predictor tiene una adición significativa al modelo debido a los cambios en el valor y se relacionan con cambios en la variable de respuesta. Por otro lado, un valor $Pr$ mayor a 0.05 indica que los cambios en el predictor no están asociados con cambios en la respuesta.

En la tabla 2 se observa que las variables predictoras $K$ (ganancia del sistema) y $\tau$ (constante de tiempo en el sistema) son significativas, porque sus probabilidades $Pr < 0{,}05$.

Tabla 2: Observación de resultados obtenidos

| | $K$ | $\tau$ |
|---|---|---|
| Valores Estimados | 6,718e-01 | 4,405e+01 |
| Error Estandar $\sigma$ | 9,968e-04 | 2,897e-01 |
| Valor t | 674,0 | 152,1 |
| Pr | <2e-17 | <2e-17 |

Para la implementación del control predictivo se entrenó la red neuronal para representar la dinámica del sistema de nivel de líquido de la planta FESTO MPS PA® Compact Workstation. La salida real de la planta (dinámica del sistema) fue aproximada mediante el entrenamiento de una red neuronal feed-forward. Los pesos sinápticos encontrados

en el entrenamiento de la red neuronal fueron utilizados para el testeo de dicha red. Para lo cual se aplicaron distintos setpoints (diferentes al entrenamiento) y se visualizó las predicciones alcanzadas por la red neuronal como se mostró en la figura 38.

Se observa que la predicación es aceptable cuando se aplica diferentes entradas escalón al sistema de nivel, demostrando así que las redes neuronales son una herramienta muy útil para la identificación de sistemas dinámicos y, en el caso particular, para un sistema de nivel de líquido.

Para visualizar el comportamiento de la planta al implementar el controlador predictivo de matriz dinámica con redes neuronales se realizó un experimento aplicando varios setpoint de diferentes valores, correspondientes a $4$, $1$ y $5$ litros; los valores de setpoint seleccionados son alcanzados y los valores de la señal de control se los indica en la figura 42

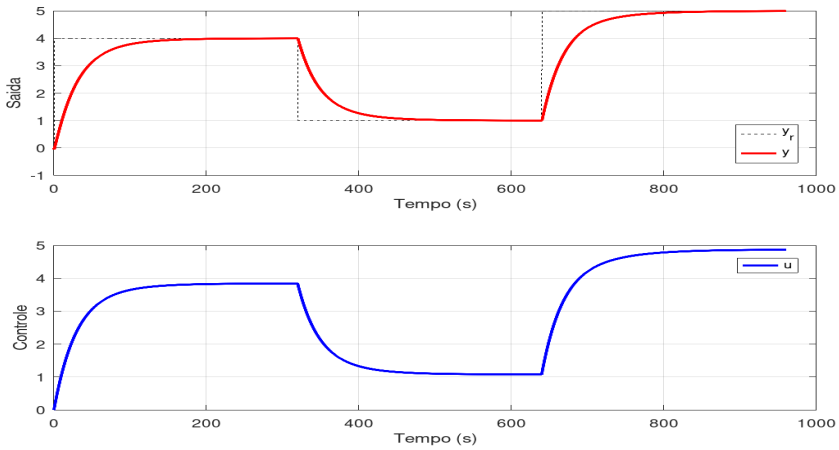


Figura 42: Experimento con diferentes setpoints.
Fuente: Elaborado por la investigadora en base a los resultados obtenidos.

De igual manera, se aplicaron setpoints que están fuera del rango de nivel del tanque de la planta Festo como se indica en la figura 45. Éstos setpoints no son alcanzados en la simulación del sistema de control y no se lo implementó en forma física debido a los posibles daños en los componentes de dicha planta. Éste tipo de problemas se debe a los datos de entrenamiento los cuales no sobrepasan el nivel de $5{,}5$ litros.

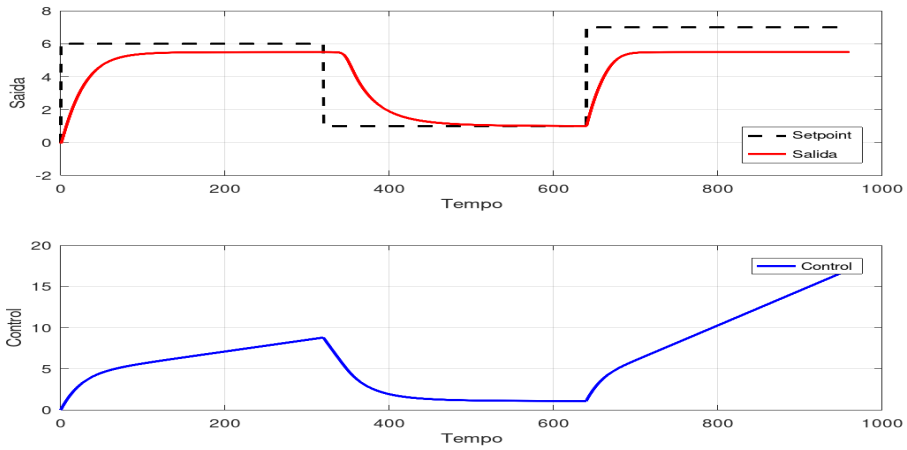


Figura 43: Setpoints no alcanzados por el controlador.

Fuente: Elaborado por la autora.

