



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA ELECTROMECAÁNICA

**TRABAJO DE TITULACIÓN, PREVIO A LA OBTENCIÓN DEL TÍTULO
DE INGENIERO ELECTROMECAÁNICO**

**TEMA: CARACTERIZACIÓN Y EVALUACIÓN DE CONTROLADORES
PID, LÓGICA DIFUSA Y REDES NEURONALES DE SISTEMAS
DISTRIBUIDOS INDUSTRIALES EN EL LABORATORIO DE HIDRÓNICA
Y NEURÓNICA DE LA UNIVERSIDAD DE LAS FUERZAS ARMADAS
ESPE LATACUNGA**

AUTOR: PACHECO TACO, DIEGO ERNESTO

DIRECTOR: ING. SÁNCHEZ OCAÑA, WILSON EDMUNDO

LATACUNGA

2019



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA
CARRERA DE INGENIERÍA ELECTROMECAÁNICA

CERTIFICACIÓN

Certifico que el trabajo de titulación, “**CARACTERIZACIÓN Y EVALUACIÓN DE CONTROLADORES PID, LÓGICA DIFUSA Y REDES NEURONALES DE SISTEMAS DISTRIBUIDOS INDUSTRIALES EN EL LABORATORIO DE HIDRÓNICA Y NEURÓNICA DE LA UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE LATACUNGA**” fue realizado por el señor **Pacheco Taco, Diego Ernesto**, el mismo que ha sido revisado en su totalidad, analizado por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Latacunga, 25 de enero de 2019

A handwritten signature in blue ink, which appears to read 'Wilson Sánchez Ocaña', is positioned above the printed name of the director.

Ing. Wilson Sánchez Ocaña
C.C. 0501529937
DIRECTOR



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA EN ELECTROMECAÁNICA

AUTORÍA DE RESPONSABILIDAD

Yo, **Pacheco Taco, Diego Ernesto**, declaro que el contenido, ideas y criterios del trabajo de titulación: **“CARACTERIZACIÓN Y EVALUACIÓN DE CONTROLADORES PID, LÓGICA DIFUSA Y REDES NEURONALES DE SISTEMAS DISTRIBUIDOS INDUSTRIALES EN EL LABORATORIO DE HIDRÓNICA Y NEURÓNICA DE LA UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE LATACUNGA”**, es de mi autoría y responsabilidad, cumpliendo con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Consecuentemente el contenido de la investigación mencionada es veraz.

Latacunga, 25 de enero de 2019

A handwritten signature in blue ink, which appears to read 'Diego Pacheco', is written over a horizontal line.

Diego Ernesto Pacheco Taco
C.C.: 050347172-4



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA EN ELECTROMECAÁNICA

AUTORIZACIÓN

Yo, **Pacheco Taco, Diego Ernesto**, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **“CARACTERIZACIÓN Y EVALUACIÓN DE CONTROLADORES PID, LÓGICA DIFUSA Y REDES NEURONALES DE SISTEMAS DISTRIBUIDOS INDUSTRIALES EN EL LABORATORIO DE HIDRÓNICA Y NEURÓNICA DE LA UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE LATACUNGA”**, en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Latacunga, 25 de enero de 2019

A handwritten signature in blue ink, which appears to read 'Diego Pacheco', is positioned above the printed name of the author.

Diego Ernesto Pacheco Taco
C.C.: 050347172-4

DEDICATORIA

Quiero dedicar este trabajo de culminación de mi vida universitaria primeramente a Dios, por haberme brindado la vida y la salud necesaria para completar esta etapa de mi vida; luego a mis padres, Wilian y Francisca, gracias a los cuales con su esfuerzo, sacrificios y enseñanzas hoy soy un profesional de calidad, y en especial a mi madre, la cual siempre nos ha dejado claro que la mayor riqueza que ella nos dejará a mí y mis hermanas, es la educación. A mis queridas hermanas Vero y Diana, que siempre han sido como unas madres en mi vida y las cuales han aportado significativamente en mi formación humana como académica, y sin las cuales no podría haber finalizado este trabajo.

Diego

AGRADECIMIENTO

Siempre a Dios, por la vida, no solamente mía, también de mis padres, hermanas, sobrinos y demás familiares que han aportado en mi formación académica. A mis padres, pues sin ellos no habría podido conseguir este logro en mi vida, ya que cada uno fue indispensable a su manera: a Wilian por ser un padre que ha cuidado siempre a su familia, y a Francisca por ser la mejor educadora en todos los aspectos de mi vida. A Diana y Vero, mis hermanas, que siempre han estado al pendiente de mi formación y han contribuido totalmente en mi vida. A mi querido “Papato” Ernesto, y a mis tíos Fernando y Yolanda, cada uno de las palabras y consejos que siempre buscaron mi crecimiento profesional. Además, a Karla Magaly, quien fue un gran apoyo en mi vida universitaria y personal, y con quien siempre estaré agradecido por todo lo que ha hecho por mí.

Agradezco a la Universidad de las Fuerzas Armadas ESPE, en cuyas aulas pude formarme como un profesional que pondrá en alto el nombre de tan prestigiosa institución; también al Ing. Wilson Sánchez, que con su conocimiento y ayuda me permitieron desarrollar y culminar este trabajo.

Diego

ÍNDICE DE CONTENIDOS

CARATULA

CERTIFICACIÓN	i
AUTORÍA DE RESPONSABILIDAD.....	ii
AUTORIZACIÓN	iii
DEDICATORIA.....	iv
AGRADECIMIENTO	v
ÍNDICE DE CONTENIDOS.....	vi
ÍNDICE DE FIGURAS	xi
ÍNDICE DE TABLAS	xvi
RESUMEN.....	xvii
ABSTRACT.....	xviii

CAPITULO I

PROBLEMA

1.1. Antecedentes	1
1.2. Planteamiento del problema	2
1.3. Justificación e Importancia.....	3
1.4. Objetivos.....	4
1.4.1. Objetivo General	4
1.4.2. Objetivos Específicos.....	4

CAPÍTULO II

MARCO TEÓRICO

2.1. Introducción	5
2.2. Sistema de Control	5
2.2.1. Definiciones básicas	6
2.3. Tipos de Sistemas de Control	7
2.3.1. Sistemas de ciclo abierto	7
2.3.2. Sistemas de ciclo cerrado.....	8
2.4. Sistemas de Control Avanzado.....	9

2.4.1.	Control Experto:	10
2.4.2.	Control Óptimo.....	10
2.4.3.	Control Robusto	11
2.4.4.	Control Ajustable o Adaptable	11
2.4.5.	Control Neuronal.....	12
2.5.	Control PID	12
2.5.1.	Control Proporcional	13
2.5.2.	Control Integral	14
2.5.3.	Control Derivativo	16
2.5.4.	Funcionamiento	17
2.6.	Lógica Difusa	18
2.6.1.	Conjuntos difusos	20
2.6.2.	Operaciones lógicas	22
2.6.3.	Variable lingüística.....	24
2.6.4.	Funciones de membresía	25
2.6.5.	Inferencia difusa.....	26
2.6.6.	Defuzificación.....	27
2.6.7.	Controladores difusos	29
2.7.	Redes Neuronales	33
2.7.1.	Neurona Biológica.....	33
2.7.2.	Características de las redes neuronales artificiales	35
2.7.3.	Funcionamiento de la red neuronal.....	38
2.7.4.	Ventajas de utilizar RNA	39
2.7.5.	Entrenamiento de la red neuronal.....	40
2.8.	Comunicaciones Industriales	41
2.8.1.	Niveles en una Red Industrial	42
2.8.2.	Buses de Campo	43
2.8.3.	PROFIBUS	44
2.8.4.	Industrial Ethernet.....	45
2.8.5.	PROFINET	45

2.9.	Controladores Lógicos Programables (PLC)	46
2.9.1.	Procesador	47
2.9.2.	Memoria	48
2.9.3.	Entradas y salidas.....	48
2.9.4.	PLC S7-1200	49
2.9.5.	PLC S7-1500	50
2.10.	Matlab	51
2.10.1.	Características principales	51
2.10.2.	Simulink	52
2.11.	Hipótesis	53
2.12.	Variables de la investigación	53

CAPÍTULO III

METODOLOGÍA

3.1.	Antecedentes investigativos	54
3.2.	Modalidad de la Investigación.....	57
3.2.1.	Método Bibliográfico Documental	57
3.2.2.	Método Inductivo – Deductivo.....	57
3.2.3.	Método Experimental	58
3.3.	Tipo de investigación	58
3.3.1.	Investigación Cuantitativa	58

CAPÍTULO IV

ANÁLISIS RESULTADOS DE LA INVESTIGACIÓN

4.1.	Análisis de resultados	59
4.1.1.	Descripción de la planta industrial	59
4.1.2.	Sensores utilizados.....	61
4.1.3.	Actuador	63
4.1.4.	Variable manipulada	64
4.2.	Comprobación de hipótesis	65
4.2.1.	Planteamiento de hipótesis.....	65
4.2.2.	Viabilidad del proyecto	66

CAPÍTULO V

PROPUESTA

5.1.	Datos informativos	67
5.1.1.	Beneficiarios	67
5.2.	Objetivos.....	67
5.2.1.	Objetivo General.....	67
5.2.2.	Objetivos Específicos.....	68
5.3.	Fundamentación propuesta	68
5.4.	Diseño de la propuesta	69
5.5.	Toma de datos.....	70
5.6.	Generación del modelo matemático del proceso de nivel.....	73
5.7.	Desarrollo del controlador PID.....	77
5.7.1.	Pruebas de funcionamiento del bloque generado.....	81
5.8.	Desarrollo del controlador difuso	82
5.8.1.	Fuzzy Logic Toolbox	82
5.8.2.	Variables lingüísticas	84
5.8.3.	Reglas de inferencia	87
5.8.4.	Puesta a prueba del controlador difuso	90
5.9.	Desarrollo del controlador Neuronal	95
5.9.1.	Datos de entrenamiento para la red.....	95
5.9.2.	Entrenamiento del controlador	97
5.9.3.	Puesta a prueba del controlador neuronal	107
5.10.	Diseño de la red industrial	109
5.10.1.	Desarrollo del proyecto mediante TIA PORTAL	109
5.11.	Implementación de los controladores a la planta física.....	113
5.11.1.	Simulink® PLC Coder™	113
5.11.2.	Migración de los controladores hacia el PLC.....	114
5.12.	Resultados de la implementación	120
5.12.1.	Función STEPINFO	120
5.12.2.	Resultados de los controladores simulados.....	122

5.12.3. Resultados de los controladores implementados..... 136

CAPÍTULO VI

CONCLUSIONES Y RECOMENDACIONES

6.1. Conclusiones 150

6.2. Recomendaciones 151

Bibliografía..... 153

ÍNDICE DE FIGURAS

Figura 1. Componentes básicos de un sistema de control	5
Figura 2. Elementos de un sistema de control de ciclo abierto.....	8
Figura 3. Elementos de un sistema de control de ciclo cerrado.....	9
Figura 4. Diagrama de bloques del sistema de control PID	13
Figura 5. Acción proporcional	14
Figura 6. Acción Integral.....	15
Figura 7. Acción derivativa.....	16
Figura 8. Ejemplificaron de una inferencia difusa	19
Figura 9. Ejemplo de un conjunto clásico	20
Figura 10. Ejemplo de un conjunto difuso.....	21
Figura 11. Lógica Clásica versus Lógica Difusa.	22
Figura 12. Tablas de verdad estándar.	23
Figura 13. Operaciones lógicas en conjuntos difusos.....	24
Figura 14. Relación entre temperatura y la variable lingüística “caliente”	25
Figura 15. Función de membresía difusa.	25
Figura 16. Diferentes tipos de funciones de membresía.....	26
Figura 17. Proceso de defuzificación.....	28
Figura 18. Defuzificación acorde al método COG.....	29
Figura 19. Estructura Interna de un Controlador Difuso	30
Figura 20. Diagrama a bloques de un controlador difuso genérico.....	31
Figura 21. Procesamiento difuso de las variables en un controlador difuso	31
Figura 22. Elementos neurales	34
Figura 23. Modelo de una neurona artificial.....	35
Figura 24. Red Neuronal con varias capas.....	37
Figura 25. Funciones de Activación.....	38
Figura 26. Red neuronal con una capa oculta	38
Figura 27. Procedimiento de identificación	40
Figura 28. Control Distribuido	41

Figura 29. Niveles de una Red Industrial	43
Figura 30. Estructura básica de los elementos de un PLC	47
Figura 31. Partes principales del PLC Siemens S7-1200	49
Figura 32. Partes principales de un conjunto de PLC Siemens S7-1500	50
Figura 33. Módulo didáctico utilizado	60
Figura 34. Plano P&ID	60
Figura 35. Sensor ultrasónico UB800-18GM40	62
Figura 36. Sensor de Temperatura PT100 de tres hilos	62
Figura 37. Sensor de flujo YF-S201	63
Figura 38. Sensor de Presión Danfoss MBS33M	63
Figura 39. Variador de frecuencia SINAMICS V20	64
Figura 40. P&ID para el lazo de control de nivel	65
Figura 41. Señal de nivel en función del tiempo	71
Figura 42. Salida del variador de velocidad en función del tiempo	71
Figura 43. Señal de nivel para la segunda medición	72
Figura 44. Salida del variador de velocidad para la segunda medición	72
Figura 45. Complemento PID Tuner	73
Figura 46. Importación de datos al modelo de Matlab	73
Figura 47. Acceso al complemento PID Tuner	74
Figura 48. Identificación de nueva planta	74
Figura 49. Configuración de los parámetros de importación	75
Figura 50. Resultado de la estimación automática	75
Figura 51. Resultado del espacio de estado de nivel 8	76
Figura 52. Exportación del modelo de planta generado	76
Figura 53. Implementación del modelo matemático en Simulink	77
Figura 54. Ingreso hacia el entorno de trabajo de Simulink	78
Figura 55. Diagrama en lazo cerrado para el controlador PID	78
Figura 56. Propiedades del bloque PID	79
Figura 57. Sintonización del bloque PID mediante PID Tuner	80
Figura 58. Parámetros finales del bloque PID	80

Figura 59. Respuesta del controlador PID al escalón.....	81
Figura 60. Respuesta del controlador PID a valores de consigna aleatorios.....	81
Figura 61. Aplicación Fuzzy Logic Designer FLD	83
Figura 62. Funciones de membresía variable “error”	85
Figura 63. Funciones de membresía variable “variación”	86
Figura 64. Funciones de membresía variable “frecuencia”	87
Figura 65. Reglas del sistema difuso desarrollado	89
Figura 66. Exportación de la estructura difusa a Matlab.....	90
Figura 67. Bloque ‘Fuzzy Logic Controller’	90
Figura 68. Propiedades del bloque ‘Fuzzy Logic Controller’	91
Figura 69. Diagrama del controlador difuso	92
Figura 70. Acumulación por medio de integración.....	93
Figura 71. Acumulación por medio de memoria.	93
Figura 72. Respuesta del controlador difuso al escalón	94
Figura 73. Respuesta del controlador difuso a valores de consigna aleatorios	94
Figura 74. Creación de la estructura ‘data_neural_plant’	96
Figura 75. Creación de los campos en la estructura.....	96
Figura 76. Estructura ‘data_neural_plant’	97
Figura 77. Bloque ‘NARMA-L2 Controller’	97
Figura 78. Propiedades del bloque ‘NARMA-L2 Controller’	98
Figura 79. Importación de los datos de entrenamiento de la red neuronal de planta .	100
Figura 80. Importación de los datos de entrenamiento de la red neuronal de planta .	101
Figura 81. Parámetros de la arquitectura del controlador neuronal.....	102
Figura 82. Resultados de entrenamiento de la red neuronal	103
Figura 83. Tendencia de validación del entrenamiento	104
Figura 84. Diagrama de regresión del entrenamiento.....	104
Figura 85. Resultado del entrenamiento de la red neuronal de planta	105
Figura 86. Resultado de validación de la red neuronal de planta	105
Figura 87. Resultado de pruebas de la red neuronal de planta	106
Figura 88. Diagrama de la arquitectura de la red.....	106

Figura 89. Diagrama del controlador neuronal.....	107
Figura 90. Respuesta del controlador neuronal al escalón	108
Figura 91. Respuesta del controlador neuronal a valores de consigna aleatorios.....	108
Figura 92. Red industrial del proyecto	110
Figura 93. Esquema generalizado de la funcionalidad I-Device.	111
Figura 94. Configuración I-Device.	112
Figura 95. Configuración de las áreas de transferencia.	112
Figura 96. Bloques de programación para el Maestro1500	113
Figura 97. Creación del bloque de subsistema.....	115
Figura 98. Cambio de nombre de las variables del controlador.....	115
Figura 99. Parámetros del bloque de subsistema.....	116
Figura 100. Parámetros del PLC Coder.....	116
Figura 101. Generación del código estructurado.	117
Figura 102. Código estructurado del controlador.....	118
Figura 103. Generación automática de los bloques.....	118
Figura 104. Bloques de programa generados.....	119
Figura 105. Llamada de la función generada.....	119
Figura 106. Parámetros de la instrucción 'stepinfo'	122
Figura 107. Respuesta de la simulación a un escalón de 3.....	123
Figura 108. Respuesta de la simulación a un escalón de 4.....	125
Figura 109. Respuesta de la simulación a un escalón de 5.....	126
Figura 110. Respuesta de la simulación a un escalón de 6.....	128
Figura 111. Respuesta de la simulación a un escalón de 7.....	129
Figura 112. Respuesta de la simulación a un escalón de 8.....	131
Figura 113. Respuesta de la simulación a un escalón de 9.....	132
Figura 114. Respuesta del controlador PID simulado a los escalones	133
Figura 115. Respuesta del controlador difuso simulado a los escalones	134
Figura 116. Respuesta del controlador neuronal simulado a los escalones	135
Figura 117. Respuesta de la planta a un escalón de 3.....	136
Figura 118. Respuesta de la planta a un escalón de 4.....	137

Figura 119. Respuesta de la planta a un escalón de 5.....	139
Figura 120. Respuesta de la planta a un escalón de 6.....	141
Figura 121. Respuesta de la planta a un escalón de 7.....	143
Figura 122. Respuesta de la planta a un escalón de 8.....	144
Figura 123. Respuesta de la planta a un escalón de 9.....	145
Figura 124. Respuesta del controlador PID implementado a los escalones	146
Figura 125. Respuesta del controlador difuso implementado a los escalones	147
Figura 126. Respuesta del controlador neuronal implementado a los escalones	148

ÍNDICE DE TABLAS

Tabla 1	Descripción de las funciones de membresía de la variable "error"	84
Tabla 2	Descripción de las funciones de membresía de la variable "variación"	85
Tabla 3	Descripción de las funciones de membresía de la variable "frecuencia"	87
Tabla 4	Descripción de los elementos de la estructura 'data_neural_plant'	96
Tabla 5	Descripción de los equipos que conforman la red industrial	110
Tabla 6	Respuesta de los controladores simulados para un Setpoint de 3.....	123
Tabla 7	Respuesta de los controladores simulados para un Setpoint de 4.....	124
Tabla 8	Respuesta de los controladores simulados para un Setpoint de 5.....	126
Tabla 9	Respuesta de los controladores simulados para un Setpoint de 6.....	127
Tabla 10	Respuesta de los controladores simulados para un Setpoint de 7.....	129
Tabla 11	Respuesta de los controladores simulados para un Setpoint de 8.....	130
Tabla 12	Respuesta de los controladores simulados para un Setpoint de 9.....	132
Tabla 13	Respuesta de la planta para un Setpoint de 3.	136
Tabla 14	Respuesta de la planta para un Setpoint de 4.	138
Tabla 15	Respuesta de la planta para un Setpoint de 5.	140
Tabla 16	Respuesta de la planta para un Setpoint de 6.	141
Tabla 17	Respuesta de la planta para un Setpoint de 7.	142
Tabla 18	Respuesta de la planta para un Setpoint de 8.	144
Tabla 19	Respuesta de la planta para un Setpoint de 9.	146

RESUMEN

El presente trabajo de titulación se basa en la evaluación de tres tipos diferentes de métodos de control para procesos industriales dentro de sistemas distribuidos, como lo es el control convencional PID, el control aplicando lógica difusa y el control a través de redes neuronales artificiales; los cuales son implementados en un sistema industrial didáctico. Inicialmente, se detallan las características de los controles proporcional, integral y derivativo, así como una introducción a la lógica difusa, su funcionamiento y modelos de control, junto con un breve resumen acerca de las redes neuronales artificiales y su comportamiento. A continuación, se desarrolla cada uno de los modelos de controladores en el entorno Simulink de la herramienta matemática Matlab, en donde se comprueba el correcto funcionamiento de estos a través de un modelo simulado del sistema. Posteriormente, se realiza la exportación de cada uno de los controladores desarrollados hacia el entorno de TIA PORTAL para su implementación en la red de dispositivos SIEMENS disponibles en el sistema industrial, desde el cual se realiza las mediciones de respuesta de cada controlador hacia una serie de escalones para su caracterización utilizando la herramienta Matlab. Finalmente, se establecen las ventajas que presenta cada tipo de control para el proceso industrial seleccionado.

PALABRAS CLAVE:

- **SISTEMAS DE CONTROL**
- **CONTROL PID**
- **CONTROL DIFUSO**
- **CONTROL NEURONAL**
- **MATLAB**

ABSTRACT

This work is based on an evaluation of three different types of control methods for industrial processes within distributed systems, such as conventional PID control, control applying fuzzy logic and control using artificial neural networks, which be implemented in an industrial didactic system. Initially, the characteristics of the proportional, integral and derivative controls are detailed, as well as an introduction to fuzzy logic, its functioning and control models, together with a brief summary about artificial neural networks and their behavior. Each of the controller models is then developed in Simulink environment of Matlab mathematical tool, where the correct functioning of these is checked through a simulated system model. Subsequently, each of the controllers developed is exported to TIA PORTAL for implementation in the network of SIEMENS devices available in the industrial system, from which the response measurements of each controller are made to a series of steps for characterization using the Matlab tool. Finally, the advantages of each type of control for the established industrial process are selected.

KEYWORDS:

- **CONTROL SYSTEMS**
- **PID CONTROL**
- **FUZZY CONTROL**
- **NEURAL CONTROL**
- **MATLAB**

CAPITULO I

1. PROBLEMA

1.1. Antecedentes

El control de procesos industriales es en la actualidad una actividad que abarca varias áreas del conocimiento, dentro de las cuales se debe tener en cuenta aspectos técnicos, científicos y económicos.