# CAPÍTULO V
# CONCLUSIONES Y RECOMENDACIONES

## 5.1. Conclusiones.

- Se realizó una indagación de las diferentes estructuras de las redes feed-forward para resolver una gran variedad de problemas de interés, y en particular para implementar un sistema de control. Las redes feed-forward se dividen en redes simples que tienen una única capa, redes multicapa las cuales tienen una capa oculta y redes profundas que contienen varias capas ocultas. Para el sistema de control predictivo implementado se utilizó una red neuronal de múltiples capas ocultas.

- Para determinar la dinámica del proceso de nivel de la planta Festo mediante el entrenamiento de una red neuronal feed-forward se aplicaron diferentes valores de setpoint a la planta y se observó la respuesta a dicho setpoint. Para el entrenamiento y testeo de la red neuronal se utilizó entradas de control anteriores ($u(k-1)$), las cuales son los diferentes setpoint aplicados a la planta y salidas de la planta anteriores ($y^s(k-1)$) que representan las respuestas a dichos setpoints, logrando así identificar correctamente el sistema dinámico del proceso de nivel de la planta FESTO MPS PA® Compact Workstation.

- La red neuronal profunda feed-forward utilizada para el entrenamiento e identificación del sistema dinámico de la planta de nivel tiene la siguiente estructura: 1 capa de entrada, 2 capas ocultas con 5 y 2 neuronas respectivamente y 1 capa de salida con una neurona; la función de activación utilizada para todas las neuronas fue la tangente hiperbólica y con esta estructura se alcanzó un correcto funcionamiento del sistema de control implementado.

- La red neuronal feed-forward de múltiples capas ocultas fue entrenada con las características antes mencionadas y con los datos de entrenamiento y testeo capturados, logrando alcanzar el objetivo de identificación del sistema dinámico de nivel de una sola entrada y una sola salida (SISO), verificando así que las redes neuronales artificiales constituyen una excelente herramienta para la identificación de sistemas dinámicos no lineales.

- En un proceso de nivel de la planta Festo Compact Workstation se logró la implementación de un sistema de control predictivo con identificación del sistema dinámico mediante la aplicación de redes neuronales, verificando así los beneficios que otorgan los controladores predictivos y las redes neuronales en el desarrollo de la teoría de control para la industria y la academia. La dinámica del sistema fue identificada mediante el entrenamiento de redes neuronales y la señal de control se la obtuvo mediante la optimización de la función de coste empleada en un determinado horizonte de control $N = 5$ y horizonte de predicción $P = 15$.

## 5.2. Recomendaciones.

- Se recomienda utilizar redes neuronales recurrentes para la identificación de la dinámica del sistema de nivel, ya que dotan de una "memoria interna" adecuadas para problemas de aprendizaje automático que involucran datos secuenciales.

- Se sugiere probar con tarjetas de adquisición de datos de mejores prestaciones técnicas que el Arduino-Uno para determinar si no se presentan errores en la lectura y escritura de datos.

- Se aconseja utilizar diferentes librerías para machine learning en el entrenamiento de la red neuronal, por ejemplo scikit-learn, Keras, etc. y comparar los resultados con los establecidos en éste trabajo.

- En posteriores investigaciones se recomienda, de ser posible, implementar el controlador predictivo con identificación de redes neuronales en una planta de condiciones industriales (alta potencia, condiciones ambientales extremas, ruidos, vibraciones,etc).

# 6. BIBLIOGRAFÍA


[1] B. M. Åkesson and H. T. Toivonen, "A neural network model predictive controller," *Journal of Process Control*, vol. 16, no. 9, pp. 937–946, 2006.

[2] A. Vasicˇkaninova and M. Bakosˇova, "Neural network predictive control of a chemical reactor," *Acta Chimica Slovaca*, vol. 2, no. 2, pp. 21–36, 2009.

[3] T. J. van den Boom*, M. A. Botto, and P. Hoekstra, "Design of an analytic constrained predictive controller using neural networks," *International Journal of Systems Science*, vol. 36, no. 10, pp. 639–650, 2005.

[4] A. Vasičkaninová, M. Bakošová, A. Mészáros, and J. J. Klemeš, "Neural network predictive control of a heat exchanger," *Applied Thermal Engineering*, vol. 31, no. 13, pp. 2094–2100, 2011.

[5] X. Yu-Geng, L. De-Wei, and L. Shu, "Model predictive control—status and challenges," *Acta Automatica Sinica*, vol. 39, no. 3, pp. 222–236, 2013.

[6] V. Rankovic, J. Radulovic, N. Grujovic, and D. Divac, "Neural network model predictive control of nonlinear systems using genetic algorithms," *International Journal of Computers Communications & Control*, vol. 7, no. 3, pp. 540–549, 2014.

[7] M. Maiworm, T. Bäthge, and R. Findeisen, "Scenario-based model predictive control: Recursive feasibility and stability," *IFAC-PapersOnLine*, vol. 48, no. 8, pp. 50–56, 2015.

[8] F. Smarra, A. Jain, T. de Rubeis, D. Ambrosini, A. D'Innocenzo, and R. Mangharam, "Data-driven model predictive control using random forests for building energy optimization and climate control," *Applied energy*, vol. 226, pp. 1252–1272, 2018.

[9] D. Honc, R. Sharma, A. Abraham, F. Dušek, and N. Pappa, "Teaching and practicing model predictive control," *IFAC-PapersOnLine*, vol. 49, no. 6, pp. 34–39, 2016.

[10] M. Bakošová and J. Oravec, "Robust model predictive control for heat exchanger network," *Applied Thermal Engineering*, vol. 73, no. 1, pp. 924–930, 2014.

[11] E. Entchev, L. Yang, M. Ghorab, A. Rosato, and S. Sibilio, "Energy, economic and environmental performance simulation of a hybrid renewable microgeneration system with neural network predictive control," *Alexandria engineering journal*, vol. 57, no. 1, pp. 455–473, 2018.

[12] D.-H. Jung, H.-J. Kim, J. Y. Kim, T. S. Lee, and S. H. Park, "Model predictive control via output feedback neural network for improved multi-window greenhouse ventilation control," *Sensors*, vol. 20, no. 6, p. 1756, 2020.

[13] Z. Tian, S. Li, and Y. Wang, "Ts fuzzy neural network predictive control for burning zone temperature in rotary kiln with improved hierarchical genetic algorithm," *International Journal of Modelling, Identification and Control*, vol. 25, no. 4, pp. 323–334, 2016.

[14] M. Elsisi, "Design of neural network predictive controller based on imperialist competitive algorithm for automatic voltage regulator," *Neural Computing and Applications*, vol. 31, no. 9, pp. 5017–5027, 2019.

[15] M. Heidari, "Improving efficiency of photovoltaic system by using neural network mppt and predictive control of converter," *Int J Renew Energy Res*, vol. 6, no. 4, pp. 1524–1529, 2016.

[16] D. Karayel, "Prediction and control of surface roughness in cnc lathe using artificial neural network," *Journal of materials processing technology*, vol. 209, no. 7, pp. 3125–3137, 2009.

[17] J.-H. Shin, H.-B. Jun, and J.-G. Kim, "Dynamic control of intelligent parking guidance using neural network predictive control," *Computers & Industrial Engineering*, vol. 120, pp. 15–30, 2018.

[18] N. Sharma and K. Singh, "Model predictive control and neural network predictive control of tame reactive distillation column," *Chemical Engineering and Processing: Process Intensification*, vol. 59, pp. 9–21, 2012.

[19] A. A. Aldair, "Hardware implementation of the neural network predictive controller for coupled tank system," *American Journal of Electrical and Electronic Engineering*, vol. 2, no. 1, pp. 40–47, 2014.

[20] R. H. Gaviño, *Introducción a los Sistemas de Control: Conceptos, Aplicaciones y Simulación con MATLAB*. Prentice Hall, 2010.

[21] S. C. Chapra, *Metodos Numericos Para Engenharia*. McGraw-Hill, 2008.

[22] F. Berzal, *Redes Neuronales Deep Learning*. EUG, Editorial Universidad de Granada, 1st ed., 2018.

[23] C. B. E. F. Camacho, *Model Predictive Control*. Advanced textbooks in control and signal processing, Springer, 1999.

[24] FESTO, "Pneumatic electric automation technology festo usa." Available at `https://www.festo-didactic.com/`.

# 7. ANEXOS

**Variables del controlador**

$\mathbf{N}_P$.- representa el Horizonte de predicción.

$\mathbf{N}_u$.- representa el Horizonte de control.

$\delta$.- representa el factor de ponderación para seguimiento trayectoria

$\lambda$.- representa el factor de ponderación para incrementos de control

$\mathbf{G}_{N_{P,u}}$.- representa la matriz de predicción del DMC con horizontes (NP ;Nu)