Varios investigadores han considerado como primordial el análisis y evaluación de los sistemas de control, colocando a los controladores como los dispositivos fundamentales del caso; aquí tenemos, por ejemplo, los controladores PID, que son suficientes para resolver la necesidad de control en la mayoría de las aplicaciones de la industria, siempre y cuando la dinámica del proceso lo permitan.

De igual manera se presenta la opción de utilizar un controlador que utiliza lógica difusa, la cual nos es ventajosa en situaciones en las cuales no se puede contar con un modelo matemático de la planta a controlar; pero debido a este motivo, es que no se puede realizar simulaciones como tal, pudiendo así utilizar controladores autoajustables.

Como última parte, se puede mencionar a una nueva forma de control que se ha venido utilizando, como son las redes neuronales artificiales, pueden utilizarse en un gran número y variedad de aplicaciones, donde lo principal que se toma en cuenta es el reconocimiento de patrones, la principal ventaja de las redes neuronales es su capacidad para aprender de la experiencia, de generalizar casos anteriores a nuevos casos.

1.2. Planteamiento del problema

La finalidad de cualquier proceso industrial es la realización de un producto final con determinadas características que cumpla con las especificaciones y niveles de fabricación establecidos. La misión del sistema de control de proceso es regular las desviaciones provocadas en las variables de proceso respecto de unos valores preestablecidos, considerados correctos para conseguir los resultados requeridos en el producto producido.

Las técnicas de control de procesos inteligentes están siendo aplicadas en la actualidad en varios sectores de la industria como robots, comunicaciones, desarrollo de equipos, entre muchos más. Las áreas donde se está desarrollando avances sobre todos estos aspectos son: redes neuronales, control difuso, algoritmos genéticos, sistemas de planeación, sistemas expertos e híbridos, los cuales son combinación de más de una de las técnicas anteriores.

Debido a esta variedad de métodos y técnicas de control, cada una de ellas con sus características, procesos, eficiencia y rendimientos propios, un proceso de tipo industrial puede ser controlado con cada uno de estos, lo cual provocará un comportamiento particular según el controlador que se haya elegido.

Con esto en consideración, se mencionan metodologías específicas como algunas de las más importantes tales como: Controladores PID (Proporcional–Integral–Derivativo), Redes Neuronales (Neural Networks) y Control Difuso (Fuzzy Control), las que hoy en día, se han establecido en una poderosa herramienta de diseño alternativo para la

resolución de problemas en una variada clase de aplicaciones en sistemas de control, por lo cual es primordial el determinar cuál de todas ellas resulta el más óptimo y adecuado para el manejo de un proceso en específico.

1.3. Justificación e Importancia

Cuando se planea realizar el control de un sistema a través de métodos convencionales, es totalmente obligatorio obtener un modelo matemático del proceso a ser regulado, pero si el sistema a controlar es algo complejo o con un comportamiento no constante, esta tarea puede tornarse muy complicado.

En estos casos se puede optar por nuevas formas de control, los cuales presentan sus propias ventajas y desventajas según la aplicación en la cual se vaya a implementar. Aquí aparece el control difuso, que es motivado por su característica de capturar estrategias cualitativas de control y su capacidad de implementar un comportamiento de control altamente flexible; y el control neuronal, el cual está caracterizado por su capacidad de “aprender” y reaccionar a casos desconocidos que se puedan presentar en el sistema. El mismo proceso industrial puede ser controlado con cada uno de estos tipos, pero es probable que reaccionen de diferente manera a cada uno de ellos.

Debido a esta gran importancia que tienen los controles industriales, es necesario considerar los diferentes modelos que se tienen disponibles para realizar, junto con sus características más relevantes para así decidir cuál será el más eficiente en aplicaciones industriales didácticas como: procesos multivariable (nivel, flujo, presión y temperatura) y procesos de hidráulica y neumática proporcional.

1.4. Objetivos

1.4.1. Objetivo General

Caracterizar y evaluar los lazos de control PID, lógica difusa y redes neuronales de sistemas distribuidos industriales en el laboratorio de Hidrónica y Neutrónica de la Universidad de las Fuerzas Armadas ESPE.

1.4.2. Objetivos Específicos

- Investigar acerca de las características y parámetros de los controladores PID, así como las características de los controladores inteligentes difusos (Fuzzy) y redes neuronales artificiales.
- Diseñar y comparar el desempeño de cada uno de los controladores PID, lógica difusa y redes neuronales respecto a los otros, utilizando para ello una aplicación industrial de control multivariable.
- Establecer las ventajas y desventajas de cada uno de los tipos de controladores usados dentro de un sistema distribuido industrial.

CAPÍTULO II

2. MARCO TEÓRICO

2.1. Introducción

En el presente capítulo se abarcará definiciones y terminología, la cual nos ayudará a tener un breve conocimiento acerca del área de estudio y, sobre todo, identificar los diferentes elementos que interfieren en el análisis de sistemas de control, así como también los diversos tipos de controladores que se pueden utilizar dentro de un ámbito industrial. Se detallará brevemente los conceptos sobre Control PID, Lógica Difusa y Redes Neuronales, así como también lo correspondientes a Redes Industriales, y sobre los equipos a utilizar en el desarrollo del presente proyecto.

2.2. Sistema de Control

El concepto básico del sistema de control puede describirse mediante el sencillo diagrama de bloques que se muestra en la Figura 1, donde el objetivo del sistema es controlar la variable o “Resultado” de una manera prescrita por la señal de accionamiento u “Objetivo” a través de los elementos del sistema de control. (Kuo, 2009)



Figura 1. Componentes básicos de un sistema de control

Fuente: (Kuo, 2009)

En términos más comunes, la variable controlada es la salida del sistema, y la señal de actuación es la entrada. Como ejemplo simple, en el mando de dirección de un

automóvil, la dirección de las dos ruedas delanteras puede considerarse como la variable controlada (salida). La posición del volante es la entrada, la señal de accionamiento. El proceso o sistema controlado en este caso se compone de los mecanismos de dirección, incluyendo la dinámica de todo el automóvil; sin embargo, si el objetivo es controlar la velocidad del automóvil, entonces la cantidad de presión ejercida sobre el acelerador es la señal de accionamiento, con la velocidad considerada como la variable controlada.

2.2.1. Definiciones básicas

Para comprender lo referente a un sistema de control, es necesario añadir ciertas definiciones.

- **Variable Controlada y Variable Manipulada:** La variable controlada es la magnitud que se mide y controla, y la variable manipulada es la magnitud que es modificada por el controlador para afectar el valor de la variable controlada. Normalmente, la variable controlada es la salida del sistema. Se entiende por control la medición del valor de la variable controlada del sistema y la asignación de la variable manipulada al sistema para corregir o limitar la desviación del valor medido respecto de un valor deseado. (Ogata, 2010)
- **Planta:** Una planta puede ser un aparato o un conjunto de piezas de una máquina que funcionan juntas con el fin de realizar una operación determinada; se puede llamar una planta a cualquier objeto físico a ser controlado. (Ogata, 2010)
- **Proceso:** Cualquier operación a ser controlada. (Ogata, 2010)

- **Sistemas:** Un sistema es una combinación de dos o más elementos que funcionan juntos y realizan un objetivo específico; no se restringe a los sistemas físicos. (Ogata, 2010)
- **Perturbaciones:** Una perturbación es una señal que tiene la tendencia a influir negativamente en el valor de la salida de un sistema. Si se origina una perturbación dentro del sistema, se denomina perturbación interna, mientras que una de tipo externa se genera fuera del sistema y es una entrada. (Ogata, 2010)
- **Control de retroalimentación:** El control de realimentación se refiere a una operación que, en caso de existir perturbaciones, permite reducir la diferencia entre la salida de un sistema y una determinada entrada de referencia, y que lo hace en función de dicha diferencia. (Ogata, 2010)

2.3. Tipos de Sistemas de Control

Se puede realizar una clasificación de los sistemas de control de acuerdo con el procedimiento lógico aplicado por el controlador para manejar la estabilidad del proceso. A continuación, se describen los diferentes tipos de control utilizados en los procesos industriales.

2.3.1. Sistemas de ciclo abierto

Cuando un sistema no tiene la estructura de realimentación, se denomina sistema de bucle abierto, que es el tipo de sistema de control más simple y económico, donde la variable controlada no influye en ningún punto con la variable manipulada. En general, los elementos este tipo de sistema están representados por el diagrama de bloques de

la Figura 2, donde una señal de entrada o comando r es aplicada al controlador, cuya salida actúa como la señal de actuación u . La señal de actuación entonces activa el proceso controlado y es de esperarse que la variable controlada y alcance el valor deseado. Desafortunadamente, los sistemas de control de lazo abierto carecen de precisión y versatilidad, y sólo pueden utilizarse en los tipos de aplicaciones más simples; pero, aunque son de uso limitado, constituyen los elementos básicos de los sistemas de control de ciclo cerrado. (Kuo, 2009)

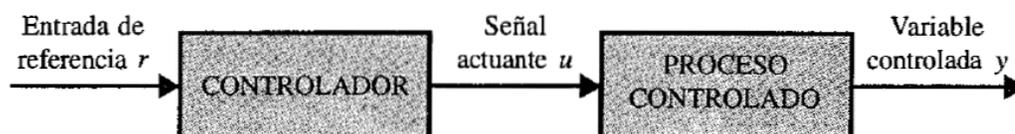


Figura 2. Elementos de un sistema de control de ciclo abierto

Fuente: (Kuo, 2009)

2.3.2. Sistemas de ciclo cerrado

Un sistema con una o más conexiones de realimentación se denomina sistema de circuito cerrado, donde dichas conexiones consisten en que la señal controlada debe ser retroalimentada y contrastada con la entrada de referencia, y una señal de accionamiento proporcional a la diferencia entre la salida y la entrada debe ser enviada a través del sistema para corregir el error. Los elementos básicos y el diagrama de bloques de un sistema de control de ciclo cerrado se muestran en la Figura 3; pero en forma general, la configuración de un sistema de control de realimentación no puede limitarse solo al mostrado, ya que en sistemas complejos puede haber una multitud de lazos de retroalimentación y bloques de elementos. (Kuo, 2009)

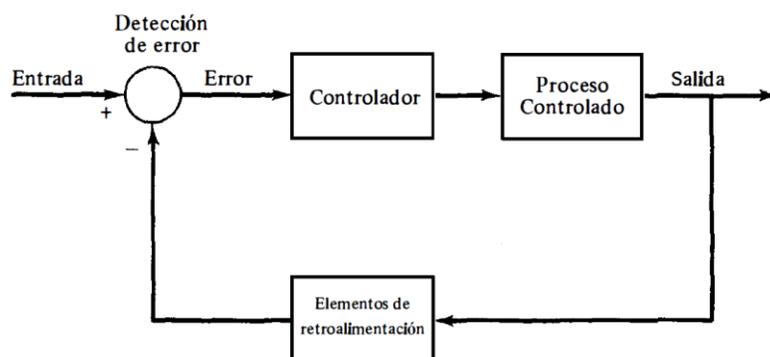


Figura 3. Elementos de un sistema de control de ciclo cerrado

Fuente: (Kuo, 2009)

2.4. Sistemas de Control Avanzado

Son aquellos sistemas que tienen a alejarse de sus pares analógicos como los controles analógicos PID, cuya finalidad es aumentar la eficiencia económica del proceso, que se presenta como alternativa cuando las técnicas de control convencional presentan diferentes problemáticas respecto a retardos o perturbaciones, por ejemplo. Su aplicación permite conseguir beneficios en la automatización de una planta, como son el ahorro de energía, disminución de los costos de operación y aumento de la capacidad de fabricación, entre otros. (Creus, 2011)

Dependiendo del caso, el control avanzado puede significar cosas diferentes, pudiendo tratarse de la aplicación de esquemas de alimentación o de control en cascada, de compensadores de retardo de tiempo, de algoritmos de autoajuste o adaptativos, hasta de estrategias de optimización; además se requiere de una apreciación del problema desde el punto de vista de la ingeniería con una comprensión del comportamiento de la planta de proceso junto con el uso de tecnologías de control no necesariamente de última generación. (Kant, 2011)

2.4.1. Control Experto:

Un controlador experto ideal debe ser capaz de satisfacer los siguientes objetivos:

1. Capacidad para controlar una gran clase de procesos que pueden variar en el tiempo, no lineales, con una variedad de perturbaciones.
2. Requisito de conocimiento previo mínimo sobre el proceso.
3. Capacidad para mejorar su rendimiento con el tiempo, a medida que adquiere más y más conocimientos
4. Capacidad para supervisar el rendimiento del sistema y detectar problemas con sensores, actuadores y otros componentes.

El mejor ejemplo de este tipo de control, por el nivel de acogida que ha tenido a nivel industrial, es el Control de Lógica Difusa, el cual se basa en la aplicación del álgebra difusa propuesta por Lofti Zadeh en 1965, cuando se trata de representar el pensamiento lógico humano, y que se ha convertido en el sistema de control avanzado que más ha sido acogido, ya que en los últimos 50 años se ha generado más de 50.000 patentes de esta metodología sólo en Japón y Estados Unidos. (Fundacion BBVA, 2013)

2.4.2. Control Óptimo

Este control se basa en la elaboración de una función o funcionalidad que suele incluir el error de control y la acción de control y/o sus variaciones, con pesos que permiten el "ponderado" relativo de cada uno de ellos, y se establece un criterio de optimización sobre este último, que se ajusta a los objetivos del control; además de poderse incluir las restricciones correspondientes, lo cual produce una acción de control óptima según los

criterios sujetos a ellas. El mejor representante de este tipo de control, por la amplia aceptación que ha tenido en el ámbito del control de procesos es el Control Predictivo de Modelos o "MPC", en donde el sistema utiliza un modelo matemático del proceso para anticipar el comportamiento futuro del sistema frente a posibles maniobras de control a efectuar. (Vera, 2009)

2.4.3. Control Robusto

Consiste en establecer una arquitectura de control que tenga un rendimiento acorde con las exigencias del sistema, independientemente de las perturbaciones a las que se esté sometido. Los últimos desarrollos en esta área están relacionados con Control Óptimo; sin embargo, otro buen ejemplo que se ha aplicado a nivel industrial es el Control Interno de Modelos (IMC), ya que igual que en MPC, incorpora un modelo matemático de la planta, pero en este caso se utiliza para compensar la dinámica modelable y no modelable de la planta, y su buena aceptación proviene de la capacidad del controlador de equipararse al control PID. (Vera, 2009)

2.4.4. Control Ajustable o Adaptable

Este sistema de control es adecuado para sistemas que son variantes a lo largo del tiempo. Una de las estrategias de control más conocidas dentro de este tipo de control es "Control Adaptivo por Modelo de Referencia" ("Reference Model Adaptive Control" o "RMAC"), en el cual se establece un modelo con una dinámica como la que se desea que tenga el sistema, con el objetivo de tener un sistema de lazo cerrado cuyos parámetros puedan ser regulados para determinar el tipo de respuesta deseada del sistema, en otras

palabras, la que tendría si el sistema fuera el mismo que el modelo de referencia. (Vera, 2009)

2.4.5. Control Neuronal

Los controladores neuronales son estructuras matemáticas que pueden procesar las interacciones entre las entradas y salidas de un proceso, por lo que puede Interpretar el comportamiento de un proceso de bucle abierto o directamente la relación de bucle cerrado. El propósito de usar redes neuronales es tener un sistema que se comporte como una "caja negra" que pueda simular el comportamiento de un sistema, para lo cual se requiere una etapa de formación (en la que la información disponible del sistema se introduce intentando que la red "aprenda" lo que quiere que se le enseñe) y una de "validación" (en la que se contrasta el aprendizaje de la red con otros datos disponibles del sistema que no se utilizan para la formación); cuando finalmente la red ha "aprendido", puede ser como si funcionara con el sistema real. (Vera, 2009)

2.5. Control PID

Un PID (Proporcional Integral Derivativo) es un dispositivo de control por realimentación que se utiliza en sistemas de control industriales, el cual regula el error entre un valor medido y el valor que se quiere obtener calculando y aplicando una acción correctora que se puede ajustar al proceso. El algoritmo de cálculo para este control se da en tres parámetros distintos: el proporcional, integral, derivativo. El valor proporcional determina la reacción del error actual; el integral genera una corrección proporcional a la integral del error, lo que asegura que el error estacionario se reduce a cero al aplicar un

esfuerzo de control suficiente, mientras que el derivativo establece la reacción del tiempo en el que el error se produce. La suma de estas tres acciones es usada para ajustar al proceso vía un elemento de control, y mediante su ajuste en el algoritmo de control del PID, el controlador puede proveer un control diseñado para lo que requiera el proceso, como puede verse en la Figura 4. Se pretenderá lograr que el bucle de control corrija eficazmente y en el mínimo tiempo posible los efectos de las perturbaciones (Salas, 2009).

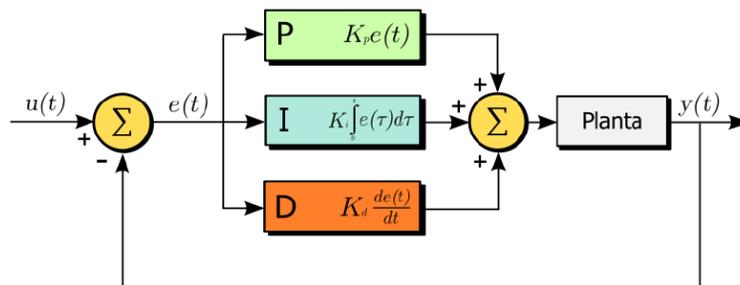


Figura 4. Diagrama de bloques del sistema de control PID
Fuente: (Salas, 2009)

2.5.1. Control Proporcional

La parte proporcional consiste en el producto entre la señal de error y la constante proporcional como para que hagan que el error en estado estacionario sea casi nulo, pero en la mayoría de los casos, estos valores solo serán óptimos en una determinada porción del rango total de control, siendo distintos los valores óptimos para cada porción del rango. La parte proporcional no considera el tiempo, como se ve en la Figura 5, por lo tanto, la mejor manera de solucionar el error permanente y hacer que el sistema contenga alguna componente que tenga en cuenta la variación respecto al tiempo, es incluyendo y configurando las acciones integral y derivativa. (Salas, 2009)

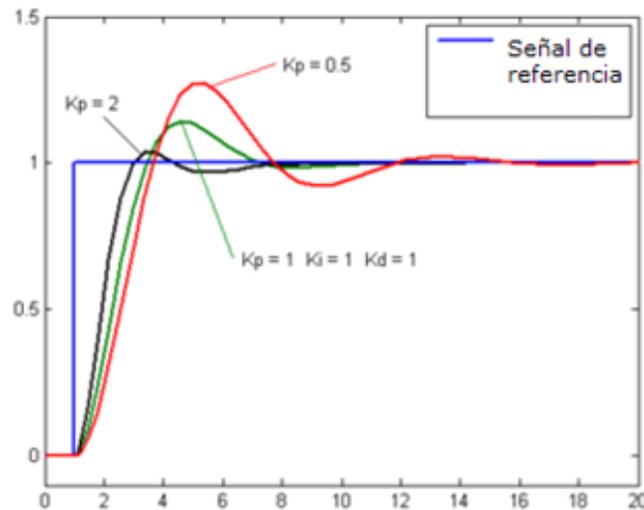


Figura 5. Acción proporcional
Fuente: (Salas, 2009)

La fórmula del proporcional está dada por la Ecuación (1):

$$u(t) = k_p \cdot e(t) \quad (1)$$

Donde:

- $u(t)$, salida de control
- k_p , ganancia proporcional ajustable
- $e(t)$, señal de error

2.5.2. Control Integral

El modo de control integral tiene como propósito disminuir y eliminar el error en estado estacionario, provocado por el modo proporcional; éste actúa cuando hay una desviación entre la variable y el punto de consigna, integrando esta desviación en el tiempo y sumándola a la acción proporcional. El error es integrado, lo cual tiene la función de promediarlo o sumarlo por un periodo de tiempo determinado; luego es multiplicado por una constante I , que representa la constante de integración. Posteriormente, la respuesta

integral es adicionada al modo proporcional para formar el control P+I con el propósito de obtener una respuesta estable del sistema sin error estacionario (véase Figura 6) (Salas, 2009)

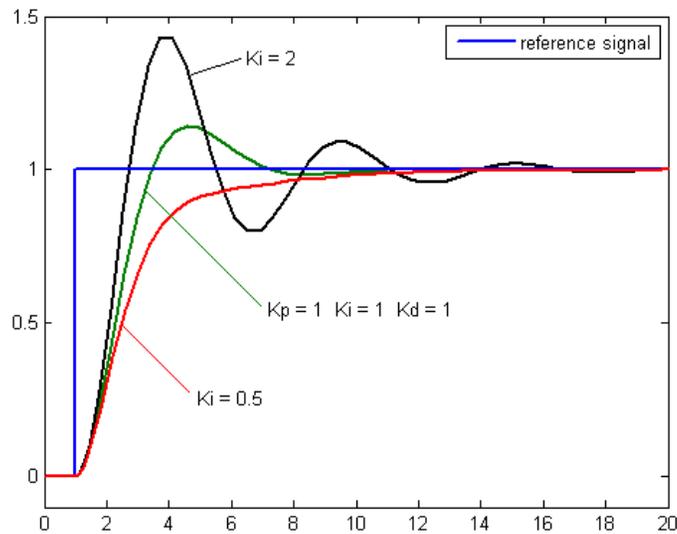


Figura 6. Acción Integral
Fuente: (Salas, 2009)

La fórmula de la integral está dada por la Ecuación (2):

$$u(t) = k_i \cdot \int_0^t e(\tau) d\tau \quad (2)$$

Donde:

- $u(t)$, salida de control
- k_i , ganancia integral ajustable
- t , tiempo instantáneo
- $e(\tau)$, variable de integración (valores desde 0 hasta la t actual).