**f**.- representa el vector de respuesta libre

**u**.- representa el vector de incrementos futuros de control

**w**.- representa el vector de futuras referencias

**y**.- representa el vector de salidas para el horizonte de predicción

**Abreviaturas**

**ANNMPC**.- Adaptive Neural Network Model Predictive Control, Control Predictivo Adaptativo con Red Neuronal de Modelo.

**AVR**.- Automatic Voltage Regulator, Regulador de Voltaje Automático.

**ICA**.- Imperialist Competitive Algorithm, Algoritmo Competitivo Imperialista.

**LMPC**.- Linear Model Predictive Control, Controlador Predictivo Lineal de Modelo.

**MPC**.- Model Predictive Control, Controlador Predictivo de Modelo.

**MPBC**.- Controlador Predictivo basado en Modelo.

**NMPC**.- Nonlinear Model Predictive Control, Controlador Predictivo No Lineal de Modelo.

**NN**.- Neural Network, Red Neuronal.

**NNPC**.- Neural Network Predictive Control, Control Predictivo de una Redes Neuronales.

**OFNN**.- Output Feedback Neural-Network, Predicción de una Red Neuronal de Retroalimentación.

**RMPC**.- Robust Model Predictive Control, Control Robusto Predictivo de Modelo.

**P**.-Controlador proporcional.

**PI**.-Controlador proporcional integral.

**PID**.-Controlador proporcional integral derivativo.

| g1 | g2 | g3 | g4 | g5 | g6 | g7 | g8 | g9 | g10 |
|---|---|---|---|---|---|---|---|---|---|
| 0.0000 | 0.0061 | 0.0121 | 0.0181 | 0.0240 | 0.0298 | 0.0356 | 0.0414 | 0.0471 | 0.0527 |
| g11 | g12 | g13 | g14 | g15 | g16 | g17 | g18 | g19 | g20 |
| 0.0583 | 0.0639 | 0.0694 | 0.0748 | 0.0802 | 0.0855 | 0.0908 | 0.0961 | 0.1013 | 0.1065 |
| g21 | g22 | g23 | g24 | g25 | g26 | g27 | g28 | g29 | g30 |
| 0.1116 | 0.1166 | 0.1217 | 0.1266 | 0.1316 | 0.1364 | 0.1413 | 0.1461 | 0.1508 | 0.1555 |
| g31 | g32 | g33 | g34 | g35 | g36 | g37 | g38 | g39 | g40 |
| 0.1602 | 0.1648 | 0.1694 | 0.1740 | 0.1785 | 0.1829 | 0.1873 | 0.1917 | 0.1961 | 0.2004 |
| g41 | g42 | g43 | g44 | g45 | g46 | g47 | g48 | g49 | g50 |
| 0.2046 | 0.2088 | 0.2130 | 0.2172 | 0.2213 | 0.2254 | 0.2294 | 0.2334 | 0.2374 | 0.2413 |
| g51 | g52 | g53 | g54 | g55 | g56 | g57 | g58 | g59 | g60 |
| 0.2452 | 0.2490 | 0.2528 | 0.2566 | 0.2604 | 0.2641 | 0.2678 | 0.2714 | 0.2751 | 0.2786 |
| g61 | g62 | g63 | g64 | g65 | g66 | g67 | g68 | g69 | g70 |
| 0.2822 | 0.2857 | 0.2892 | 0.2927 | 0.2961 | 0.2995 | 0.3029 | 0.3062 | 0.3095 | 0.3128 |
| g71 | g72 | g73 | g74 | g75 | g76 | g77 | g78 | g79 | g80 |
| 0.3160 | 0.3192 | 0.3224 | 0.3256 | 0.3287 | 0.3318 | 0.3349 | 0.3379 | 0.3410 | 0.3439 |
| g81 | g82 | g83 | g84 | g85 | g86 | g87 | g88 | g89 | g90 |
| 0.3469 | 0.3498 | 0.3528 | 0.3556 | 0.3585 | 0.3613 | 0.3641 | 0.3669 | 0.3697 | 0.3724 |
| g91 | g92 | g93 | g94 | g95 | g96 | g97 | g98 | g99 | g100 |
| 0.3751 | 0.3778 | 0.3804 | 0.3831 | 0.3857 | 0.3883 | 0.3908 | 0.3934 | 0.3959 | 0.3984 |
| g101 | g102 | g103 | g104 | g105 | g106 | g107 | g108 | g109 | g110 |
| 0.4009 | 0.4033 | 0.4057 | 0.4081 | 0.4105 | 0.4129 | 0.4152 | 0.4176 | 0.4198 | 0.4221 |
| g111 | g112 | g113 | g114 | g115 | g116 | g117 | g118 | g119 | g120 |
| 0.4244 | 0.4266 | 0.4288 | 0.4310 | 0.4332 | 0.4354 | 0.4375 | 0.4396 | 0.4417 | 0.4438 |
| g121 | g122 | g123 | g124 | g125 | g126 | g127 | g128 | g129 | g130 |
| 0.4459 | 0.4479 | 0.4499 | 0.4519 | 0.4539 | 0.4559 | 0.4578 | 0.4598 | 0.4617 | 0.4636 |
| g131 | g132 | g133 | g134 | g135 | g136 | g137 | g138 | g139 | g140 |
| 0.4655 | 0.4673 | 0.4692 | 0.4710 | 0.4728 | 0.4746 | 0.4764 | 0.4782 | 0.4799 | 0.4817 |
| g141 | g142 | g143 | g144 | g145 | g146 | g147 | g148 | g149 | g150 |
| 0.4834 | 0.4851 | 0.4868 | 0.4884 | 0.4901 | 0.4917 | 0.4934 | 0.4950 | 0.4966 | 0.4982 |
| g151 | g152 | g153 | g154 | g155 | g156 | g157 | g158 | g159 | g160 |
| 0.4997 | 0.5013 | 0.5028 | 0.5044 | 0.5059 | 0.5074 | 0.5089 | 0.5103 | 0.5118 | 0.5132 |
| g161 | g162 | g163 | g164 | g165 | g166 | g167 | g168 | g169 | g170 |
| 0.5147 | 0.5161 | 0.5175 | 0.5189 | 0.5203 | 0.5217 | 0.5230 | 0.5244 | 0.5257 | 0.5270 |
| g171 | g172 | g173 | g174 | g175 | g176 | g177 | g178 | g179 | g180 |
| 0.5283 | 0.5296 | 0.5309 | 0.5322 | 0.5334 | 0.5347 | 0.5359 | 0.5372 | 0.5384 | 0.5396 |
| g181 | g182 | g183 | g184 | g185 | g186 | g187 | g188 | g189 | g190 |
| 0.5408 | 0.5420 | 0.5431 | 0.5443 | 0.5454 | 0.5466 | 0.5477 | 0.5488 | 0.5500 | 0.5511 |
| g191 | g192 | g193 | g194 | g195 | g196 | g197 | g198 | g199 | g200 |
| 0.5521 | 0.5532 | 0.5543 | 0.5554 | 0.5564 | 0.5575 | 0.5585 | 0.5595 | 0.5605 | 0.5615 |

Figura 44: Valores de respuesta al escalón unitario.