2.5.3. Control Derivativo

La acción derivativa se manifiesta cuando hay un cambio en el valor absoluto del error, si éste es constante, solamente actúan los modos proporcional e integral. El error es la desviación existente entre el punto de medida y el valor consigna, por lo que la función de la acción derivativa es mantenerlo al mínimo corrigiéndolo proporcionalmente con la misma velocidad que se produce; de esta manera evita que el error se incremente. Se deriva con respecto al tiempo y se multiplica por una constante D y luego se suma a las señales anteriores (P+I). Es importante adaptar la respuesta de control a los cambios en el sistema ya que una mayor derivativa corresponde a un cambio más rápido y el controlador puede responder acordeamente, como se observa en la Figura 7. (Salas, 2009)

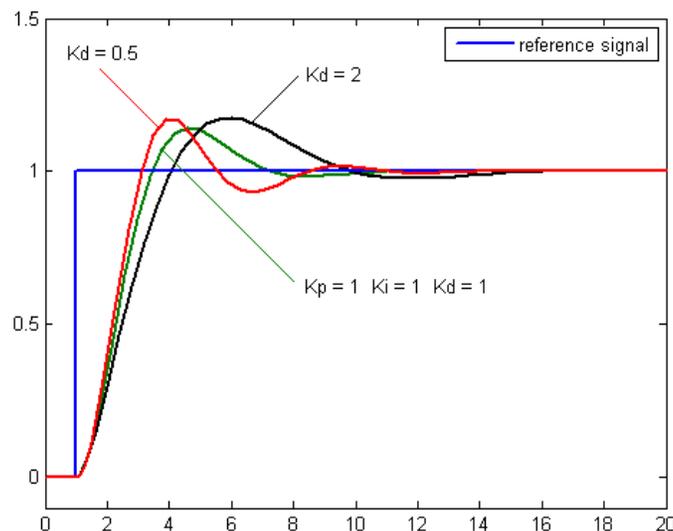


Figura 7. Acción derivativa
Fuente: (Salas, 2009)

La fórmula del derivativo está dada por la Ecuación (3):

$$u(t) = k_d \cdot \frac{de(t)}{dt} \quad (3)$$

Donde: $u(t)$, salida de control
 k_d , ganancia derivativa ajustable
 $e(t)$, señal de error

Esta acción determina por anticipado hacia donde se dirige un proceso y aplica la corrección necesaria para modificar la variación que se está produciendo en el error.

2.5.4. Funcionamiento

Para el correcto funcionamiento de un controlador PID que regule un proceso o sistema se necesita, al menos: un sensor que determine el estado del sistema; un controlador que genere la señal que gobierna al actuador; y un actuador que modifique al sistema de manera controlada. El sensor proporciona una señal analógica o digital al controlador, la cual representa el punto actual en el que se encuentra el proceso o sistema en tensión eléctrica, intensidad de corriente eléctrica o frecuencia. El controlador lee una señal externa que representa el valor que se desea alcanzar, llamada punto de consigna (o punto de referencia), la cual es de la misma naturaleza y tiene el mismo rango de valores que la señal que proporciona el sensor. También resta la señal de punto actual a la señal de punto de consigna, obteniendo así la señal de error, que determina en cada instante la diferencia que hay entre el valor deseado y el valor medido. Esta señal es utilizada por cada uno de los 3 componentes del controlador PID, cuya señal resultante es la salida que el controlador va a utilizar para gobernar al actuador. La señal resultante de la suma

de estas tres se llama variable manipulada y no se aplica directamente sobre el actuador, sino que debe ser transformada para ser compatible con el actuador (Salas, 2009).

2.6. Lógica Difusa

La lógica difusa es una manera de interconectar procesos intrínsecamente analógicos que se mueven a través de un rango continuo de valores con una computadora digital, la cual solo ve las cosas como valores numéricos discretos bien definidos. (Goebel , 2003)

La lógica difusa tiene dos significados diferentes. En un sentido estricto, es un sistema lógico, que es una extensión de la lógica multivaluada; sin embargo, en un sentido más amplio, es casi sinónimo de la teoría de los conjuntos difusos, una teoría que se relaciona con clases de objetos con límites poco definidos en los que la membresía es una cuestión de grado. En esta perspectiva, la lógica difusa en su sentido estricto es una rama de la segunda definición. (MathWorks, Inc., 2018)

El objetivo de la lógica difusa es mapear un espacio de entrada a un espacio de salida, y el mecanismo principal para hacer esto es una lista de sentencias “si - entonces” llamadas reglas. Todas las reglas se evalúan en paralelo, y el orden de las reglas no es importante. Las reglas mismas son útiles porque se refieren a variables y a los adjetivos que describen esas variables. Antes de que pueda construir un sistema que interprete reglas, debe definir todos los términos que planea usar y los adjetivos que los describen, por ejemplo, para decir que el agua está caliente, es necesario definir el rango en el que se puede esperar que varíe la temperatura del agua, así como lo que queremos decir con la palabra caliente. (MathWorks, Inc., 2018)

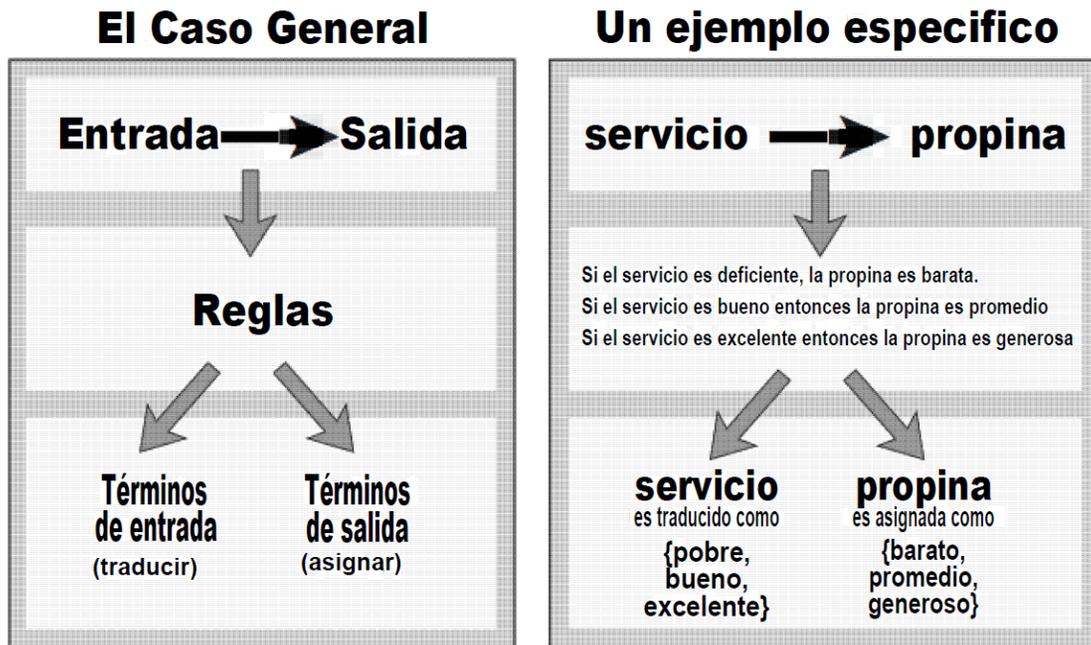


Figura 8. Ejemplificaron de una inferencia difusa
 Fuente: (MathWorks, Inc., 2018)

La Figura 8 proporciona una hoja de ruta para el proceso de inferencia difusa, en donde se muestra la descripción general de un sistema difuso a la izquierda y un sistema difuso específico a la derecha. Para resumir el concepto de representado en esta figura, la inferencia difusa es un método que interpreta los valores en la parte de entrada y, basado en un conjunto de reglas, asigna valores a cada una de salida. En general, la lógica difusa modela la toma de decisiones como lo haría una persona, basadas en información con las características descritas en los párrafos anteriores; además utiliza expresiones que no son totalmente verdaderas ni correctas, sino que pueden tomar un valor cualquiera de entre dos extremos, en los cuales se encuentra la verdad absoluta y la falsedad total. (Pineda & Vivas, 2008)

2.6.1. Conjuntos difusos

La lógica difusa comienza con el concepto de un conjunto difuso, el cual es un conjunto sin un límite nítido y claramente definido, o sea, que puede contener elementos con un grado parcial de membresía. Para entender lo que es un conjunto difuso, primero se debe considerar la definición de un conjunto clásico, que es un contenedor que incluye o excluye totalmente cualquier elemento dado. Por ejemplo, en la Figura 9 se nos indica el conjunto de días de la semana, el cual incuestionablemente incluye lunes, jueves y sábado; de la misma manera, es incuestionable que excluye la mantequilla, la libertad, las aletas dorsales, etc. En la lógica clásico, *“de cualquier tema, una cosa debe ser afirmada o negada”*; según esta ley con anotaciones, tenemos que "de cualquier tema (lunes), una cosa (día de la semana) debe ser afirmada o negada (se afirma que lunes es un día de la semana)". (MathWorks, Inc., 2018)



Figura 9. Ejemplo de un conjunto clásico
Fuente: (MathWorks, Inc., 2018)

Ahora, se considera el conjunto de días que comprende un fin de semana, la Figura 10 se intenta clasificarlos y se tiene que la mayoría estaría de acuerdo en que el sábado y el domingo pertenecen, pero ¿qué pasa con el viernes? Se considera como una parte del fin de semana, pero de alguna manera parece que debería ser técnicamente excluido;

por lo tanto, en el diagrama anterior, viernes hace todo lo posible para "ingresar al grupo". La lógica clásica no toleraría este tipo de clasificación, ya que, o algo está dentro o está fuera, aun cuando la experiencia humana sugiere algo diferente. En la lógica difusa, *"la verdad de cualquier afirmación se convierte en una cuestión de grado"*. Cualquier enunciado puede ser difuso, y he aquí la mayor ventaja que ofrece el razonamiento difuso, que es la capacidad de responder a una pregunta de sí - no con una respuesta que no es del todo sí-o-no. Los humanos hacen este tipo de cosas todo el tiempo, pero es un truco bastante nuevo para las computadoras. (MathWorks, Inc., 2018)



Figura 10. Ejemplo de un conjunto difuso
Fuente: (MathWorks, Inc., 2018)

Las primeras diferencias que se hacen evidentes entre los conjuntos clásicos y los conjuntos difusos pueden visualizarse gráficamente en la Figura 11, donde se puede especificar:

- La función de membresía asociada a los conjuntos concretos sólo puede tener dos valores: 1 o 0, mientras que en los conjuntos difusos puede tener cualquier valor entre 0 y 1.
- Un elemento puede pertenecer (parcialmente) a un conjunto difuso y simultáneamente pertenecer (parcialmente) al complemento de dicho conjunto. Lo

anterior no es posible en los conjuntos clásicos, ya que constituiría una violación al principio del tercer excluido.

- Las fronteras de un conjunto clásico son exactas, en tanto que las de un conjunto difuso son, precisamente, difusas, ya que existen elementos en las fronteras mismas, y estos elementos están a la vez dentro y fuera del conjunto. (Ruiz, 2007)

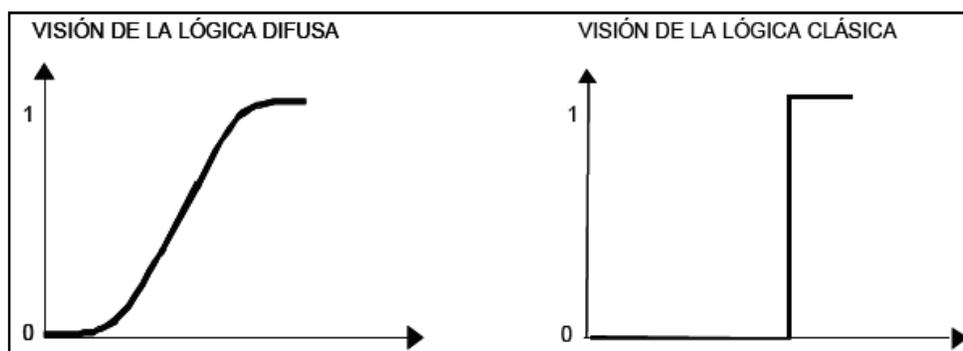


Figura 11. Lógica Clásica versus Lógica Difusa.
Fuente: (Pineda & Vivas, 2008)

2.6.2. Operaciones lógicas

Lo más importante sobre el razonamiento lógico difuso es el hecho de que es un superconjunto de la lógica booleana estándar. En otras palabras, si mantiene los valores difusos en sus extremos de 1 (completamente verdadero) y 0 (completamente falso), las operaciones lógicas estándar se mantendrán, como se puede observar en la Figura 12. Ahora, debido a que en la lógica difusa la verdad de cualquier afirmación es una cuestión de grado, los valores de entrada pueden ser números reales entre 0 y 1, por lo tanto, es necesario establecer nuevas formas de operar cada una de las operaciones lógicas. Por ejemplo, para AND, una respuesta es la operación mínima; es decir, resolver la sentencia $A \text{ AND } B$, donde A y B se limitan al rango $(0,1)$, mediante la selección del menor valor entre ambos. Utilizando el mismo razonamiento, se puede sustituir la operación OR por

la del valor máximo, de forma que $A \text{ OR } B$ se convierte en equivalente a la selección del mayor valor entre ambos. Finalmente, la operación 'NOT A' pasa a ser equivalente a la operación $1-A$. Además, debido a que hay una función detrás de la tabla de verdad y no sólo la tabla de verdad en sí misma, ahora se puede considerar valores distintos de 1 y 0 (MathWorks, Inc., 2018).

A	B	A and B
0	0	0
0	1	0
1	0	0
1	1	1

AND

A	B	A or B
0	0	0
0	1	1
1	0	1
1	1	1

OR

A	not A
0	1
1	0

NOT

Figura 12. Tablas de verdad estándar.

Fuente: (MathWorks, Inc., 2018)

En la Figura 13, la tabla de verdad se convierte en un gráfico de dos conjuntos difusos aplicados juntos para crear un conjunto difuso. La parte superior de la figura muestra gráficos correspondientes a las dos tablas de verdad anteriores, mientras que la parte inferior de la figura muestra cómo funcionan las operaciones en un rango continuamente variable de valores de verdad A y B de acuerdo con las operaciones difusas que haya definido. (MathWorks, Inc., 2018)

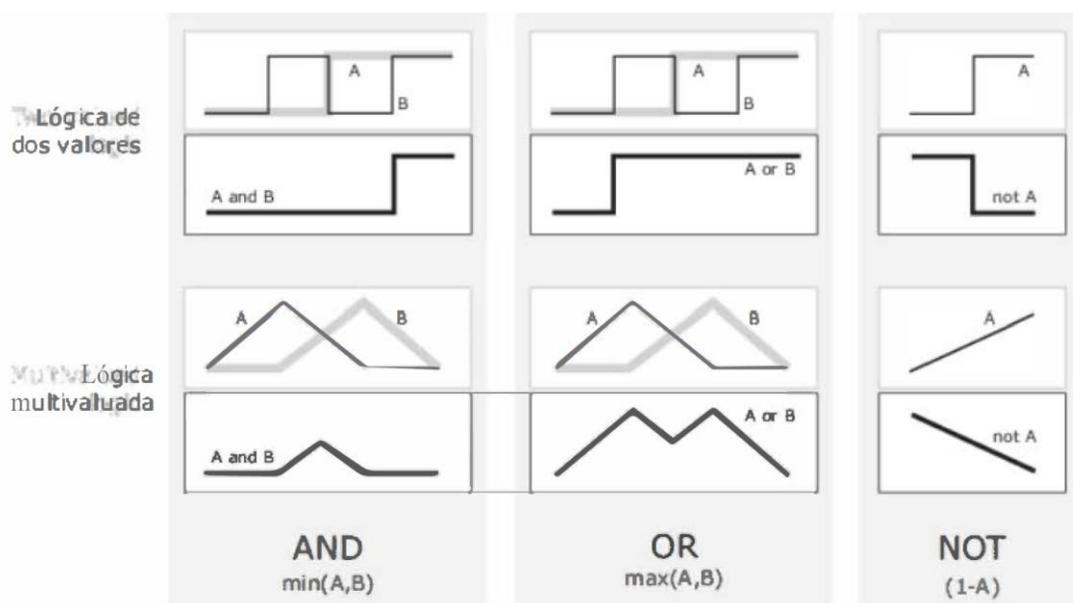


Figura 13. Operaciones lógicas en conjuntos difusos
Fuente: (MathWorks, Inc., 2018)

2.6.3. Variable lingüística

Las variables lingüísticas son fundamentales para las manipulaciones de lógica difusa, contienen valores que se distribuyen uniformemente entre 0 y 1, dependiendo de la relevancia de un término lingüístico dependiente del contexto; por ejemplo, podemos decir que la habitación está caliente y el horno está caliente, y la variable lingüística “caliente” tiene diferentes significados dependiendo de si nos referimos a la habitación o al interior del horno. El valor asignado de 0 significa que el término lingüístico no es verdadero y el valor asignado de 1 indica que el término es verdadero; estas “variables” utilizadas en el lenguaje cotidiano transmiten información relativa sobre nuestro entorno o un objeto bajo observación y pueden transmitir una cantidad sorprendente de información. (Banks & Hayward, 2002)

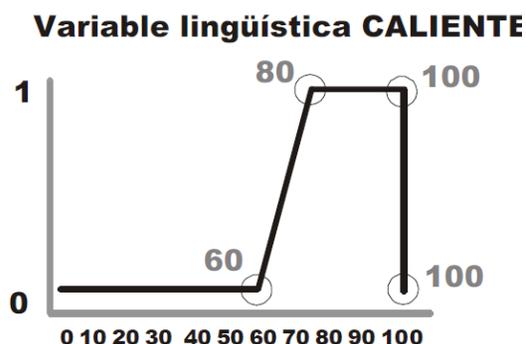


Figura 14. Relación entre temperatura y la variable lingüística “caliente”
Fuente: (Banks & Hayward, 2002)

La variable lingüística CALIENTE en la Figura 14 tiene un valor entre 0 y 1 en el rango 60-80 (donde 0 es “nada caliente” y 1 es “totalmente caliente”). Para cada número nítido en un espacio variable (por ejemplo, una habitación), se pueden aplicar varios términos lingüísticos.

2.6.4. Funciones de membresía

En los conjuntos difusos se contempla la pertenencia parcial de un elemento a un conjunto, lo cual está definido mediante una función propia que está asociada al conjunto difuso, conocida como Función de Membresía (Pineda & Vivas, 2008)

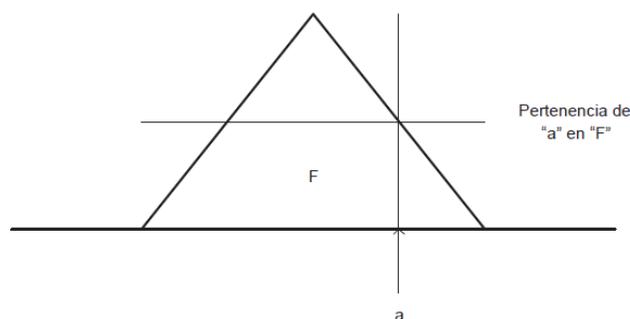


Figura 15. Función de membresía difusa.
Fuente: (Ruiz, 2007)

. Es una curva que define cómo cada punto en el espacio de entrada es mapeado a un valor de membresía (o grado de membresía) entre 0 y 1, como se puede ver en la Figura 15. La única condición que una función de membresía debe realmente satisfacer es que debe variar entre 0 y 1. La función misma puede ser una curva arbitraria cuya forma podemos definir como una función que nos conviene desde el punto de vista de la simplicidad, la conveniencia, la velocidad y la eficiencia. (MathWorks, Inc., 2018). Las funciones características más comúnmente utilizadas por su simplicidad matemática y su manejabilidad son: triangular, trapezoidal, gaussiana, sigmoideal, gama, pi, campana, etc., como se visualiza en la Figura 16; siendo la triangular la forma más utilizada para una función de membresía.

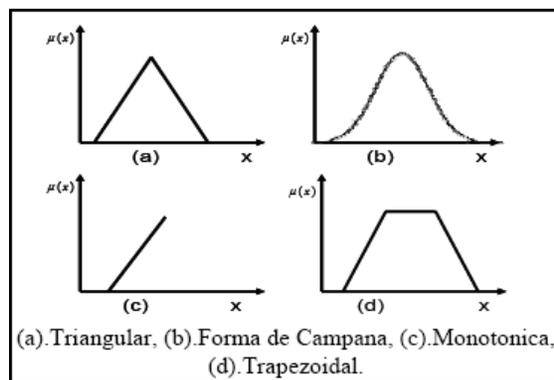


Figura 16. Diferentes tipos de funciones de membresía.
Fuente: (Pineda & Vivas, 2008)

2.6.5. Inferencia difusa

El proceso de aplicar una relación en particular para la interpretación y evaluación de una implicación difusa se denomina proceso de inferencia o simplemente inferencia (Ruiz, 2007).

Estas sentencias de reglas “si-entonces” se utilizan para formular las sentencias condicionales que comprenden la lógica difusa, las cuales asumen la forma: *Si ‘x’ es A, entonces ‘y’ es B*; donde A y B son valores lingüísticos definidos por conjuntos difusos en los rangos (universos de discurso) X e Y, respectivamente. La parte ‘si’ de la regla “x es A” se denomina antecedente o premisa, mientras que la parte ‘entonces’ de la regla “y es B” se denomina consecuencia o conclusión. Un ejemplo de tal regla podría ser: *“Si el servicio es bueno entonces la propina es promedio”*. El concepto ‘bueno’ se representa como un número entre 0 y 1, por lo que el antecedente es una interpretación que devuelve un único número entre 0 y 1. Por el contrario, ‘promedio’ se representa como un conjunto difuso, por lo que el consecuente es una asignación que asigna todo el conjunto difuso B a la variable de salida y. En la regla “si-entonces”, la palabra se utiliza de dos maneras completamente diferentes dependiendo de si aparece en el antecedente o en el consecuente. En general, la entrada a una regla “si-entonces” es el valor actual para la variable de entrada (en este caso, servicio) y la salida es un conjunto difuso completo (en este caso, promedio). Este conjunto se desfuzificará posteriormente, asignando un valor a la salida. La interpretación de una regla “si-entonces” implica dos pasos: evaluación del antecedente, o fuzificar las entradas y aplicar los operadores difusos necesarios; y la aplicación del resultado al consecuente. . (MathWorks, Inc., 2018)

2.6.6. Defuzificación

Después de calcular las reglas de inferencia y evaluar las variables lingüísticas, y tener como resultado salidas difusas, se necesita trasladar estos valores nuevamente hacia el mundo real. En los sistemas difusos, esta información poco clara debe convertirse en un

valor numérico representativo porque, por ejemplo, una variable manipulada no puede procesar una función de membresía. Este cálculo de valores numéricos se conoce como defuzificación, donde requeriremos de una función de membresía para cada una de las variables de salida (Ruiz, 2007).