Fuente: Elaborado por la autora.

| g201 | g202 | g203 | g204 | g205 | g206 | g207 | g208 | g209 | g210 |
|---|---|---|---|---|---|---|---|---|---|
| 0.5625 | 0.5635 | 0.5645 | 0.5655 | 0.5664 | 0.5674 | 0.5683 | 0.5693 | 0.5702 | 0.5711 |
| g211 | g212 | g213 | g214 | g215 | g216 | g217 | g218 | g219 | g220 |
| 0.5720 | 0.5729 | 0.5738 | 0.5747 | 0.5756 | 0.5764 | 0.5773 | 0.5782 | 0.5790 | 0.5798 |
| g221 | g222 | g223 | g224 | g225 | g226 | g227 | g228 | g229 | g230 |
| 0.5807 | 0.5815 | 0.5823 | 0.5831 | 0.5839 | 0.5847 | 0.5855 | 0.5863 | 0.5871 | 0.5878 |
| g231 | g232 | g233 | g234 | g235 | g236 | g237 | g238 | g239 | g240 |
| 0.5886 | 0.5893 | 0.5901 | 0.5908 | 0.5916 | 0.5923 | 0.5930 | 0.5937 | 0.5944 | 0.5951 |
| g241 | g242 | g243 | g244 | g245 | g246 | g247 | g248 | g249 | g250 |
| 0.5958 | 0.5965 | 0.5972 | 0.5979 | 0.5985 | 0.5992 | 0.5998 | 0.6005 | 0.6011 | 0.6018 |
| g251 | g252 | g253 | g254 | g255 | g256 | g257 | g258 | g259 | g260 |
| 0.6024 | 0.6030 | 0.6037 | 0.6043 | 0.6049 | 0.6055 | 0.6061 | 0.6067 | 0.6073 | 0.6079 |
| g261 | g262 | g263 | g264 | g265 | g266 | g267 | g268 | g269 | g270 |
| 0.6084 | 0.6090 | 0.6096 | 0.6101 | 0.6107 | 0.6112 | 0.6118 | 0.6123 | 0.6129 | 0.6134 |
| g271 | g272 | g273 | g274 | g275 | g276 | g277 | g278 | g279 | g280 |
| 0.6139 | 0.6145 | 0.6150 | 0.6155 | 0.6160 | 0.6165 | 0.6170 | 0.6175 | 0.6180 | 0.6185 |
| g281 | g282 | g283 | g284 | g285 | g286 | g287 | g288 | g289 | g290 |
| 0.6190 | 0.6194 | 0.6199 | 0.6204 | 0.6208 | 0.6213 | 0.6218 | 0.6222 | 0.6227 | 0.6231 |
| g291 | g292 | g293 | g294 | g295 | g296 | g297 | g298 | g299 | g300 |
| 0.6235 | 0.6240 | 0.6244 | 0.6248 | 0.6253 | 0.6257 | 0.6261 | 0.6265 | 0.6269 | 0.6273 |
| g301 | g302 | g303 | g304 | g305 | g306 | g307 | g308 | g309 | g310 |
| 0.6277 | 0.6281 | 0.6285 | 0.6289 | 0.6293 | 0.6297 | 0.6301 | 0.6304 | 0.6308 | 0.6312 |
| g311 | g312 | g313 | g314 | g315 | g316 | g317 | g318 | g319 | g320 |
| 0.6316 | 0.6319 | 0.6323 | 0.6326 | 0.6330 | 0.6333 | 0.6337 | 0.6340 | 0.6344 | 0.6347 |
| g321 | g322 | g323 | g324 | g325 | g326 | g327 | g328 | g329 | g330 |
| 0.6351 | 0.6354 | 0.6357 | 0.6360 | 0.6364 | 0.6367 | 0.6370 | 0.6373 | 0.6376 | 0.6379 |
| g331 | g332 | g333 | g334 | g335 | g336 | g337 | g338 | g339 | g340 |
| 0.6382 | 0.6385 | 0.6388 | 0.6391 | 0.6394 | 0.6397 | 0.6400 | 0.6403 | 0.6406 | 0.6409 |
| g341 | g342 | g343 | g344 | g345 | g346 | g347 | g348 | g349 | g350 |
| 0.6412 | 0.6414 | 0.6417 | 0.6420 | 0.6423 | 0.6425 | 0.6428 | 0.6430 | 0.6433 | 0.6436 |
| g351 | g352 | g353 | g354 | g355 | g356 | g357 | g358 | g359 | g360 |
| 0.6438 | 0.6441 | 0.6443 | 0.6446 | 0.6448 | 0.6451 | 0.6453 | 0.6455 | 0.6458 | 0.6460 |
| g361 | g362 | g363 | g364 | g365 | g366 | g367 | g368 | g369 | g370 |
| 0.6462 | 0.6465 | 0.6467 | 0.6469 | 0.6472 | 0.6474 | 0.6476 | 0.6478 | 0.6480 | 0.6483 |
| g371 | g372 | g373 | g374 | g375 | g376 | g377 | g378 | g379 | g380 |
| 0.6485 | 0.6487 | 0.6489 | 0.6491 | 0.6493 | 0.6495 | 0.6497 | 0.6499 | 0.6501 | 0.6503 |
| g381 | g382 | g383 | g384 | g385 | g386 | g387 | g388 | g389 | g390 |
| 0.6505 | 0.6507 | 0.6509 | 0.6511 | 0.6513 | 0.6514 | 0.6516 | 0.6518 | 0.6520 | 0.6522 |
| g391 | g392 | g393 | g394 | g395 | g396 | g397 | g398 | g399 | g400 |
| 0.6523 | 0.6525 | 0.6527 | 0.6529 | 0.6530 | 0.6532 | 0.6534 | 0.6535 | 0.6537 | 0.6539 |

Figura 45: Valores de respuesta al escalón unitario.