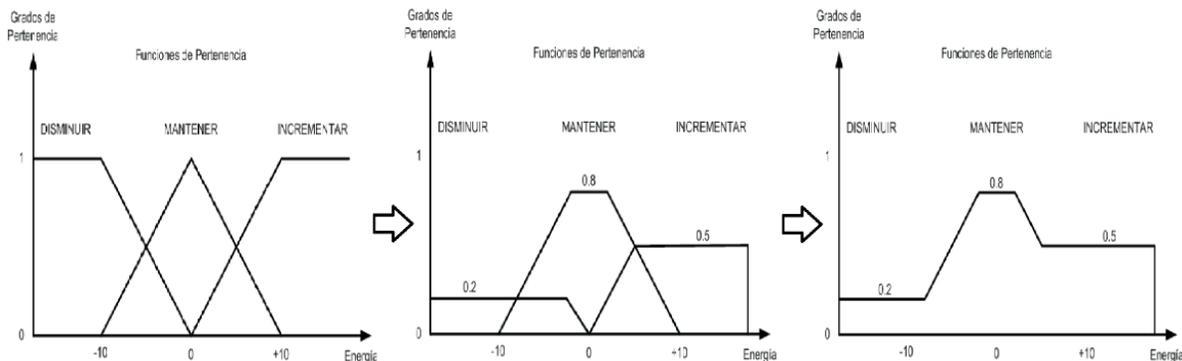


Figura 17. Proceso de defuzificación.
Fuente: (Ruiz, 2007)

Cada función de membresía es truncada al valor de la respectiva variable difusa. A partir de aquí se construye una nueva función de membresía de salida, tomando para cada punto en el eje horizontal, el máximo valor entre las funciones de membresía. El resultado de esta operación es mostrado en la Figura 17. (Ruiz, 2007)

Para completar el proceso de defuzificación, todo lo que resta por hacer es encontrar un punto de equilibrio. Un método para encontrar el valor de salida es mediante el método del centro de gravedad (COG) se muestra en la Ecuación (4), donde $f(x)$ representa la función de membresía final de la variable de salida y a, b son los límites de la función.:

$$COG = \frac{\int_a^b f(x)x dx}{\int_a^b f(x) dx} \quad (4)$$

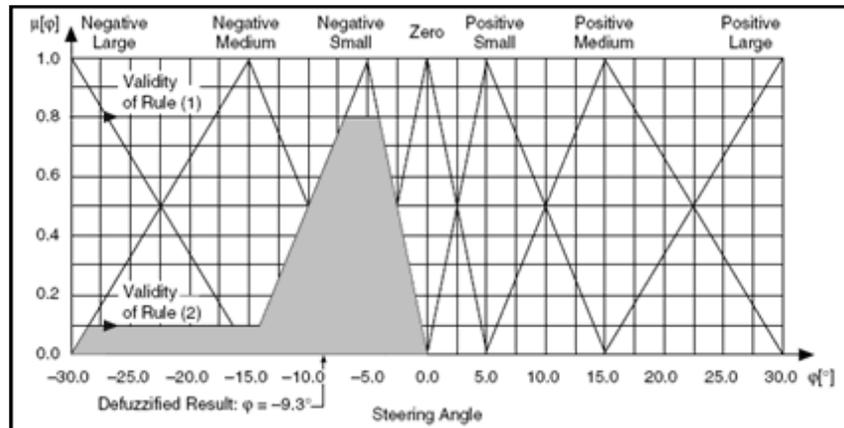


Figura 18. Defuzificación acorde al método COG.

Fuente: (Pineda & Vivas, 2008)

Es el método más utilizado en aplicaciones de la lógica difusa a la ingeniería, ya que se obtiene una solución única, su representación se muestra en la Figura 18. Debido a que la función de membresía es el conjunto de las funciones originales, se puede ampliar la Ecuación (4) de forma que se suma cada una de las áreas de las distintas funciones, como se muestra en la Ecuación (5) (Ruiz, 2007)

$$COG = \frac{\int x f_A(x) dx + \int x f_b(x) dx + \dots}{\int f_A(x) dx + \int f_b(x) dx + \dots} \quad (5)$$

2.6.7. Controladores difusos

Los controladores difusos son muy simples conceptualmente, ya que consisten en una etapa de entrada, una etapa de procesamiento y una etapa de salida. La etapa de entrada mapea el sensor u otras entradas, tales como interruptores, ruedas selectoras, etc., a las funciones de membresía apropiadas y a los valores de verdad; la etapa de procesamiento invoca cada regla apropiada y genera un resultado para cada una, luego combina los resultados de las reglas; y finalmente, la etapa de salida convierte el resultado combinado en un valor de salida de control específico; lo cual puede verse en la Figura 19. La forma

más común de las funciones de membresía es triangular, aunque también se utilizan trapecios y curvas de campana, pero la forma es generalmente menos importante que el número de curvas y su colocación, de tres a siete curvas son generalmente apropiadas para cubrir el rango requerido de un valor de entrada. (Goebel , 2003)

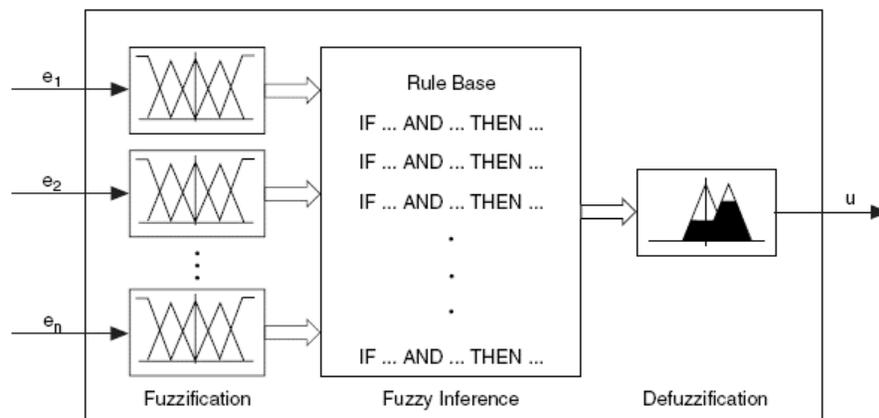


Figura 19. Estructura Interna de un Controlador Difuso
Fuente: (Pineda & Vivas, 2008)

El controlador difuso genérico se presenta en el diagrama a bloques de la Figura 20. Su forma genérica es común, debido a que generalmente tienen la misma forma de diseño, independientemente del proceso que se desee controlar. La diferencia es la sintonización de las reglas del controlador difuso y esto se hará de acuerdo con la experiencia del operador humano. La entrada del controlador es el error y la variación del error, ya que con ellos se puede determinar el comportamiento del sistema, los cuales son fuzificados, para ser enviados a un controlador de lógica difusa (CLD), que mediante las reglas difusas calcula los valores de salida, que serán defuzificados y convertidos en la señal de salida del controlador. (Guzmán & Castaño, 2006).

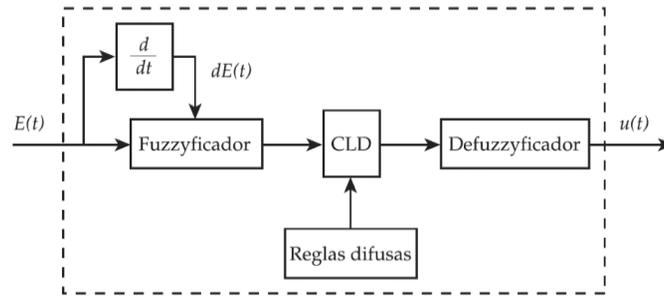


Figura 20. Diagrama a bloques de un controlador difuso genérico
Fuente: (Guzmán & Castaño, 2006)

a. Fuzificación e inferencia

En el primer paso, los grados de verdad de las variables lingüísticas se calculan a partir de los valores actuales de las señales de entrada. En la Figura 21 se establece una fuzificación para las variables de entrada de temperatura y presión, donde para la primera magnitud se obtiene los grados de verdad: frío = 0, cálido = 0,95 y caliente = 0,35; mientras que para presión: bajo = 0,25, medio = 0,85, y alto = 0. (Siemens AG, 2006)

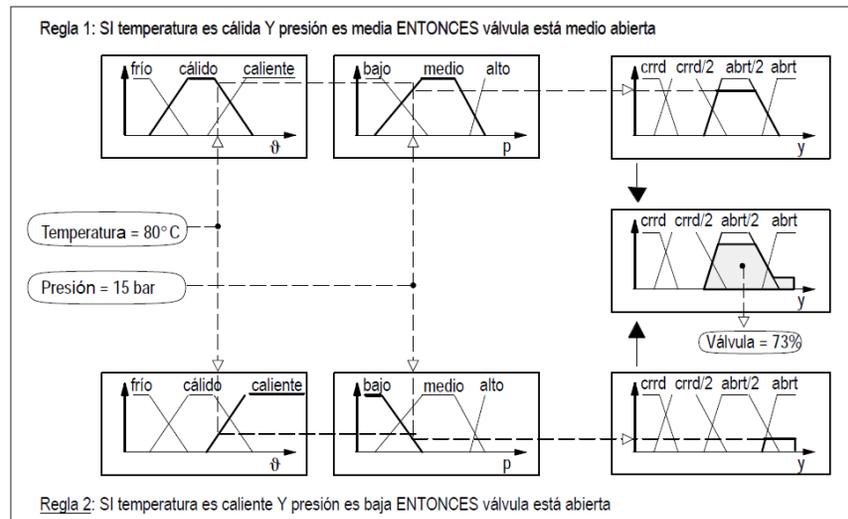


Figura 21. Procesamiento difuso de las variables en un controlador difuso
Fuente: (Pineda & Vivas, 2008)

El siguiente paso es la inferencia, donde las reglas difusas se aplican. En la parte SI de una regla, los grados de verdad que se calcularon en la fuzificación están unidos por el operador difuso, y en la mayoría de los controladores, el mínimo se utiliza para calcular el SI(AND) y el máximo para el O(OR). En el ejemplo, para la regla 01: $t_{\text{cálido}} = 0,95$ y $p_{\text{medio}} = 0,85$ enlazado por $\wedge = 0,85$; mientras que para la regla 02: $t_{\text{caliente}} = 0,35$ y $p_{\text{alto}} = 0,25$ enlazado por $\vee = 0,25$ (Siemens AG, 2006)

La vinculación de las funciones de membresía para una serie de reglas se conoce como inferencia. Una operación lógica OR o la formación de un máximo se utiliza aquí como regla aritmética. Una operación de composición se realiza con la ayuda de este método común de inferencia de máxima-mínima. Esta operación permite que las partes efectivas de todas las reglas entren en juego. La superposición de las funciones de cierre de la membresía de una serie de reglas se suma para formar una sola función de membresía resultante. (Siemens AG, 2006)

b. Defuzificación

Para el cálculo de los valores de salida, las funciones de membresía de la salida se ponderan primero con los grados actuales de verdad. Esta ponderación se puede hacer de acuerdo con diferentes métodos, pero el más común es el método del centro de gravedad (véase 2.6.6). Se calcula como la abscisa del centro de gravedad de la zona situada por debajo de la función de salida; se forma un área de unificación a partir de las porciones de área ponderadas. El centro de gravedad de esta área resultante es el resultado defuzificado que se emite al proceso como el valor de una variable manipulada.

En el ejemplo de la Figura 21, el centro de gravedad está representado por la válvula abierta en un 73%.

2.7. Redes Neuronales

Una red neuronal artificial es un sistema computacional compuesto por elementos que están interconectados en paralelo y jerárquicamente, los cuales procesan información por medio de su estado dinámico como respuesta a las entradas externas. Las redes neuronales artificiales, asemejan propiedades biológicas del funcionamiento del cerebro humano, como la capacidad de memorizar y asociar hechos, basado precisamente en los sistemas neuronales, a través de modelos matemáticos implementados en diversas industrias. Una sólo neurona es diminuta, pero cuando se conectan miles de neuronas pueden llegar a resolver complejos problemas para lo que hayan sido entrenadas. (Morales & Moreno, 2017)

Las redes neurales artificiales son aproximadores no lineales a la forma en que funciona el cerebro; por lo tanto no deben compararse directamente con el cerebro ni confundir los principios que fundamentan el funcionamiento de las redes neurales artificiales y el cerebro, ni pensar que las redes neurales se basan únicamente en las redes biológicas ya que sólo emulan en una parte muy simple el funcionamiento del cerebro humano. (Ponce, 2010)

2.7.1. Neurona Biológica

Una neurona biológica es una célula especializada en procesar información, compuesta por el cuerpo de la célula (soma) y dos tipos de ramificaciones: el axón y las

dendritas, como se muestra en la Figura 22 . La neurona recibe las señales o impulsos de otras neuronas a través de sus dendritas y transmite señales generadas por el cuerpo de la célula a través del axón.

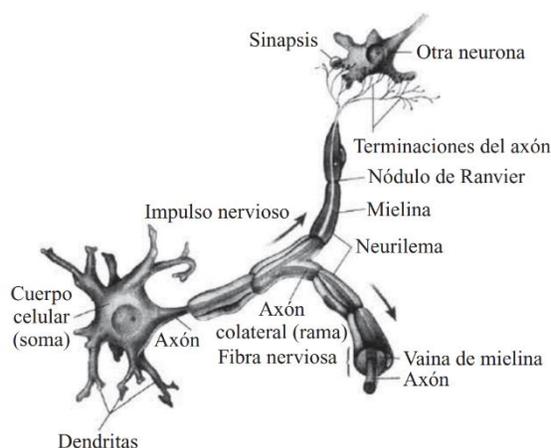


Figura 22. Elementos neurales
Fuente: (Ponce, 2010)

Para establecer una similitud directa entre la actividad sináptica y la analogía con las redes neurales artificiales se considera que las señales que llegan a la sinapsis son las entradas a la neurona; éstas son ponderadas (atenuadas o simplificadas) a través de un parámetro denominado peso, asociado a la sinapsis correspondiente. Estas señales de entrada pueden excitar a la neurona (sinapsis con peso positivo) o inhibirla (peso negativo). El efecto es la suma de las entradas ponderadas: si la suma es igual o mayor que el umbral de la neurona, entonces la neurona se activa (da salida); esta es una situación de todo o nada, cada neurona se activa o no se activa. (Ponce, 2010)

Las conexiones (sinapsis) de una neurona se consideran como se muestra en la Figura 23. En ellas, las activaciones x_i con unas determinadas intensidades w_{ji} de otras neuronas son sumadas, y se permite que en la salida de la neurona (axón) se origine una

actividad siempre que la suma $w_{ji}x_i$ supere un valor umbral (θ_j). (Ponce, 2010) La expresión matemática de esta neurona es:

$$y_i = \varphi \left(\sum_{i=1}^n w_{ji}x_i + \theta_j \right) \quad (6)$$

Donde, w_{ji} son los pesos sinápticos que ponderan las entradas, x_i y θ_j es el umbral. φ es la función de activación de la neurona y n es número total de pesos sinápticos conectados a entrada de la neurona. (Ponce, 2010)

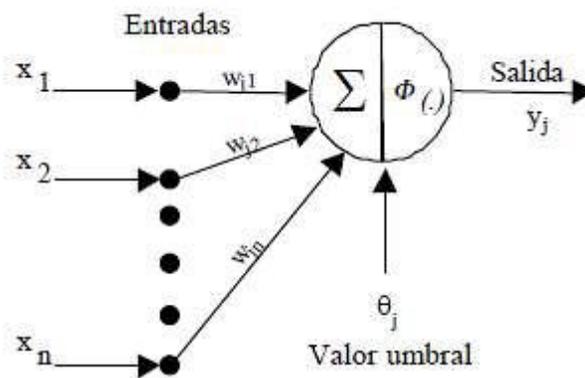


Figura 23. Modelo de una neurona artificial
Fuente: (Ponce, 2010)

2.7.2. Características de las redes neuronales artificiales

a. Pesos

En una red neuronal artificial los pesos representan la fuerza que existe en una sinapsis. Estos pesos pueden ser variables o fijos, en caso de ser fijos la tarea para la cual será utilizada la red neuronal debe estar definida previamente, en el caso de pesos variables estos van adaptándose a medida que la red va aprendiendo la tarea. (Moreno, 2005)

b. Etapa de Aprendizaje

Es la etapa en la que la Red Neuronal Artificial, modifica sus pesos en respuesta a una información de entrada y una salida deseada. La etapa de aprendizaje terminará cuando los pesos no cambien. Existen dos clases de aprendizaje, uno en el que la red aprende de forma supervisada y otro en forma no supervisada. El aprendizaje supervisado se da mediante un maestro que determina la respuesta que debe tener la red para una información de entrada establecida, el maestro verificará la salida y en caso de tener una respuesta errónea se cambiarán los pesos de conexión. (Moreno, 2005)

El aprendizaje no supervisado se da sin necesidad de un maestro, no requiere de ninguna influencia externa para ajustar los pesos de las conexiones. La red no recibe información del entorno que le indique si la salida es correcta. La salida en este tipo de redes puede ser una codificación de los datos de entrada que mantiene los datos relevantes de la información o pueden realizar un mapeo de características de forma que si se presentan datos similares se verán afectadas neuronas cercanas entre sí. (Moreno, 2005)

c. Capas

Una capa es un grupo de neuronas cuyas entradas proviene de la misma fuente y su salida se dirige al mismo destino. Las redes neuronales se pueden componer de tres tipos de capas: de entrada, que son las neuronas que reciben las señales desde el medio, capa oculta son las que poseen sus entradas y salidas dentro de la red y finalmente capa de salida cuyas neuronas envían la señal fuera de la red. Una red neuronal puede tener

una o más capas ocultas. En la Figura 24 se puede observar una red neuronal con una capa de neuronas de entrada, una capa oculta y una capa de salida. (Moreno, 2005)

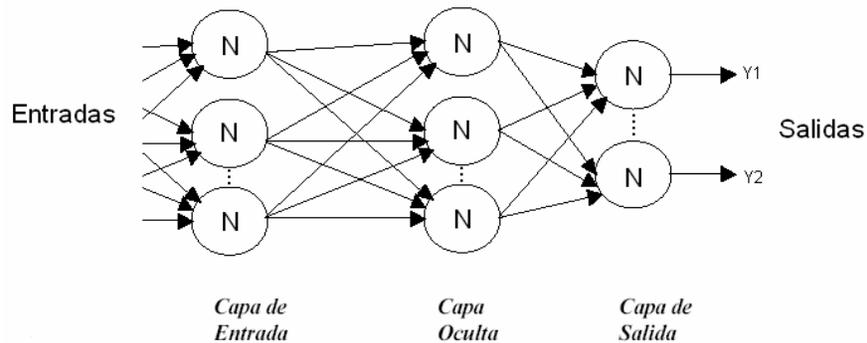


Figura 24. Red Neuronal con varias capas
Fuente: (Moreno, 2005)

d. Nivel de Activación

El nivel de activación o umbral es el valor a partir del cual se activa la salida de la neurona, si la suma de las entradas de la neurona multiplicadas por la fuerza de cada sinapsis (peso) es mayor a este valor umbral la neurona generará la función de activación. (Moreno, 2005)

e. Funciones de Activación

Es la función de salida de la neurona, valor en la salida de la neurona, algunas veces es llamada función de transferencia, esta función usualmente alimenta las entradas de otras neuronas por medio de sinapsis a menos que sea de una neurona de la capa de salida de la red. Generalmente se usan funciones de salida no lineales en redes multicapa con el fin de hacer redes que pueden resolver problemas no lineales. Las funciones de salida pueden tener distintas formas como se indica en la Figura 25 (Moreno, 2005).

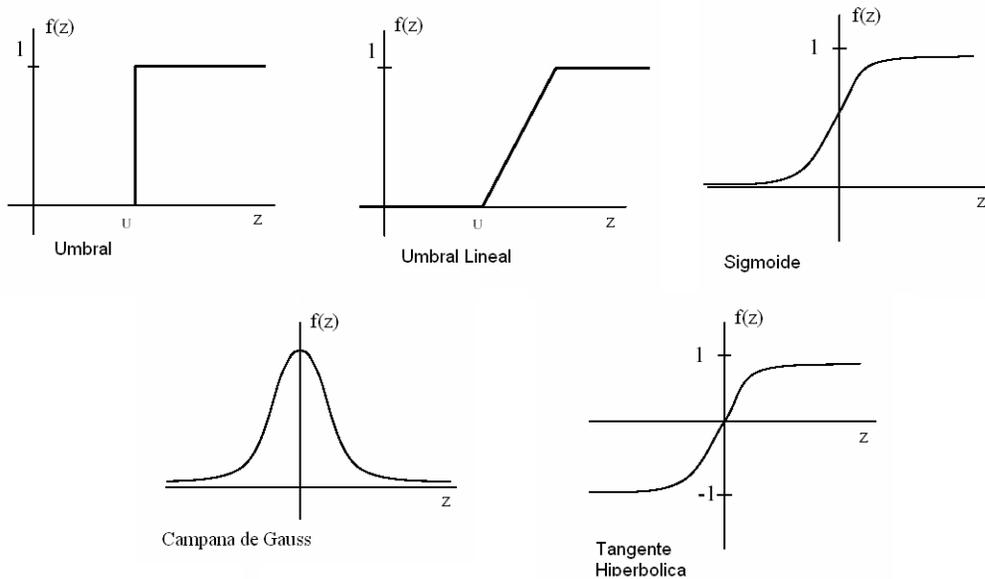


Figura 25. Funciones de Activación

Fuente: (Moreno, 2005)

2.7.3. Funcionamiento de la red neuronal

Una red neuronal tiene la capacidad de reconocer relaciones complejas y no lineales entre distintas variables, a partir de las neuronas disponiendo de una completa red en paralelo. Las más sencillas constan de una capa de entrada, una capa oculta y una capa de salida, como se puede observar en la Figura 26 (Morales & Moreno, 2017).

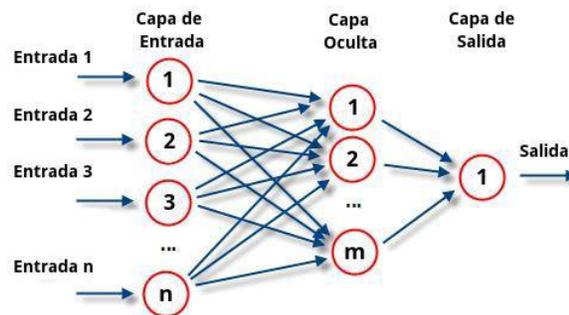


Figura 26. Red neuronal con una capa oculta

Fuente: (Morales & Moreno, 2017)

El funcionamiento de una neurona consiste en la transformación de los valores de entrada a través de las conexiones en una salida. Se caracterizan por los siguientes elementos:

1. La función de propagación, que consiste en la sumatoria de todas las entradas multiplicadas por los pesos de las conexiones más un valor de sesgo.
2. La función de activación, que, si existe, cumple la función de activar o desactivar la salida de la neurona.
3. La función de transferencia, que se aplica al resultado de la función de propagación (Morales & Moreno, 2017).

2.7.4. Ventajas de utilizar RNA

Debido a su condición de aprender a partir de experiencias, el uso de RNA en cualquier campo representa ventajas como:

- *Aprendizaje adaptativo*: Es la capacidad de aprender a realizar una actividad basada en una experiencia inicial.
- *Auto organización*: Puede crear su propia organización de la información que recibe mediante una etapa de aprendizaje.
- *Tolerancia a fallos*: Si se destruye parcialmente una red, degrada su estructura, sin embargo, puede retener algunas de sus capacidades así sufra un gran daño.
- El tiempo de operación es en tiempo real (Morales & Moreno, 2017).

2.7.5. Entrenamiento de la red neuronal

El objetivo del entrenamiento de la red es que tenga la capacidad de minimizar la función de error, lo que en otras palabras significa que la salida de la red se aproxime a la salida en los datos. Para el entrenamiento de la red neuronal artificial en Matlab se pueden utilizar diferentes herramientas como: por medio de códigos desde la línea de comandos o desde el workspace de MATLAB; utilizando el conjunto de bloques incluidos en el toolbox de Control Systems de Neural Networks Blocks en Simulink; y a través de la Interfaz Gráfica de Usuario (GUI) de Redes Neuronales Artificiales . (Morales & Moreno, 2017)

Como tal, el procedimiento del entrenamiento de la red neuronal consiste en que aprenda un comportamiento específico para poder corregir fallas en los procesos. En cuanto a la identificación del sistema los pasos a seguir son los siguientes:

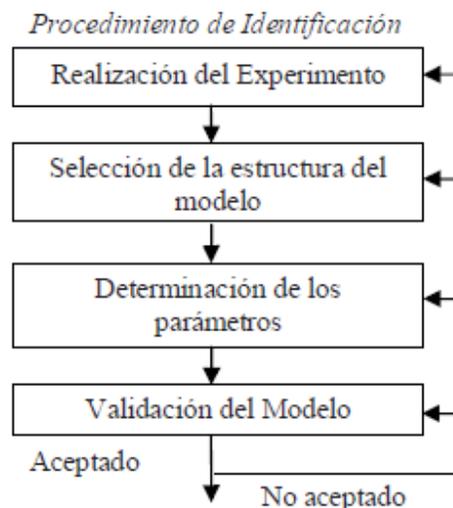


Figura 27. Procedimiento de identificación

Fuente: (Morales & Moreno, 2017)

2.8. Comunicaciones Industriales

Desde la primera máquina automatizada a base de componentes electromecánicos hasta las grandes instalaciones compuestas por multitud de máquinas trabajando coordinadamente, existe un denominador común que en mayor o menor medida ha estado siempre presente: la relación de la máquina con su entorno. Esto se visualiza en la aparición de los autómatas programables o PLC, (Programmable Logic Controllers) los cuales permitieron reducir en gran medida la cantidad de material necesario para conseguir controlar una máquina; para ello los controles se programan en su interior y las modificaciones de funcionamiento no significan necesariamente cambios físicos, además de que el tiempo necesario para el mantenimiento se reduce (Rodríguez, 2008).

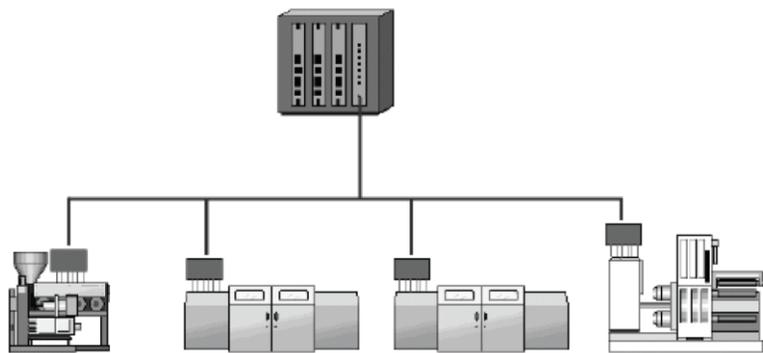


Figura 28. Control Distribuido

Fuente: (Rodríguez, 2008)

La posibilidad de conectar los autómatas entre sí permitió eliminar casi todo el cableado de control entre máquinas, quedando solamente una línea de comunicación entre ellas, a través de la cual se podía coordinar el funcionamiento de todos los componentes de un sistema, lo cual hace que el sistema sea más fiable y menos costoso, pues los elementos de control no necesitan ser tan complejos. A estas líneas se las llama

Buses de Campo, que permiten unir todos los elementos de control necesarios de forma que puedan intercambiar mensajes entre ellos. A este conjunto se le conoce como Control Distribuido (véase Figura 28), que es un sistema complejo se divide en subsistemas autónomos con control propio, que se integran gracias a un sistema de comunicaciones común (Rodríguez, 2008).

2.8.1. Niveles en una Red Industrial

En una red industrial coexistirán equipos y dispositivos de todo tipo, los cuales suelen agruparse jerárquicamente para establecer conexiones lo más adecuadas a cada área. De esta forma se definen cuatro niveles dentro de una red industrial, los cuales pueden visualizarse en la Figura (Universidad de Oviedo, 2006).

- **Nivel de gestión:** es el nivel más elevado y se encarga de integrar los niveles inferiores en una estructura de fábrica, e incluso de múltiples factorías. Las máquinas aquí conectadas suelen ser estaciones de trabajo que hacen de puente entre el proceso productivo y el área de gestión, en el cual se supervisan las ventas, stocks, etc. Se emplea una red de tipo LAN (Local Area Network) o WAN (Wide Area Network).
- **Nivel de control:** se encarga de enlazar y dirigir las distintas zonas de trabajo. A este nivel se sitúan los autómatas de gama alta y los ordenadores dedicados a diseño, control de calidad, programación, etc. Se suele emplear una red de tipo LAN.

- **Nivel de campo y proceso:** se encarga de la integración de pequeños automatismos (autómatas compactos, multiplexores de E/S, controladores PID, etc.) dentro de subredes. En el nivel más alto de estas redes se suelen encontrar uno o varios autómatas modulares, actuando como maestros de la red o maestros flotantes. En este nivel se emplean los buses de campo.
- **Nivel de E/S:** es el nivel más próximo al proceso, donde están los sensores y actuadores, encargados de manejar el proceso productivo y tomar las medidas necesarias para la correcta automatización y supervisión. (Universidad de Oviedo, 2006)

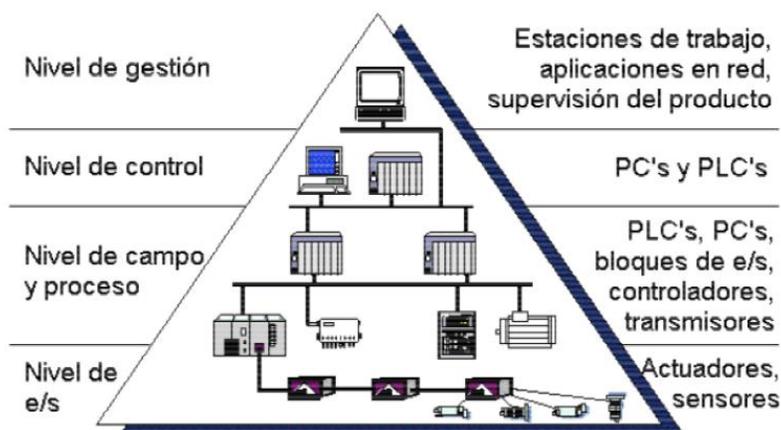


Figura 29. Niveles de una Red Industrial

Fuente: (Universidad de Oviedo, 2006)

2.8.2. Buses de Campo

Un bus de campo es un sistema de transmisión de información (datos) que simplifica enormemente la instalación y operación de máquinas y equipamientos industriales utilizados en procesos de producción; su objetivo es sustituir las conexiones punto a punto entre los elementos de campo y el equipo de control a través del tradicional bucle

de corriente de 4-20mA. Típicamente son redes digitales, bidireccionales, multipunto, montadas sobre un bus serie, que conectan dispositivos de campo como PLCs, transductores, actuadores y sensores, donde cada uno incorpora cierta capacidad de proceso, que lo convierte en un dispositivo inteligente, manteniendo siempre un costo bajo. El objetivo es reemplazar los sistemas de control centralizados por redes de control distribuido mediante el cual permita mejorar la calidad del producto, reducir los costos y mejorar la eficiencia. (Kaschel & Pinto)

2.8.3. PROFIBUS

Profibus se desarrolló bajo un proyecto financiado por el gobierno alemán. Está normalizado en Alemania por DIN E 19245 y en Europa por EN 50170. El desarrollo y posterior comercialización ha contado con el apoyo de importantes fabricantes como ABB, AEG, Siemens, Klöckner-Moeller. Está controlado por la PNO (Profibus User Organization) y la PTO (Profibus Trade Organization) (Kaschel & Pinto). Existen tres perfiles:

- **Profibus DP (Decentralized Periphery):** Orientado a sensores y/o actuadores enlazados a procesadores (PLCs) o terminales.
- **Profibus PA (Process Automation):** Para control de proceso y cumpliendo normas especiales de seguridad para la industria química (IEC 1 1 15 8-2, seguridad intrínseca).
- **Profibus FMS (Fieldbus Message Specification):** Para comunicación entre células de proceso o equipos de automatización. La evolución de Profibus hacia

la utilización de protocolos TCP/IP para enlace al nivel de proceso hace que este perfil esté perdiendo importancia (Kaschel & Pinto).

2.8.4. Industrial Ethernet

La norma IEEE 802.3 basada en la red Ethernet de Xerox se ha convertido en el método más extendido para interconexión de computadores personales en redes de proceso de datos. En la actualidad ha generado una revolución con su desplazamiento hacia las redes industriales, ya que diversos buses de campo establecidos como Profibus, Modbus etc. La han adoptado como la red apropiada para los niveles superiores. Ethernet está ocupando un área importante entre las opciones para redes industriales, además de que pueda llegar a penetrar en los niveles bajos de la pirámide de automatización. (Kaschel & Pinto)

2.8.5. PROFINET

PROFINET, es la continuación consecuente de PROFIBUS DP, el ya acreditado bus de campo, e Industrial Ethernet, el bus de comunicación para el nivel de célula. La experiencia de ambos sistemas ha sido y está siendo integrada en PROFINET. PROFIBUS International desarrolló la plataforma PROFINET que permite el uso de comunicaciones tanto en tiempo real como por TCP/IP en la misma red. Este nuevo concepto permite disponer de una tecnología común en la planta, Ethernet TCP/IP, brindando la posibilidad de aplicar soluciones y desarrollos orientados hasta ahora sólo al mundo informático, tales como: tecnologías con acceso inalámbrico; diagnóstico y acceso Web pueden implementarse como solución para la transmisión remota y segura; acceso a variadores de velocidad o dispositivos inteligentes como HMIs u otros PLCs;

diagnóstico de estaciones en aplicaciones Web y otras aplicaciones comunes, garantizando la transmisión confiable y los estándares de seguridad originales de PROFIBUS DP (Castro, y otros, 2012).

Durante los últimos años, el estándar PROFINET alcanzó un avanzado grado de desarrollo que le permite ser utilizado para satisfacer los requerimientos de automatización de procesos, tales como: integración de redes de instrumentación; integración de variadores de velocidad y maniobra de motores; redundancia y funciones de seguridad (Guano & Romero, 2014)

2.9. Controladores Lógicos Programables (PLC)

Un PLC o Controlador Lógico Programable es un dispositivo digital electrónico con una memoria programable para el almacenamiento de instrucciones, permitiendo la implementación de funciones específicas de tipo lógicas, secuenciales, temporizadas, de conteo y aritméticas; con el objeto de controlar máquinas y procesos. También se lo puede definir como un equipo electrónico, el cual realiza la ejecución de un programa de forma cíclica; aunque en momentos puede realizar otras tareas prioritarias, se garantiza ejecución completa del programa principal. Estos equipos son utilizados en ambientes industriales donde la decisión y la acción deben ser tomadas en forma muy rápida, para responder en tiempo real; además donde se requieran tanto controles lógicos como secuenciales o ambos a la vez (Automación Micromecánica, 2018)

La estructura básica de un PLC se puede observar en la Figura 30, que está compuesta por:

- La CPU (procesador y memoria)
- Las interfases de entradas.
- Las interfases de salidas.

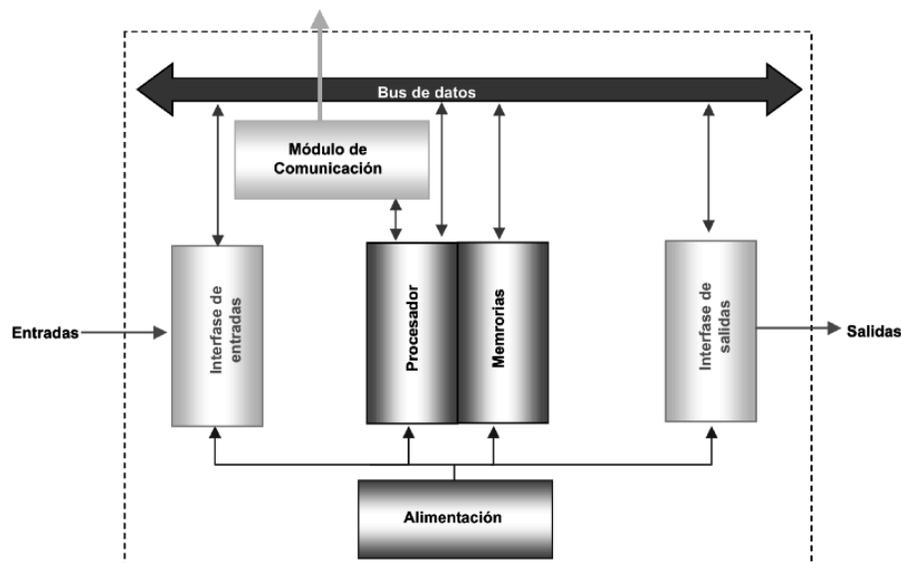


Figura 30. Estructura básica de los elementos de un PLC
Fuente: (Automación Micromecánica, 2018)

2.9.1. Procesador

Es el “cerebro” del PLC, y tiene como principales tareas la ejecución del programa realizado por el usuario, además de la administración de la comunicación entre el dispositivo de programación y la memoria, y entre el microprocesador y los bornes de entrada/ salida. Para poder realizar todas estas tareas, el procesador necesita un programa escrito por el fabricante, llamado sistema operativo, el cual no es accesible por el usuario. (Automación Micromecánica, 2018)

2.9.2. Memoria

Los PLC tienen que ser capaces de almacenar y retirar información, para lo cual utilizan memorias, que son miles de cientos de localizaciones donde la información puede ser almacenada, las cuales están muy bien organizadas. Éstas deben ser capaces de almacenar datos del proceso como señales de entradas y salidas, variables internas, datos alfanuméricos y constantes; así como los datos de control que son instrucciones de usuario/programa y la configuración propia del autómatas (Automación Micromecánica, 2018).

2.9.3. Entradas y salidas

Los dispositivos de entrada y salida son aquellos equipos que intercambian (o envían) señales con el PLC, donde cada dispositivo de entrada es utilizado para conocer una condición particular de su entorno, como temperatura, presión, posición, entre otras; mientras que los dispositivos de salida son aquellos que responden a las señales que reciben del PLC, cambiando o modificando su entorno (Automación Micromecánica, 2018).

Las entradas se pueden clasificar en:

- **Discretas:** También denominadas binarias, lógicas, digitales u on-off. Pueden tomar únicamente dos estados.
- **Analógicas:** Permiten tomar una cantidad de valores intermedios dentro de un rango. Por ejemplo: de 4 a 20 mA, 0 a 5Vcc o 0 a 10 Vcc.

- **Especiales:** Son variantes de las analógicas como ser las entradas de pulsos de alta velocidad, termocuplas, RTD, etc. (Amores & Quingatuña, 2017)

2.9.4. PLC S7-1200

El controlador S7-1200 proporciona la flexibilidad y la potencia para controlar una amplia variedad de dispositivos en apoyo de las necesidades de automatización. Posee un diseño compacto, que en conjunto con la configuración flexible y su amplio número de instrucciones lo convierten en una solución perfecta para controlar una amplia variedad de aplicaciones. En la figura se visualiza sus principales partes, donde la CPU combina un microprocesador, una fuente de alimentación integrada, circuitos de entrada y salida, PROFINET integrado, E/S de control de movimiento de alta velocidad y entradas analógicas integradas en una carcasa compacta (Siemens AG, 2016)

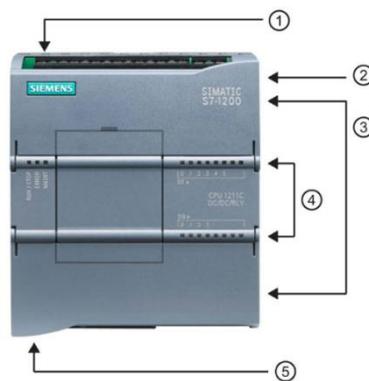


Figura 31. Partes principales del PLC Siemens S7-1200
Fuente: (Automación Micromecánica, 2018)

Sus partes más principales, indicadas en la Figura 31, son las siguientes:

- ① Conector de alimentación
- ② Ranura para tarjeta de memoria (detrás de la tapa superior)
- ③ Conectores de cableado de usuario extraíbles (detrás de las tapas)

- ④ LEDs de estado para las E/S integradas
- ⑤ Conector PROFINET (en la parte inferior de la CPU)

2.9.5. PLC S7-1500

El controlador SIMATIC S7-1500 es el sistema de control modular para una amplia variedad de aplicaciones de automatización en la automatización discreta. El diseño modular y sin ventilador, la sencilla implementación de estructuras distribuidas y su fácil manejo hacen de SIMATIC S7-1500 la solución económica y cómoda para una gran variedad de tareas. La CPU contiene el sistema operativo y ejecuta el programa de usuario, que se encuentra en la tarjeta de memoria SIMATIC y se procesa en la memoria de trabajo de la CPU. La conexión al proceso se centraliza o distribuye a través de PROFINET con módulos de E/S. Esta interfaz en la CPU permite la comunicación simultánea con dispositivos PROFINET, controladores PROFINET, dispositivos HMI, dispositivos de programación, otros controladores y otros sistemas. La CPU 1511-1 PN soporta la operación como un controlador IO y un dispositivo I. (Siemens AG, 2016)

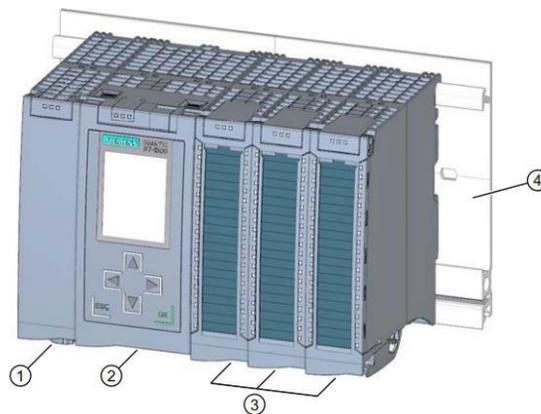


Figura 32. Partes principales de un conjunto de PLC Siemens S7-1500
Fuente: (Siemens AG, 2016)

Sus partes más principales, indicadas en la Figura 32, son las siguientes:

- ① Módulo de alimentación del sistema
- ② CPU
- ③ Módulos de periferia
- ④ Perfil soporte con perfil DIN integrado

2.10. Matlab

MATLAB es el nombre abreviado de “MATrix LABoratory”. La plataforma de MATLAB está optimizada para resolver problemas de ingeniería y científicos. El lenguaje MATLAB basado en matrices es la forma más natural del mundo de expresar las matemáticas computacionales. Los gráficos incorporados facilitan la visualización y la comprensión de los datos. Una vasta biblioteca de toolbox’s preconstruidos te permiten comenzar de inmediato con los algoritmos esenciales para el dominio. El entorno de escritorio invita a la experimentación, exploración y descubrimiento. Estas herramientas y capacidades de MATLAB están rigurosamente probadas y diseñadas para trabajar conjuntamente. (MathWorks, Inc., 2018)

2.10.1. Características principales

- Lenguaje de alto nivel para la informática científica y de ingeniería
- Entorno de escritorio optimizado para la exploración, el diseño y la resolución de problemas de forma iterativa
- Gráficos para visualizar datos y herramientas para crear gráficos personalizados

- Aplicaciones para ajuste de curvas, clasificación de datos, análisis de señales, sintonización de sistemas de control y muchas otras tareas.
- Toolbox adicionales para una amplia gama de aplicaciones científicas y de ingeniería
- Herramientas para crear aplicaciones con interfaces de usuario personalizadas
- Interfaces para C/C++, Java®, .NET, Python, SQL, Hadoop y Microsoft® Excel®. (MathWorks, Inc., 2018).

2.10.2. Simulink

Simulink® es un entorno de diagramas de bloques para simulación multidominio y diseño basado en modelos. Soporta el diseño a nivel de sistema, la simulación, la generación automática de código y las pruebas y verificaciones continuas de sistemas embebidos. Simulink proporciona un editor gráfico, librerías de bloques personalizables y solucionadores para modelar y simular sistemas dinámicos. Está integrado con MATLAB®, lo que le permite incorporar algoritmos de MATLAB en modelos y exportar resultados de simulación a MATLAB para su análisis posterior. Un bloque puede representar un componente físico, un sistema pequeño o una función; una relación de entrada/salida caracteriza completamente al bloque. La definición de un bloque sólo se completa con sus entradas y salidas y esta tarea se relaciona con el objetivo del modelo. (MathWorks, Inc., 2018)

2.11. Hipótesis

Mediante la evaluación del comportamiento de los controladores, se determinará las ventajas y desventajas de los diferentes tipos de control para la optimización y mejora de procesos productivos en sistemas distribuidos industriales.

2.12. Variables de la investigación

- VARIABLE INDEPENDIENTE: La evaluación del comportamiento de los controladores PID, lógica difusa y redes neuronales.
- VARIABLE DEPENDIENTE: Optimización y mejora de procesos productivos en sistemas distribuidos industriales

CAPÍTULO III

3. METODOLOGÍA

El presente proyecto se basa en desarrollo e implementación de controladores de diferente tipo para el manejo de procesos industriales, en la recolección de información de su comportamiento para un posterior análisis mediante el software Matlab que permitirá el procesamiento de datos mediante simulaciones y comparaciones entre ellos, hasta finalmente la obtención de resultados que nos permitirá conocer las características de cada uno de los tipos de controladores desarrollados.