Fuente: Elaborado por la autora.

| g401 | g402 | g403 | g404 | g405 | g406 | g407 | g408 | g409 | g410 |
|---|---|---|---|---|---|---|---|---|---|
| 0.6540 | 0.6542 | 0.6544 | 0.6545 | 0.6547 | 0.6548 | 0.6550 | 0.6551 | 0.6553 | 0.6554 |
| g411 | g412 | g413 | g414 | g415 | g416 | g417 | g418 | g419 | g420 |
| 0.6556 | 0.6557 | 0.6559 | 0.6560 | 0.6562 | 0.6563 | 0.6564 | 0.6566 | 0.6567 | 0.6568 |
| g421 | g422 | g423 | g424 | g425 | g426 | g427 | g428 | g429 | g430 |
| 0.6570 | 0.6571 | 0.6572 | 0.6574 | 0.6575 | 0.6576 | 0.6578 | 0.6579 | 0.6580 | 0.6581 |
| g431 | g432 | g433 | g434 | g435 | g436 | g437 | g438 | g439 | g440 |
| 0.6583 | 0.6584 | 0.6585 | 0.6586 | 0.6588 | 0.6589 | 0.6590 | 0.6591 | 0.6592 | 0.6593 |
| g441 | g442 | g443 | g444 | g445 | g446 | g447 | g448 | g449 | g450 |
| 0.6594 | 0.6596 | 0.6597 | 0.6598 | 0.6599 | 0.6600 | 0.6601 | 0.6602 | 0.6603 | 0.6604 |
| g451 | g452 | g453 | g454 | g455 | g456 | g457 | g458 | g459 | g460 |
| 0.6605 | 0.6606 | 0.6607 | 0.6608 | 0.6609 | 0.6610 | 0.6611 | 0.6612 | 0.6613 | 0.6614 |
| g461 | g462 | g463 | g464 | g465 | g466 | g467 | g468 | g469 | g470 |
| 0.6615 | 0.6616 | 0.6617 | 0.6618 | 0.6619 | 0.6620 | 0.6620 | 0.6621 | 0.6622 | 0.6623 |
| g471 | g472 | g473 | g474 | g475 | g476 | g477 | g478 | g479 | g480 |
| 0.6624 | 0.6625 | 0.6626 | 0.6626 | 0.6627 | 0.6628 | 0.6629 | 0.6630 | 0.6631 | 0.6631 |
| g481 | g482 | g483 | g484 | g485 | g486 | g487 | g488 | g489 | g490 |
| 0.6632 | 0.6633 | 0.6634 | 0.6634 | 0.6635 | 0.6636 | 0.6637 | 0.6637 | 0.6638 | 0.6639 |
| g491 | g492 | g493 | g494 | g495 | g496 | g497 | g498 | g499 | g500 |
| 0.6640 | 0.6640 | 0.6641 | 0.6642 | 0.6642 | 0.6643 | 0.6644 | 0.6644 | 0.6645 | 0.6646 |
| g501 | g502 | g503 | g504 | g505 | g506 | g507 | g508 | g509 | g510 |
| 0.6646 | 0.6647 | 0.6648 | 0.6648 | 0.6649 | 0.6650 | 0.6650 | 0.6651 | 0.6651 | 0.6652 |
| g511 | g512 | g513 | g514 | g515 | g516 | g517 | g518 | g519 | g520 |
| 0.6653 | 0.6653 | 0.6654 | 0.6654 | 0.6655 | 0.6655 | 0.6656 | 0.6657 | 0.6657 | 0.6658 |
| g521 | g522 | g523 | g524 | g525 | g526 | g527 | g528 | g529 | g530 |
| 0.6658 | 0.6659 | 0.6659 | 0.6660 | 0.6660 | 0.6661 | 0.6661 | 0.6662 | 0.6662 | 0.6663 |
| g531 | g532 | g533 | g534 | g535 | g536 | g537 | g538 | g539 | g540 |
| 0.6663 | 0.6664 | 0.6664 | 0.6665 | 0.6665 | 0.6666 | 0.6666 | 0.6667 | 0.6667 | 0.6668 |
| g541 | g542 | g543 | g544 | g545 | g546 | g547 | g548 | g549 | g550 |
| 0.6668 | 0.6669 | 0.6669 | 0.6670 | 0.6670 | 0.6670 | 0.6671 | 0.6671 | 0.6672 | 0.6672 |
| g551 | g552 | g553 | g554 | g555 | g556 | g557 | g558 | g559 | g560 |
| 0.6673 | 0.6673 | 0.6673 | 0.6674 | 0.6674 | 0.6675 | 0.6675 | 0.6675 | 0.6676 | 0.6676 |
| g561 | g562 | g563 | g564 | g565 | g566 | g567 | g568 | g569 | g570 |
| 0.6676 | 0.6677 | 0.6677 | 0.6678 | 0.6678 | 0.6678 | 0.6679 | 0.6679 | 0.6679 | 0.6680 |
| g571 | g572 | g573 | g574 | g575 | g576 | g577 | g578 | g579 | g580 |
| 0.6680 | 0.6680 | 0.6681 | 0.6681 | 0.6681 | 0.6682 | 0.6682 | 0.6682 | 0.6683 | 0.6683 |
| g581 | g582 | g583 | g584 | g585 | g586 | g587 | g588 | g589 | g590 |
| 0.6683 | 0.6684 | 0.6684 | 0.6684 | 0.6685 | 0.6685 | 0.6685 | 0.6686 | 0.6686 | 0.6686 |
| g591 | g592 | g593 | g594 | g595 | g596 | g597 | g598 | g599 | g600 |
| 0.6686 | 0.6687 | 0.6687 | 0.6687 | 0.6688 | 0.6688 | 0.6688 | 0.6688 | 0.6689 | 0.6689 |