3.1. Antecedentes investigativos

El control de procesos industriales es una de las herramientas más importantes para lograr optimizar los parámetros que reportan productividad y/o calidad en las líneas de producción, sin embargo, debido a que se fundamenta en bases físico matemáticas complejas, tiende a ser visto como algo que usarán científicos y no para el personal que hace mantenimiento o producción en nuestras industrias cotidianamente. Sin estos desarrollos físico matemáticos en todos los campos, la tecnología actual no sería lo avanzada que es, pero también hay que comprender el rol de cada uno en el momento que se vive, determinar si uno se es desarrollador, o se tiene el papel de usuario en este campo, entendiendo que ambos roles coexisten y van de las ideas a lo tangible a diario. (Intrave, 2017)

Para (Alomía & Bastidas, 2013), mientras que el control convencional está profundamente arraigado y ha podido resolver la mayoría de los problemas de control

con un mínimo costo, existiendo poca incentivo para cambiar el control convencional en aplicaciones donde este no responde de modo satisfactorio, por un control que se suponía más complejo, ellos esperan que con el desarrollo de su investigación se pueda cambiar esta percepción, demostrando las bondades del control difuso, no pretendiendo en ningún caso sustituir los controladores PID con controladores difusos, al contrario complementarlos y contribuir a encontrar su verdaderos campos de aplicación. También afirman que no existe limitaciones a las capacidades que tiene un PLC para realizar control discreto en un tiempo lo suficientemente corto (tiempo real), con el fin de poder implementar controladores avanzados para procesos complejos en la industria, aprovechándose los equipos instalados, sin necesidad de requerir de un controlador especial.

En lo que respecta a la lógica difusa, para (Guzmán & Castaño, 2006) en su trabajo concluyen que no cabe duda que el control difuso es una herramienta muy sencilla de aplicar y diseñar para todo tipo de procesos, sin embargo se encontraron importantes complicaciones, pero el auge de la lógica difusa sigue en ascenso debido a su gran simplicidad y a los avances logrados para cubrir estas deficiencias como lo son:

- La dependencia en la experiencia del experto en el proceso, para un correcto funcionamiento del controlador.
- Una vez obtenida la simulación del sistema no es fácil determinar qué cambios se necesitan hacer en las funciones de membresía, reglas difusas, métodos de inferencia y defuzificación para poder obtener un resultado de salida deseado., lo que se complica aún más por la no linealidad del controlador difuso.

- La dificultad de garantizar la estabilidad del sistema para cualquier referencia dada, dado que el controlador es no lineal.

Por otro lado, el uso de herramientas de inteligencia artificial, como las redes neuronales artificiales (RNA), es cada vez más habitual en distintos ámbitos industriales como pueden ser el control de procesos y de la calidad y la predicción de fallos operacionales. Existen herramientas basadas en RNA capaces de facilitar aspectos de la ingeniería de procesos con un esfuerzo de aprendizaje relativamente pequeño, hasta el punto de que algunas de esas herramientas pueden ser usadas como funciones o plugin de hojas de cálculo comerciales, sumando su potencia computacional a las funciones de cálculo y análisis que estas últimas ofrecen. La utilización eficaz de un modelo de RNA requiere de una sistemática, para obtener el máximo partido de la información disponible. NNpred y Alyuda Forecaster XL son dos herramientas de software de RNA que sirven de apoyo a la ingeniería de procesos industriales. (Espina, García, & Larrañaga, 2011)

Para (Ardila, 2010), entre los desafíos que actualmente enfrenta el campo del control de procesos, se encuentran:

- Control de sistemas al mismo tiempo de dinámicas tanto simbólicas como continuas. La próxima generación de sistemas combinarán operaciones lógicas (tales como el razonamiento simbólico y la toma de decisiones) con cantidades continuas (tales como tensiones y posiciones). La teoría actual no se ha adaptado para hacer frente a esos sistemas, especialmente a medida que la escala de los sistemas es muy elevada.

- El control distribuido en entornos de una red asíncrona. El control distribuido a través de múltiples unidades de computación, interconectados a través de las comunicaciones basadas en paquetes, requerirá de nuevos formalismos para garantizar la estabilidad, rendimiento y robustez. Esto es especialmente cierto en aplicaciones donde no se puede ignorar las limitaciones computacionales y de comunicaciones en las operaciones de control.

3.2. Modalidad de la Investigación.

3.2.1. Método Bibliográfico Documental

Mediante este método se pretende recolectar información necesaria sobre los métodos de control de procesos convencionales, como lo es el control PID, además de nuevas formas de control como lo son el realizado aplicando Lógica Difusa y el que se utiliza Redes Neuronales Artificiales; así como las formas en las cuales se puede aplicar dichos controladores en forma física a través de un PLC, desde el cual se podrá controlar y monitorear el comportamiento de los mismos, además se tomará en cuenta los resultados obtenidos para su futura comparación entre ellos.

3.2.2. Método Inductivo – Deductivo

Mediante este método se va a diseñar los controladores mediante el software Matlab para realizar las simulaciones y obtener los datos de su compartimiento; así mismo la programación para el controlador físico utilizando el software TIA PORTAL, donde se pretende monitorear las principales variables y el estado actual de proceso; junto con una

manera para obtener los datos reales del proceso y cotejarlos con los obtenidos de la simulación.

3.2.3. Método Experimental

Mediante esta metodología se manipulará las variables de los procesos a controlar en lazo abierto y cerrado, tomando en consideración la comunicación vía PROFINET para la obtención de los datos a analizar, de esta manera se pretende realizar una comparativa entre cada uno de los controladores determinando los beneficios y desventajas que conllevan cada uno de ellos al momento de manipular el proceso para obtener una salida deseada.

3.3. Tipo de investigación

3.3.1. Investigación Cuantitativa

El método cuantitativo manifiestan que usan la recolección de datos para verificar la hipótesis, con base en a tablas estadísticas, curvas y patrones de comportamiento de las variables para demostrar teorías, además menciona que es un proceso secuencial, donde no se puede eludir ninguna etapa de la investigación. En este tipo de investigación se utilizaron modelos matemáticos y recolección de datos y parámetros físicos para la implementación de un modelo computacional para su posterior análisis mediante el software Matlab que permite la simulación e interpretación de la información para determinar el comportamiento de los controladores.

CAPÍTULO IV

4. ANÁLISIS RESULTADOS DE LA INVESTIGACIÓN

Este capítulo presenta los resultados obtenidos luego de la investigación, permitiendo analizarlos para luego ser usados en el desarrollo del proyecto, como en la selección de la programación a realizar para los controladores, junto con el método para trasladar dichos modelos desde un ambiente computacional hacia un entorno de programación para PLC y su aplicación en un medio físico, además de tener una idea general de la planta industrial sobre la cual se desarrollará el proyecto.

4.1. Análisis de resultados

4.1.1. Descripción de la planta industrial

Es fundamental para el desarrollo del proyecto, conocer los componentes y el funcionamiento de la planta que se va a trabajar. Se utilizó la estación de trabajo compacta desarrollada por (Muñoz & Pérez, 2016), el cual es un módulo didáctico para la regulación de variables de control: caudal, presión, nivel y temperatura en lazo cerrado, a través de un PLC Siemens S7-1200, mediante la manipulación de velocidad de una bomba centrífuga. Su estructura implementada se presenta en la Figura 33. Este modelo cuenta con los siguientes implementos, que se detallan en el plano P&ID de la Figura 34:

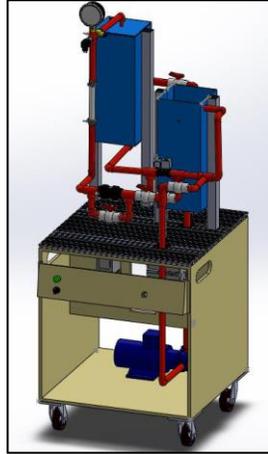


Figura 33. Módulo didáctico utilizado
Fuente: (Muñoz & Pérez, 2016)

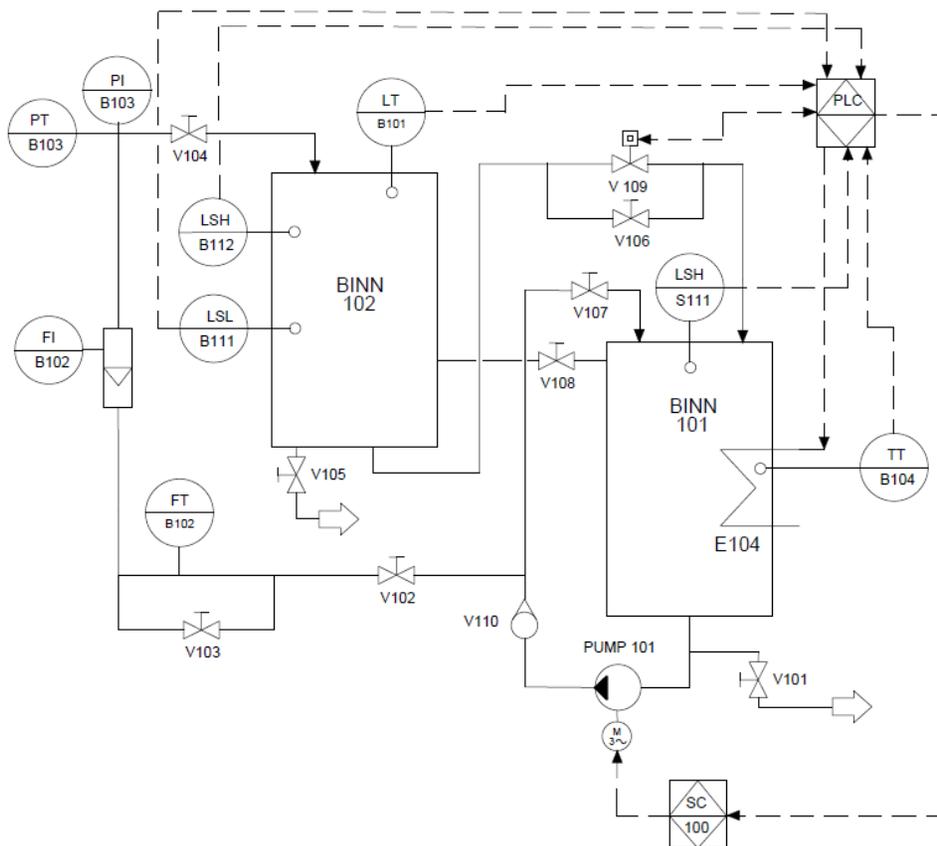


Figura 34. Plano P&ID
Fuente: (Muñoz & Pérez, 2016)

- **BINN 101, BINN 102** Esta nomenclatura corresponde a los tanques uno y dos respectivamente los mismos se utilizan para almacenar el líquido.
- **PUMP 101** Bomba centrífuga que suministra de fluido al sistema, se hará referencia a este elemento en adelante como P101.
- **E104** Elemento calefactor que eleva la temperatura del líquido en el BINN B101.
- **V101, V102, V103, V104, V105, V106, V107 y V108** Válvulas manuales que permiten o cierran el paso del fluido por las tuberías del módulo.
- **V109** Electroválvula 2/2 el cual permite el paso del fluido del BINN B102 hacia el BINN B101.
- **V110** Válvula antirretorno el cual impide el retorno del fluido a la bomba.
- **FT B102** Transmisor de flujo.
- **PI B103** Indicador de presión.
- **PT B103** Transmisor de presión.
- **FI B102** Rotámetro para indicar el nivel de caudal.
- **LT B101** Transmisor de nivel de líquido.
- **LSL B111** Interruptor de líquido de nivel bajo.
- **LSH B112** Interruptor de líquido de nivel alto.
- **LSH S111** Interruptor flotador para de nivel de líquido alto.
- **TT B104** Transmisor de temperatura.

4.1.2. Sensores utilizados

El módulo didáctico utilizado cuenta con 4 procesos manipulables: caudal, presión, nivel y temperatura; por lo cual se tiene 1 sensor de cada tipo para su utilización en los

procesos de control. Primeramente, para poder detectar el nivel de agua del BINN 102 se utiliza un sensor ultrasónico UB800-18GM40 (véase Figura 35), gracias al cual se realiza el control del lazo de nivel y poder controlar y monitorear dicha variable; el rango de detección máxima es de 800 mm y funciona a un voltaje mínimo y máximo de 10 VCD a 30 VCD respectivamente. (Muñoz & Pérez, 2016)



Figura 35. Sensor ultrasónico UB800-18GM40
Fuente: (Muñoz & Pérez, 2016)

El siguiente proceso es el de temperatura, para el cual se utiliza un sensor PT100 a tres hilos, como el de la Figura 36. Consiste en un alambre de platino que a 0 °C tiene 100 ohms y que al aumentar la temperatura aumenta su resistencia eléctrica. (Muñoz & Pérez, 2016)



Figura 36. Sensor de Temperatura PT100 de tres hilos
Fuente: (Muñoz & Pérez, 2016)

Luego, para la parte de caudal se utiliza un sensor de flujo YF-S201 (véase Figura 37), que mide el flujo de agua que pasa a través de la tubería desde la bomba hasta el BINN

102, que cuenta con un rango de medida amplio y una precisión aceptable. (Muñoz & Pérez, 2016)



Figura 37. Sensor de flujo YF-S201
Fuente: (Muñoz & Pérez, 2016)

Finalmente, en lo que respecta a presión, se utiliza un sensor de presión DANFOSS MBS33M (véase Figura 38) que cuenta con un rango de medida y una precisión aceptable



Figura 38. Sensor de Presión Danfoss MBS33M
Fuente: (Muñoz & Pérez, 2016)

4.1.3. Actuador

Para el control de las variables, se utiliza una bomba centrífuga controlada a través de un Variador de frecuencia, el cual manipula la frecuencia de entrada a la máquina entre los valores de 25 Hz a 60 Hz, ya que son los puntos operativos de la misma. Para ello, se tiene implementado un Convertidor SINAMICS V20 como el de la Figura 39.



Figura 39. Variador de frecuencia SINAMICS V20
Fuente: (Muñoz & Pérez, 2016)

4.1.4. Variable manipulada

De las cuatro variables manipuladas, el proceso de control de nivel es el que presenta un comportamiento no lineal, el cual servirá para contrastar el comportamiento de cada uno de los controladores que se desarrollaran durante el presente proyecto, por lo cual se detalla brevemente el proceso de nivel de la planta. En la Figura 40 se puede observar el P&ID del lazo de control de nivel. La bomba P101 distribuye el fluido almacenado en el BINN B101 hacia el BINN B102 por el sistema de tuberías, la señal de retroalimentación que se ingresará al controlador es obtenida del sensor ultrasónico de nivel (LT B101), las señales de perturbación pueden ser generadas mediante la apertura de la válvula manual V108 o controlando la válvula V109, esto para comprobar el funcionamiento y respuesta de los controladores implementados. También se utilizan sensores capacitivos los cuales marcaran el nivel de fluido alto (LSH B112) y bajo (LSL B111).

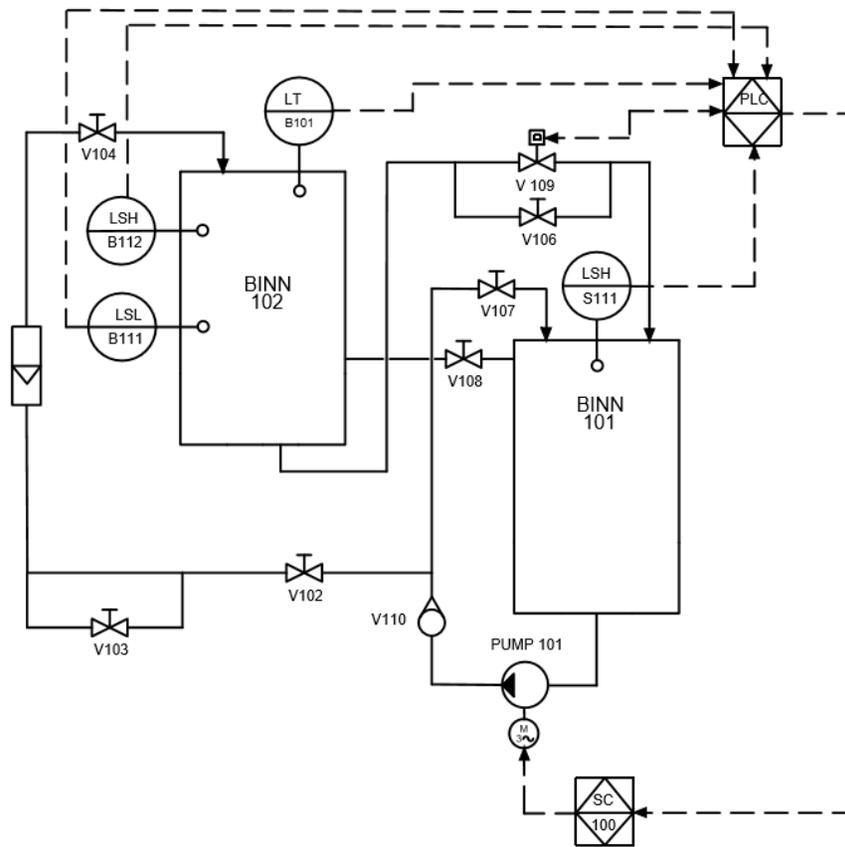


Figura 40. P&ID para el lazo de control de nivel
Fuente: (Muñoz & Pérez, 2016)

4.2. Comprobación de hipótesis

Para la comprobación de la hipótesis se analizan todos los resultados que se obtuvieron anteriormente, logrando así verificar que al evaluar el comportamiento de los controladores se lograra optimizar y mejorar los procesos productivos en sistemas distribuidos industriales, es decir, si es viable o no la implementación de los controladores.

4.2.1. Planteamiento de hipótesis

Con la evaluación del comportamiento de los controladores de tipo PID, difusos y neuronales se podrá determinar las características de operación de estos para la

optimización y mejora de procesos productivos en sistemas distribuidos industriales, utilizando una planta industrial ubicada en el Laboratorio de Hidrónica y Neutrónica de la Universidad de las Fuerzas Armadas ESPE Extensión Latacunga, para poder realizar el análisis de validación en base a los datos exportados de los equipos.

4.2.2. Viabilidad del proyecto

Con el análisis realizado se da como resultado que el proyecto es viable, permitiendo comenzar con el diseño de los controladores PID, difuso y neuronal; y a la vez luego de su implementación se realizara las mediciones del comportamiento para estableces sus características propias de cada controlador, y las diferencias entre sí.

CAPÍTULO V

5. PROPUESTA

Caracterización y evaluación de controladores PID, Lógica Difusa y Redes Neuronales de sistemas distribuidos industriales en el Laboratorio de Hidrónica y Neutrónica de la Universidad de las Fuerzas Armadas ESPE Latacunga

5.1. Datos informativos

Nombre de la empresa: Universidad de las Fuerzas Armadas – ESPE Extensión Latacunga.

Dirección: Provincia de Cotopaxi, Cantón Latacunga, Calle Quijano y Ordoñez y Hermanas Páez, Laboratorio de Fuentes Alternativas de Energía.

Tipo de empresa: Educación

Tamaño de la empresa: Grande

5.1.1. Beneficiarios

Estudiantes de la carrera de Ingeniería Electromecánica, en el campo de Automatización y Sistemas de Control.

5.2. Objetivos

5.2.1. Objetivo General

Caracterizar y evaluar los lazos de control PID, lógica difusa y redes neuronales de sistemas distribuidos industriales en el laboratorio de Hidrónica y Neutrónica de la Universidad de las Fuerzas Armadas ESPE.

5.2.2. Objetivos Específicos

- Investigar acerca de las características y parámetros de los controladores PID, así como las características de los controladores inteligentes difusos (Fuzzy) y redes neuronales artificiales.
- Diseñar y comparar el desempeño de cada uno de los controladores PID, lógica difusa y redes neuronales respecto a los otros, utilizando para ello una aplicación industrial de control multivariable.
- Establecer las ventajas y desventajas de cada uno de los tipos de controladores usados dentro de un sistema distribuido industrial.

5.3. Fundamentación propuesta

La propuesta que se presenta en este capítulo está relacionado con el diseño y la implementación de tres controladores diferentes, un PID, difuso y un neuronal, los mismo que permitirán la manipulación de la variable de nivel de una planta industrial presente en el Laboratorio de Hidrónica y Neutrónica, con lo cual se obtendrán datos de comportamiento y respuesta del controlador a diferentes valores de consigna; luego de ello se establecerán las características de cada tipo de controlador así como las ventajas

que presentan. Para ello también es necesario realizar simulaciones de cada tipo de control dentro de un modelo que se base en la planta industrial disponible.

5.4. Diseño de la propuesta

A continuación, se define el formato a considerar de acuerdo con la propuesta.

- **Obtención del modelo simulado de la planta:** Para el desarrollo de cada uno de los controladores a través de software, es necesario tener un modelo simulado que represente el comportamiento real de la planta, para lo cual, a través de las mediciones de la respuesta en estado estable, se procesará y desarrollara el modelo para su futuro uso.
- **Desarrollo de los controladores:** Una vez que se tenga el modelo de la planta, se procederá a diseñar cada uno de los controladores utilizando una herramienta de desarrollo como lo es Matlab, en donde se simulara para comprobar que no existan problemas con cada uno de los controles establecidos.
- **Implementación de los controladores:** Cuando se ha comprobado el funcionamiento de los controladores desarrollados, se los implementará físicamente en el PLC con el que cuenta la planta industrial, para luego obtener mediciones de la respuesta de cada controlador a diferentes señales de consigna.
- **Procesamiento de datos:** Con las mediciones obtenidas, se procederá a exportar los valores hacia el software Matlab, para obtener los valores de caracterización de cada respuesta según el tipo de controlador.

- **Resultado:** Determinación de las características y ventajas de cada tipo de controlador.

5.5. Toma de datos

Para el desarrollo de cada uno de los controladores, es necesario simular su respuesta antes de cargarlo físicamente a la planta, por lo cual se requiere modelar el comportamiento del proceso a controlar; para ello se realiza una serie de mediciones del proceso, en el cual se obtendrá los valores de entrada y salida del lazo de nivel, cuando éste se encuentra en lazo abierto.

Para llevar a cabo esto, se diseñó una programación para el PLC que comanda la planta, el mismo que tiene como función, almacenar en un Datalog los valores de la medición obtenida por el sensor ultrasónico (véase Figura 41) y la señal de salida al variador de frecuencia (véase Figura 42); se toma un periodo de operación de aproximadamente 60 minutos donde cada 10 segundos se hizo una variación aleatoria de consigna entregada al variador entre valores de 0% a 100% para observar la variación en el nivel. Dichos valores serán exportados a un archivo separado por comas (CSV), desde el cual se manipularán los datos para su posterior análisis.

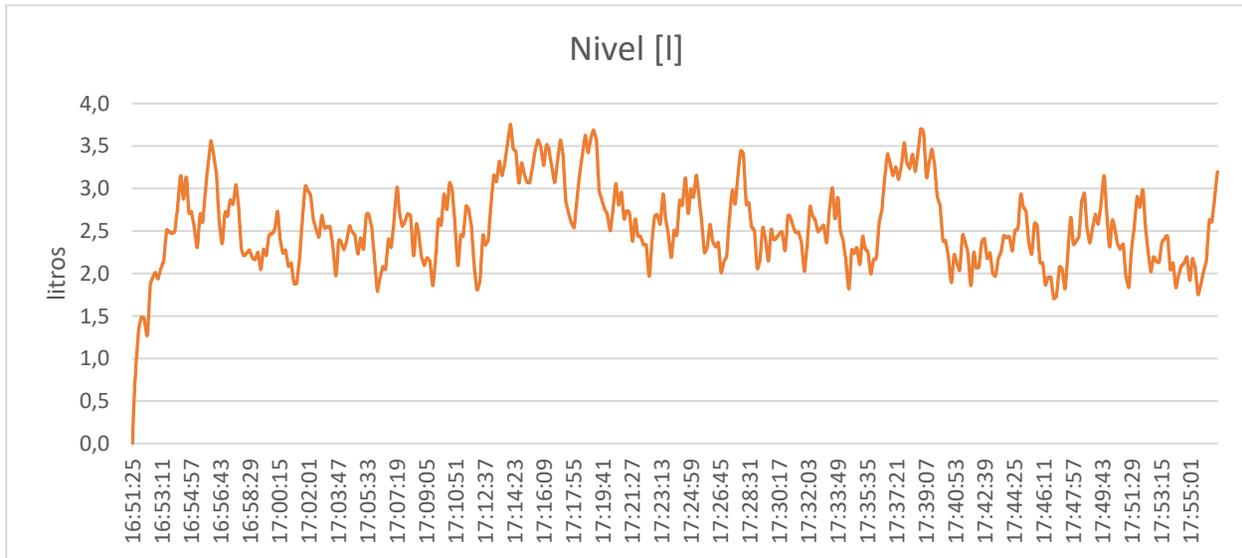


Figura 41. Señal de nivel en función del tiempo

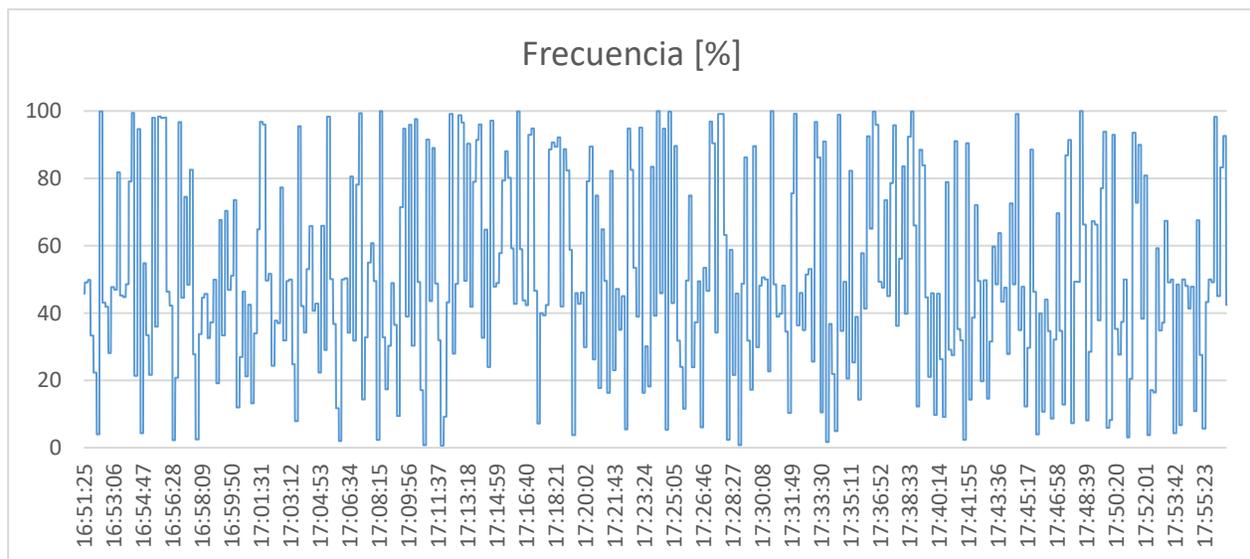


Figura 42. Salida del variador de velocidad en función del tiempo

Como se puede observar en la Figura 41, los valores de nivel obtenidos durante la primera medición no sobrepasan el nivel de 4 litros, por lo que no representa una observación clara de todo el rango posible del sistema, que es de 0 a 10 litros. Teniendo en cuenta este detalle, es necesario realizar una segunda medición corrigiendo el tiempo de muestreo a 0,5 seg., y a su vez ampliando el tiempo de cambio aleatorio de la consigna

de frecuencia a 20 seg. Con estas características, se obtuvo las mediciones de nivel y frecuencia que se pueden observar en la Figura 43 y Figura 44.

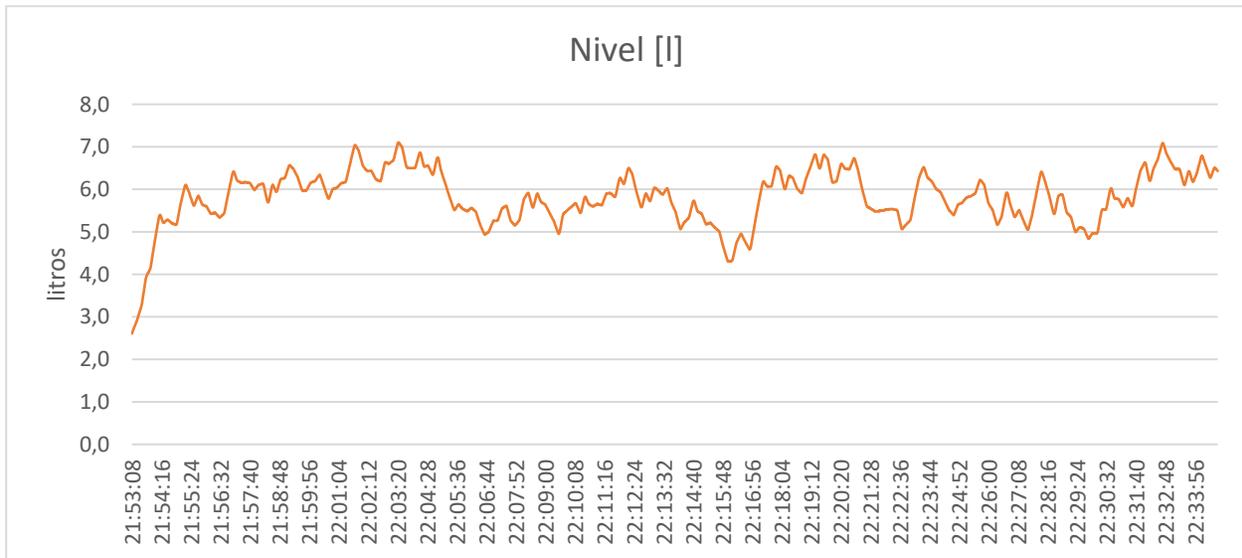


Figura 43. Señal de nivel para la segunda medición

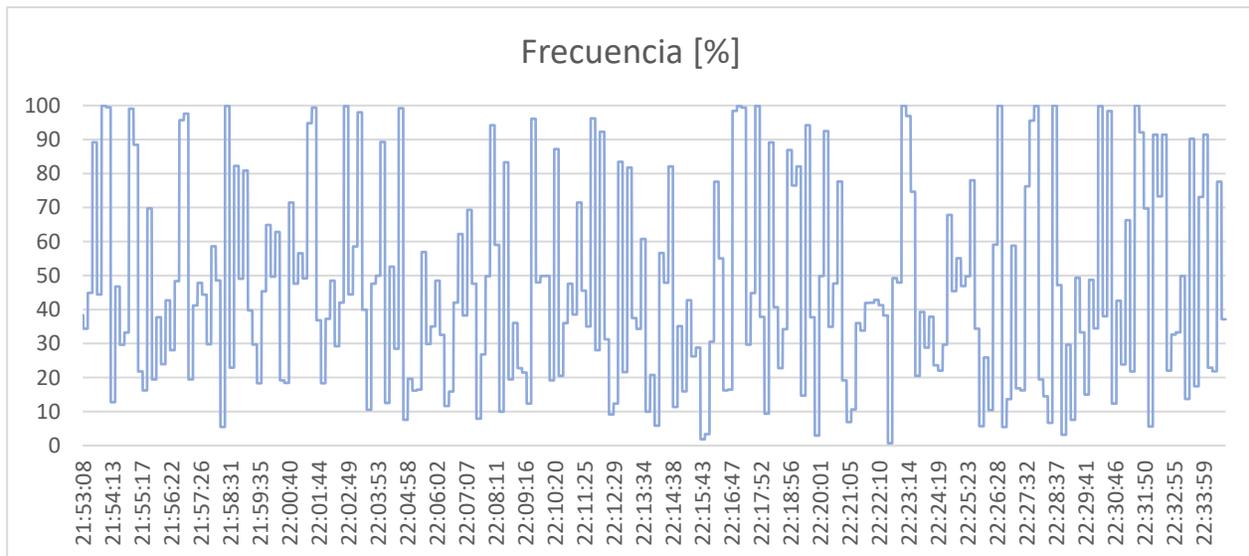


Figura 44. Salida del variador de velocidad para la segunda medición

En la segunda medición, se obtuvo un rango de valores más amplio de respuesta de la planta, llegando a valores cercanos al 8, por lo cual dichos datos serán usados para generar el modelo matemático en Matlab.

5.6. Generación del modelo matemático del proceso de nivel

Con estos valores de comportamiento se procede a exportarlos al software Matlab, desde el cual, a través del Control System Toolbox, se realizará la identificación del modelo de la planta. Con esta herramienta, se puede crear a través de valores medidos de entrada-salida de un sistema, un modelo matemático que permite emular un sistema físico. El complemento PID Tuner (Figura 45), con el cual se generará el modelo matemático, consta de tres pasos principales: la importación de los datos de prueba de entrada-salida, la identificación del modelo de la planta y el uso del modelo generado dentro del entorno.



Figura 45. Complemento PID Tuner

Dentro del entorno de Matlab, se realiza la importación de los datos generados por el Datalog hacia el espacio de trabajo (workspace) como se muestra en la Figura 46.

MedicionNivel									
Record	Date	UTTime	Nreal	Ndata	Orear	Odata	VarName8	VarName9	VarName10
1	7/23/2018	0.7024	-363.3985	37	45774380	12656		-3.6340e-04	45.7744
2	7/23/2018	0.7024	4.3608e+03	50	45774380	12656		0.0044	45.7744
3	7/23/2018	0.7024	1.6902e+04	68	45774380	12656		0.0109	45.7744
4	7/23/2018	0.7024	1.5990e+05	478	49105400	13377		0.1999	49.1054
5	7/23/2018	0.7024	2.9913	748	49105400	13377		0.2590	49.1054
6	7/23/2018	0.7024	341958	979	49105400	13377		0.2400	49.1054
7	7/23/2018	0.7024	4.0337e+05	1148	49105400	13377		0.4034	49.1054
8	7/23/2018	0.7024	1.6325e+06	1380	49105400	13377		0.4034	49.1054

Figura 46. Importación de datos al modelo de Matlab

Con los datos almacenados en variables separadas, una para los valores de entrada y otra para los valores de salida, se debe dirigir a la pestaña de Apps donde se busca el complemento PID Tuner, como se muestra en la Figura 47.

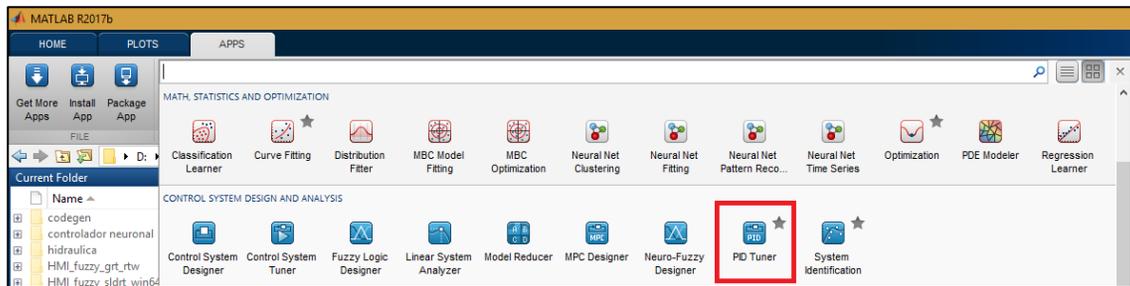


Figura 47. Acceso al complemento PID Tuner

En la ventana que se despliega, se dirige a la importación de datos para generar una nueva planta a través de los datos de entrada-salida, como se ve en la Figura 48.

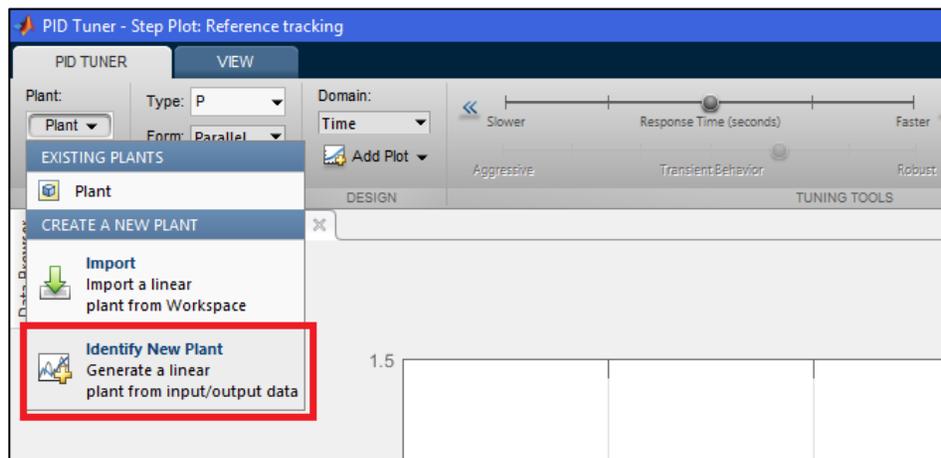


Figura 48. Identificación de nueva planta

Para la importación de los datos en la aplicación, existen diferentes formas de respuesta, de entre las cuales se selecciona Datos I/O arbitrarios. Aquí se detalla el nombre de las variables que contienen los valores de datos de entrada como los de

salida, así como el tiempo de inicio de la medición y el tiempo de muestreo de esta, como se indica en la Figura 49.

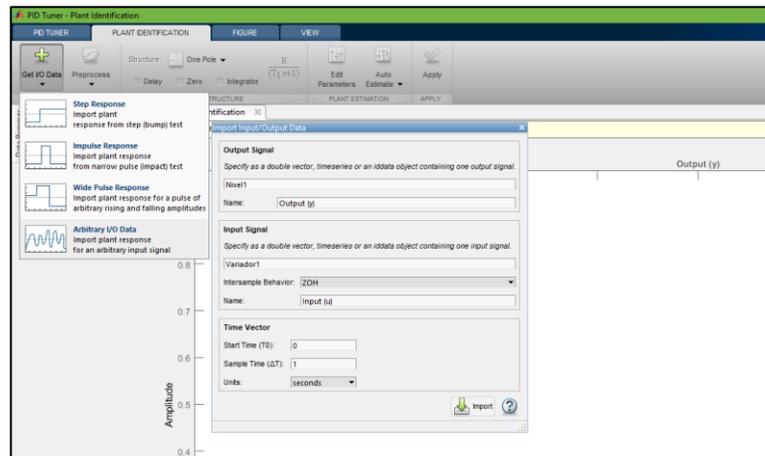


Figura 49. Configuración de los parámetros de importación

Realizada la importación, se genera una curva que representa los datos ingresados, y otra que es el resultado del proceso de identificación de la planta, el cual puede ser manipulado para que se ajuste a una estructura definida como: un polo, dos polos o un modelo de espacio de estado. En este caso se selecciona un espacio de estado, donde se realiza una estimación automática, y visualizaremos el resultado en la Figura 50.

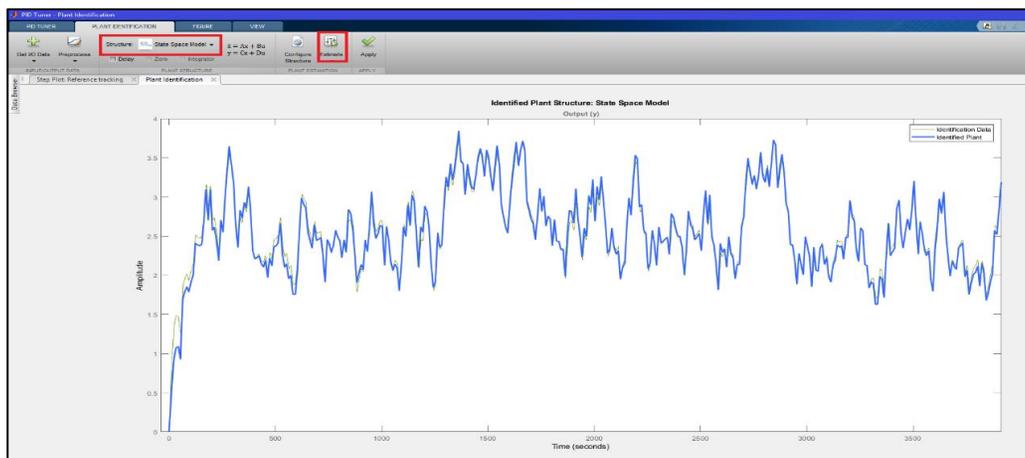


Figura 50. Resultado de la estimación automática

Para que la curva generada se ajuste más a los parámetros medidos, se puede realizar una modificación en el orden de la planta y volver a realizar la estimación, hasta que el resultado sea aceptable, así como se visualiza en la Figura 50. Luego de esto, se acepta el modelo final.

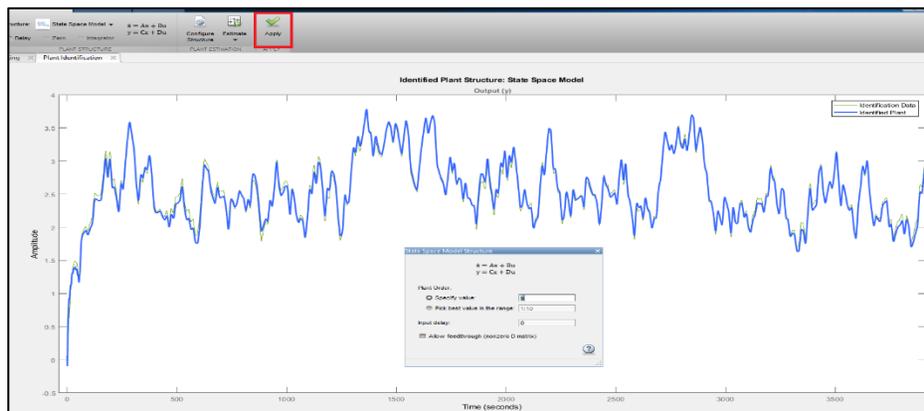


Figura 51. Resultado del espacio de estado de nivel 8

Finalmente, se exporta el modelo de la planta generado hacia el workspace para sus futuras aplicaciones, según se muestra en la Figura 52.

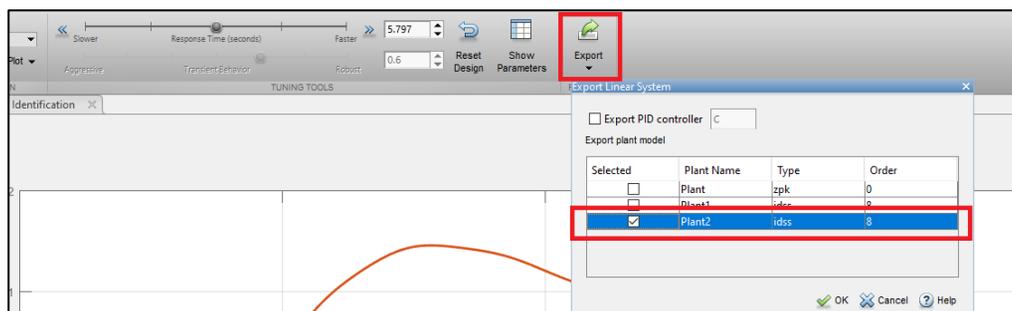


Figura 52. Exportación del modelo de planta generado

Este modelo será utilizado posteriormente para el desarrollo de los controladores mediante la parte de Simulink, donde se trabajará con este modelo a través de un objeto de tipo 'LTI System', como se indica en la Figura 53.

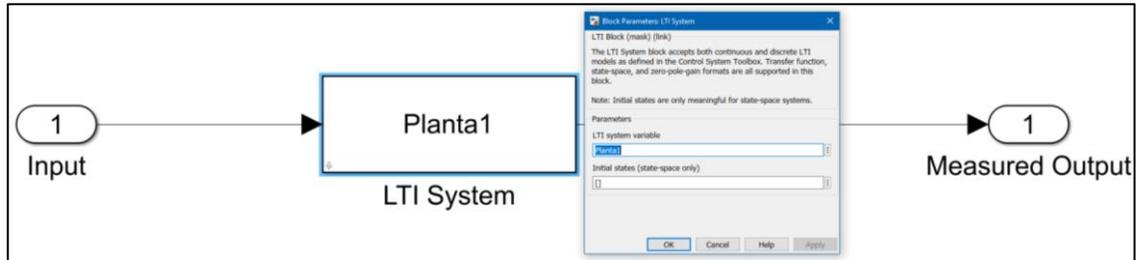


Figura 53. Implementación del modelo matemático en Simulink

5.7. Desarrollo del controlador PID

Considerando que un control de tipo PID forma parte del control convencional, y del cual se encuentran herramientas prediseñadas para su diseño, no se verá con mayor detalle explicativo el proceso, solamente se dará énfasis a la parte de sintonización mediante el software Matlab.

Para el diseño del control mediante PID, se lo realizará utilizando los bloques predefinidos de la herramienta Simulink y del complemento PID Tuner, que permitirá sintonizar el controlador para obtener una respuesta que provoque en la planta la salida deseada; por lo cual tenemos primeramente que abrir una nueva hoja de trabajo como se muestra en la Figura 54.