Figura 46: Valores de respuesta al escalón unitario.

Fuente: Elaborado por la autora.

| g601 | g602 | g603 | g604 | g605 | g606 | g607 | g608 | g609 | g610 |
|---|---|---|---|---|---|---|---|---|---|
| 0.6689 | 0.6689 | 0.6690 | 0.6690 | 0.6690 | 0.6690 | 0.6691 | 0.6691 | 0.6691 | 0.6691 |
| g611 | g612 | g613 | g614 | g615 | g616 | g617 | g618 | g619 | g620 |
| 0.6692 | 0.6692 | 0.6692 | 0.6692 | 0.6693 | 0.6693 | 0.6693 | 0.6693 | 0.6693 | 0.6694 |
| g621 | g622 | g623 | g624 | g625 | g626 | g627 | g628 | g629 | g630 |
| 0.6694 | 0.6694 | 0.6694 | 0.6695 | 0.6695 | 0.6695 | 0.6695 | 0.6695 | 0.6696 | 0.6696 |
| g631 | g632 | g633 | g634 | g635 | g636 | g637 | g638 | g639 | g640 |
| 0.6696 | 0.6696 | 0.6696 | 0.6697 | 0.6697 | 0.6697 | 0.6697 | 0.6697 | 0.6698 | 0.6698 |
| g641 | g642 | g643 | g644 | g645 | g646 | g647 | g648 | g649 | g650 |
| 0.6698 | 0.6698 | 0.6698 | 0.6698 | 0.6699 | 0.6699 | 0.6699 | 0.6699 | 0.6699 | 0.6700 |
| g651 | g652 | g653 | g654 | g655 | g656 | g657 | g658 | g659 | g660 |
| 0.6700 | 0.6700 | 0.6700 | 0.6700 | 0.6700 | 0.6701 | 0.6701 | 0.6701 | 0.6701 | 0.6701 |
| g661 | g662 | g663 | g664 | g665 | g666 | g667 | g668 | g669 | g670 |
| 0.6701 | 0.6701 | 0.6702 | 0.6702 | 0.6702 | 0.6702 | 0.6702 | 0.6702 | 0.6702 | 0.6703 |
| g671 | g672 | g673 | g674 | g675 | g676 | g677 | g678 | g679 | g680 |
| 0.6703 | 0.6703 | 0.6703 | 0.6703 | 0.6703 | 0.6703 | 0.6704 | 0.6704 | 0.6704 | 0.6704 |
| g681 | g682 | g683 | g684 | g685 | g686 | g687 | g688 | g689 | g690 |
| 0.6704 | 0.6704 | 0.6704 | 0.6704 | 0.6705 | 0.6705 | 0.6705 | 0.6705 | 0.6705 | 0.6705 |
| g691 | g692 | g693 | g694 | g695 | g696 | g697 | g698 | g699 | g700 |
| 0.6705 | 0.6705 | 0.6706 | 0.6706 | 0.6706 | 0.6706 | 0.6706 | 0.6706 | 0.6706 | 0.6706 |
| g701 | g702 | g703 | g704 | g705 | g706 | g707 | g708 | g709 | g710 |
| 0.6706 | 0.6706 | 0.6707 | 0.6707 | 0.6707 | 0.6707 | 0.6707 | 0.6707 | 0.6707 | 0.6707 |
| g711 | g712 | g713 | g714 | g715 | g716 | g717 | g718 | g719 | g720 |
| 0.6707 | 0.6707 | 0.6708 | 0.6708 | 0.6708 | 0.6708 | 0.6708 | 0.6708 | 0.6708 | 0.6708 |
| g721 | g722 | g723 | g724 | g725 | g726 | g727 | g728 | g729 | g730 |
| 0.6708 | 0.6708 | 0.6709 | 0.6709 | 0.6709 | 0.6709 | 0.6709 | 0.6709 | 0.6709 | 0.6709 |
| g731 | g732 | g733 | g734 | g735 | g736 | g737 | g738 | g739 | g740 |
| 0.6709 | 0.6709 | 0.6709 | 0.6709 | 0.6709 | 0.6710 | 0.6710 | 0.6710 | 0.6710 | 0.6710 |
| g741 | g742 | g743 | g744 | g745 | g746 | g747 | g748 | g749 | g750 |
| 0.6710 | 0.6710 | 0.6710 | 0.6710 | 0.6710 | 0.6710 | 0.6710 | 0.6710 | 0.6711 | 0.6711 |

Figura 47: Valores de respuesta al escalón unitario.

Fuente: Elaborado por la autora.

**Función controlador predictivo**

```
data_controlX = [DATOS_PREDIC(:,2)';DATOS_PREDIC(:,3)'];
data_controlY = [DATOS_PREDIC(:,3)'];

##data_controlX = [DATOS_PREDIC(1:381,2)';DATOS_PREDIC(1:381,3)'];
##data_controlY = [DATOS_PREDIC(1:381,3)'];

dataX = (data_controlX-min(data_controlX))./(max(data_controlX)-min(data_contro

trainedModel = load('Trained_control_nivel.mat');

W = trainedModel.W;
transferFunctions = trainedModel.transferFunctions;
options = trainedModel.options;


numLayers = length(transferFunctions);
reluThresh = options.reluThresh;

N = size(dataX, 1);
vectorOnes = ones(N,1);
A = cell(1, numLayers);
Z = cell(1, numLayers);
Z{1} = dataX;
A{1} = [vectorOnes Z{1}];
```

```
for i = 2:numLayers
    Z{i} = A{i - 1}*W{i - 1}';
%     % Transfer function values

    switch transferFunctions{i}
        case 'logsig'
            A{i} = [vectorOnes logsig(Z{i})];
        case 'relu'
            A{i} = [vectorOnes bsxfun(@max, Z{i}, reluThresh)];
        case 'tanh'
            A{i} = [vectorOnes tansig_aux(Z{i})];
        case 'purelin'
            A{i} = [vectorOnes Z{i}];
        case 'softplus'
            A{i} = [vectorOnes log(exp(Z{i}) + 1)];
        case 'elu'
            A{i} = [vectorOnes elu(Z{i})];
        otherwise
            error('Invalid transfer function. Valid options are elu, softplus, relu, logsig, tanh, and purelin');
    end
end
```

```
dataYTestHat_net = A{end}(:,2);
dataYTestHat_net_Normal = dataYTestHat_net*(max(data_controlY)-min(data_controlY))+min(data_controlY);
% GRAFICOS DE TEST
figure(1);
grid on
hold all;
plot(DATOS_PREDIC(:,1), DATOS_PREDIC(:,2) , '-r','LineWidth', 1.5);
plot(DATOS_PREDIC(:,1), DATOS_PREDIC(:,3), '-b','LineWidth', 1.5);
plot(DATOS_PREDIC(:,1), dataYTestHat_net_Normal, '-g','LineWidth', 1.5);
hold off;
xlabel('Tiempo');
ylabel('Nivel');
legend('Datos de testeo', 'Datos de testeo', 'Predicion');
```

**Programa**

```
data_controX = [DATOS(:,2)';DATOS(:,3)'];
data_controlY = [DATOS(:,3)'];


% DATOS DE ENTRENAMIENTO
dataX = [(data_controX(1,:)-min(data_controX(1,:)))/(max(data_controX(1,:))-min(data_controX(1,:)));...
        (data_controX(2,:)-min(data_controX(2,:)))/(max(data_controX(2,:))-min(data_controX(2,:)))];
dataY = (data_controlY-min(data_controlY))/(max(data_controlY)-min(data_controlY));

[R] = corrcoef([dataX',dataY']);

dataXTrain = dataX';
dataYTrain = dataY';

% REGRESION USANDO RNA FEED FORWARD
% ARQUITECTURA DE LA RED NEURONAL
n = size(dataXTrain,2);
numNeuronsLayers = [n 5 2 1];
transferFunctions{1} = 'none';
transferFunctions{2} = 'tanh';
transferFunctions{3} = 'tanh';
transferFunctions{4} = 'tanh';
```

```
options.lambda = 10;
options.reluThresh = 1e-2;

% ENTRENAMIENTO DE LA RED NEURONAL
options.numIterations = 200;
W = ffnnRegTrain(dataXTrain, dataYTrain, numNeuronsLayers, transferFunctions, options);
```

```
% % ERROR DE ENTRENAMIENYO
% Etrain_net = (1/(2*size(dataXTrain,1)))*sum((dataYTest - dataYTestHat_net).^2);

% TESTEO DE LA RED NEURONAL CON TRAIN
[Z, A] = ffnnRegTest(dataXTrain, W, transferFunctions, options);
dataYTrainHat_net = A{end}(:,2);


%Impresion de resulatdos
fprintf('Coeficiente de correlacion = %1.2f \n',R(1,2));
% fprintf('Error de testeo regresion lineal= %3.2f \n',Etest_lin);
% fprintf('Error de testeo de la NRA = %3.2f \n',Etest_net);
% fprintf('Error de bayes = %3.2f \n',EBayes);
```

```
% DATOS PREDICHOS DEL TRAIN RED NEURONAL NORMALIZADOS
dataYTrainHat_net_Normal = dataYTrainHat_net*(max(data_controlY)-min(data_controlY))+min(data_controlY);


% GRAFICOS DE TRAIN
figure(3);
grid on
hold all;
plot(DATOS(:,1), DATOS(:,2) , '-r','LineWidth', 1.5);
plot(DATOS(:,1), DATOS(:,3), '-b','LineWidth', 1.5);
plot(DATOS(:,1), dataYTrainHat_net_Normal, '-g','LineWidth', 1.5);
hold off;
xlabel('Tiempo');
ylabel('Nivel');
legend('Datos de entrenamiento', 'Datos de entrenamiento', 'Prediccion');
save('Trained_control_nivel.mat', 'W', 'transferFunctions', 'options');
```