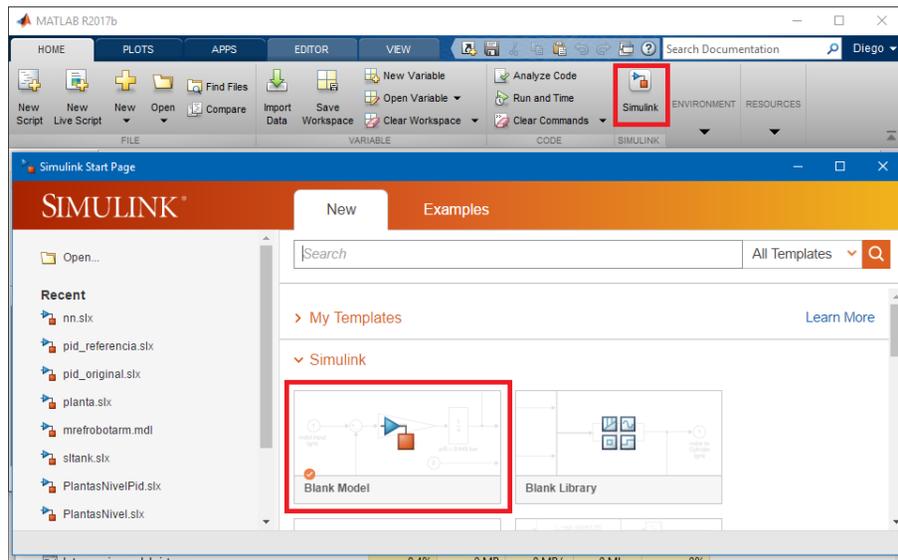


Figura 54. Ingreso hacia el entorno de trabajo de Simulink

Aquí se grafica un modelo de control en lazo cerrado típico, como el que se muestra en la Figura 55, que consta de una consigna de entrada, en este caso una función escalón, el cálculo del error entre éste y la planta, el controlador PID y la planta en sí.

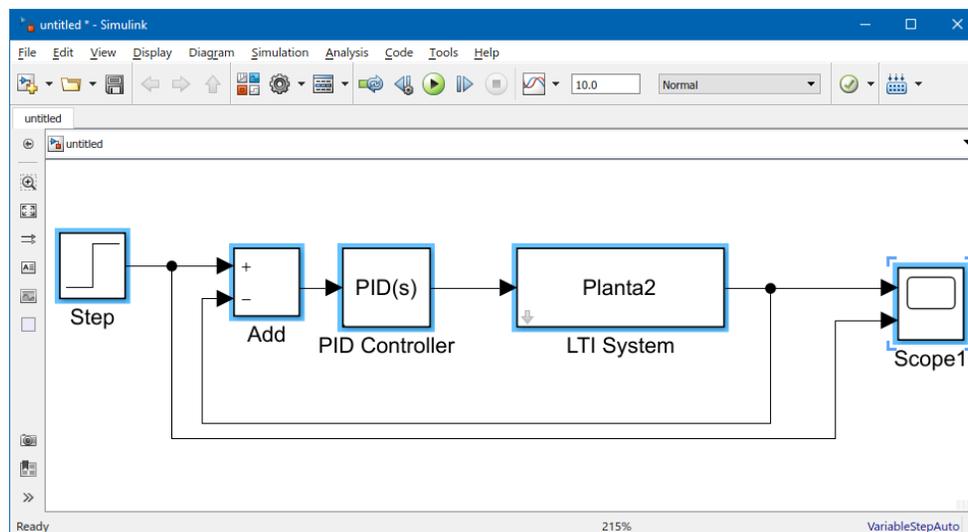


Figura 55. Diagrama en lazo cerrado para el controlador PID

Se abre las propiedades del bloque 'PID Controller', donde se encuentran los parámetros proporcional, integral y derivativo, como muestra la Figura 56. Estos valores

pueden ser modificados manualmente, pero para efecto de simplificación del procedimiento, se utilizó el sintonizador de PID Tuner.

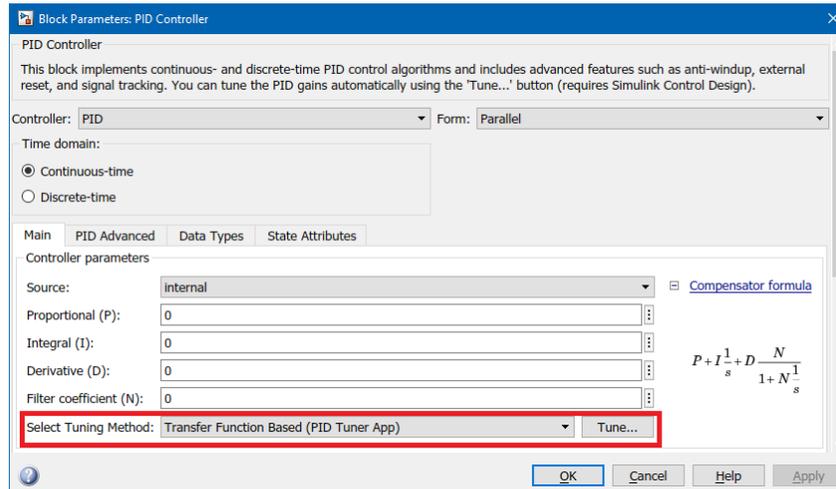


Figura 56. Propiedades del bloque PID

El complemento PID Tuner ajusta automáticamente las ganancias de un controlador PID para una planta SISO (1 entrada – 1 salida) para lograr un equilibrio entre rendimiento y robustez; se puede especificar el tipo de controlador, como PI, PID con filtro derivado y otros varios, además se puede optimizar interactivamente el rendimiento del controlador para ajustar el ancho de banda del lazo y el margen de fase, o para favorecer el seguimiento del punto de ajuste o el rechazo de perturbaciones.

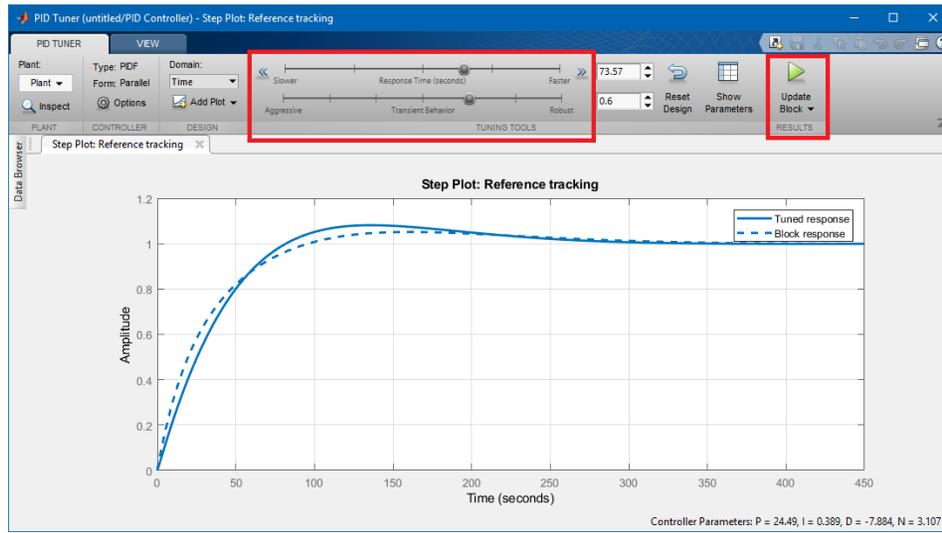


Figura 57. Sintonización del bloque PID mediante PID Tuner

Cuando se visualiza la aplicación, como en la Figura 57, se puede manipular la respuesta del controlador al escalón para que se comporte de la manera que se desee, variando su tiempo de respuesta y su comportamiento transitorio. Una vez que se tenga un comportamiento aceptable, se puede actualizar el bloque de PID con los nuevos parámetros, como indica la Figura 58.

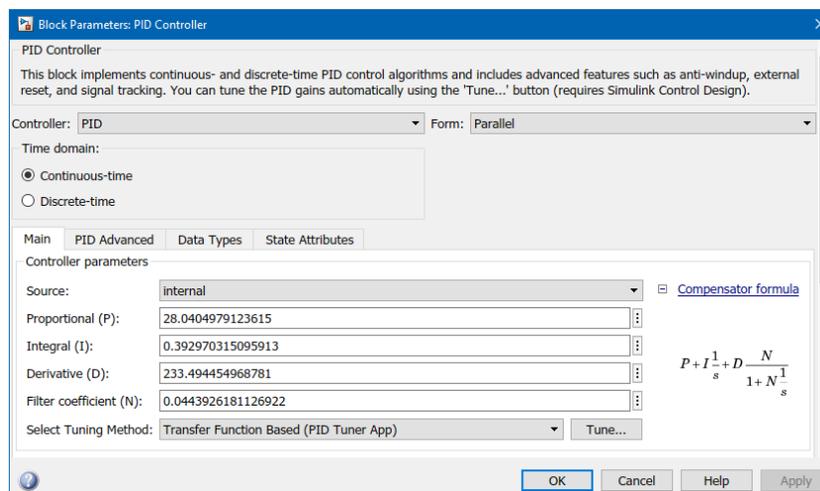


Figura 58. Parámetros finales del bloque PID

5.7.1. Pruebas de funcionamiento del bloque generado

Una vez completado la parametrización del bloque, se procede a simular el comportamiento del controlador, primero con una señal de tipo escalón (Figura 59), y luego con una serie de consignas aleatorias (Figura 60). En estas gráficas, la línea verde (Step/Uniform Random Number) representa la consigna o setpoint deseado, y la línea roja (LTI System) representa la salida del modelo de planta generado.

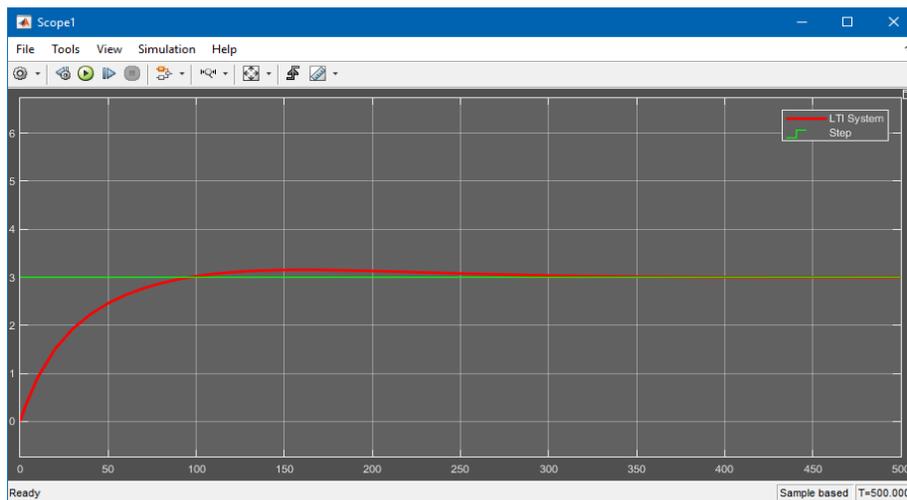


Figura 59. Respuesta del controlador PID al escalón

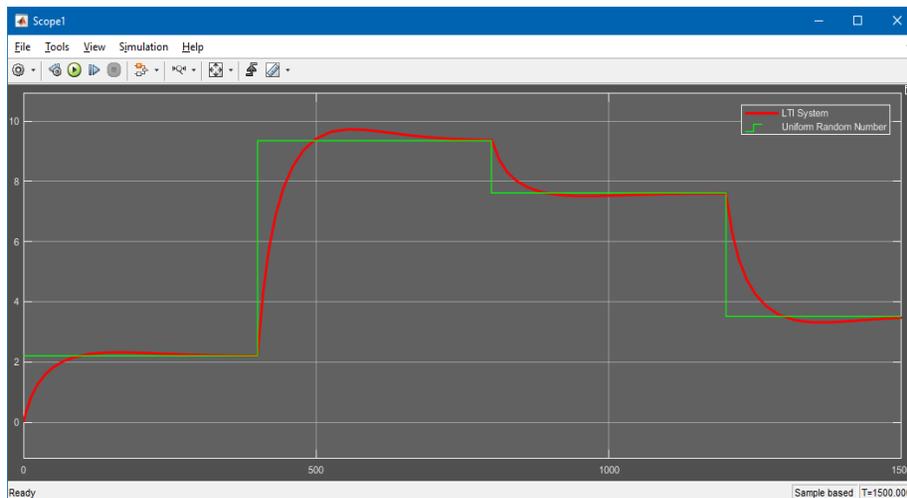


Figura 60. Respuesta del controlador PID a valores de consigna aleatorios

Como se puede observar, en la respuesta al escalón, el controlador provoca una señal de salida para que la planta se estabilice, lográndolo en un tiempo relativamente avanzado, lo que es dependiente del modelo de la planta. En cuanto a la respuesta de la consigna variable, el controlador logra que la planta llegue al valor establecido en cada uno de los valores aleatorios.

5.8. Desarrollo del controlador difuso

Para el desarrollo de este controlador, es importante tener en cuenta ciertos detalles de su aplicación. Como variables de entrada se tendrá, según el modelo previsto por (Guzmán & Castaño, 2006), el error entre la consigna deseada y el valor actual de la planta, y la variación de dicho error; como salida se tiene el valor de frecuencia para el variador de velocidad. Cada uno de los parámetros detallados en este punto fueron colocados empíricamente, según el conocimiento propio del comportamiento real de la planta y del proceso, por lo cual los valores de cada función de membresía, así como su tipo, fueron ajustados luego de varias simulaciones hasta obtener un comportamiento adecuado.

5.8.1. Fuzzy Logic Toolbox

Para el desarrollo de este controlador, se utiliza el complemento “Fuzzy Logic Toolbox” de Matlab y Simulink, mediante el cual se puede analizar, diseñar y simular sistemas basados en lógica difusa; donde se puede realizar la inferencia difusa y aplicarla dentro de un modelo complejo que representa un sistema dinámico. Para acceder a la aplicación de diseño de sistemas difusos (Fuzzy Logic Designer o FLD) se lo hace a través de la

ventana de comandos de Matlab, con la utilización del comando “fuzzy”, como se muestra en la Figura 61.

Mediante esta aplicación, se puede desarrollar todo un controlador de carácter difuso, desde la selección del tipo de controlador (Mamdani o Sugeno), así como añadir las variables de entrada y salida necesarias para el mismo; también permite crear el conjunto de reglas, definir las funciones de membresía y analizar el comportamiento de un sistema de inferencia difusa (FIS).

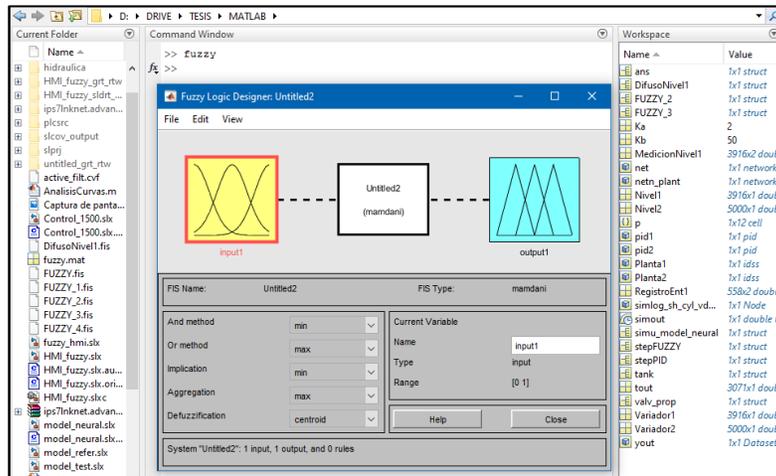


Figura 61. Aplicación Fuzzy Logic Designer FLD

La interfaz gráfica del toolbox cuenta con los siguientes elementos:

- **Editor FIS:** Muestra información general sobre un sistema de inferencia difusa.
- **Editor de funciones de membresía:** Permite mostrar y editar las funciones de membresía asociadas con las variables de entrada y salida del FIS.
- **Editor de reglas:** Permite ver y editar reglas difusas usando notación simbólica.

- **Visor de reglas:** Permite ver el comportamiento detallado de un FIS para ayudar a diagnosticar el comportamiento de reglas específicas o estudiar el efecto de cambiar las variables de entrada.
- **Visor de superficie:** Genera una superficie tridimensional a partir de dos variables de entrada y la salida de un FIS.

5.8.2. Variables lingüísticas

5.8.2.1. Error

La variable lingüística “error” proviene de la diferencia entre la consigna o setpoint a la que se desea llegar y el valor medido. El error es positivo si el nivel del líquido se encuentra por debajo del valor de la consigna; y es negativo si el nivel del líquido se encuentra por encima de dicho valor.

Tabla 1

Descripción de las funciones de membresía de la variable "error"

Función	Tipo de curva	Descripción	Parámetros
Alto+	Gaussiana	Nivel del tanque muy elevado	[0.349 -0.9889]
Alto-	Gaussiana	Nivel del tanque ligeramente elevado	[0.2 -0.5132]
OK	Gaussiana	Nivel del tanque en el valor deseado	[0.08763 0]
Bajo-	Gaussiana	Nivel del tanque ligeramente disminuido	[0.2 0.5185]
Bajo+	Gaussiana	Nivel del tanque muy disminuido	[0.38 0.9793]

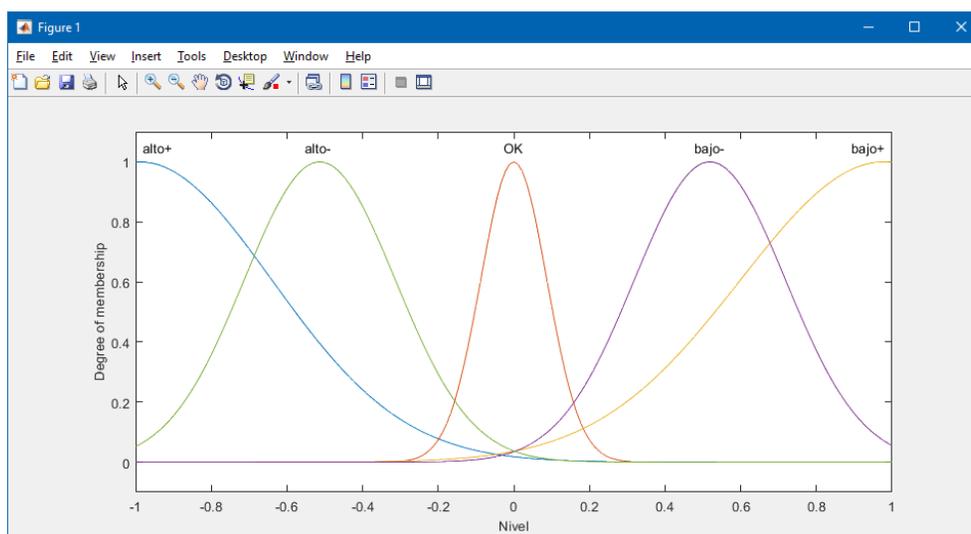


Figura 62. Funciones de membresía variable “error”

Para valorar el error se establecen 5 funciones de membresía comprendidas en un rango de +/- 1, como se visualiza en la Figura 62. En la Tabla 1 se indica los parámetros y características de diseño de cada una de las funciones de membresía, los cuales son producto de la experimentación con la planta real.

5.8.2.2. Variación del error

La variable lingüística “variación” proviene de la derivación de la señal del error, en otras palabras, de la razón de cambio en el error, que es igual al negativo de velocidad con la que cambia el valor medido en el tanque.

Tabla 2

Descripción de las funciones de membresía de la variable "variación"

Función	Tipo de curva	Descripción	Parámetros
Negativa	Gaussiana	Variación de nivel de subida	[0.003 -0.01]
Nada	Gaussiana	Sin variación de nivel	[0.0009662 0]
Positiva	Gaussiana	Variación de nivel de bajada	[0.003033 0.01]

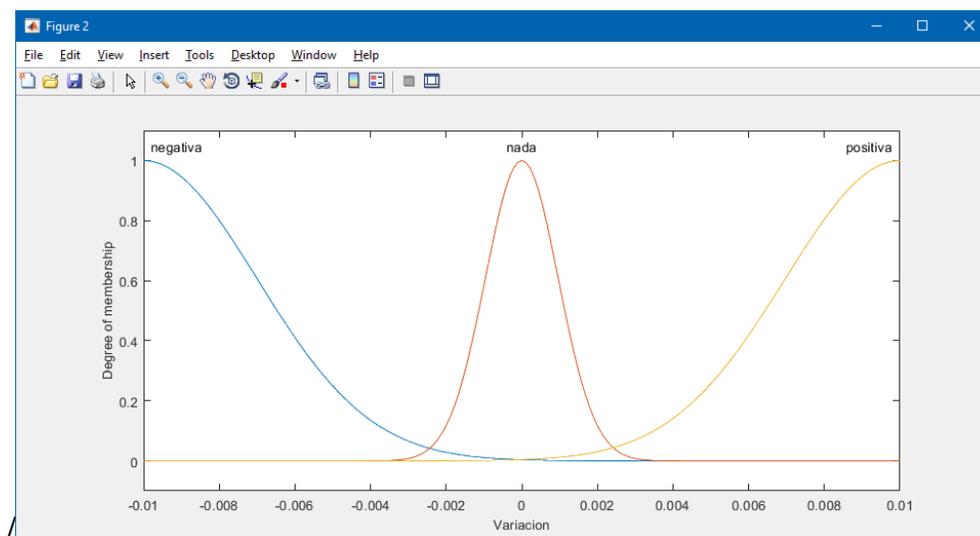


Figura 63. Funciones de membresía variable “variación”

Para valorar la variación del error se establecen 3 funciones de membresía comprendidas en un rango de ± 0.01 , como se visualiza en la Figura 63. En la Tabla 2 se indica los parámetros y características de diseño de cada una de las funciones de membresía, los cuales son producto de la experimentación con la planta real.

5.8.2.3. Salida

La variable lingüística “frecuencia” es la respuesta del controlador a las dos variables de entrada; ésta provee de una señal que sirve de consigna de variación para el control de velocidad del actuador, en otras palabras, la señal indicará el valor que hay que aumentar o disminuir a la frecuencia de salida del variador de velocidad, lo que provoca una variación en la cantidad de fluido que ingresa al tanque, y, por ende, de su nivel.

Tabla 3

Descripción de las funciones de membresía de la variable "frecuencia"

Función	Tipo de curva	Descripción	Parámetros
Bajar+	Triangular	Disminuir la frecuencia un valor de 9	[-10 -9 -8]
Bajar-	Triangular	Disminuir la frecuencia un valor de 6.5	[-7.5 -6.5 -5.5]
Mantener	Triangular	No cambiar la frecuencia	[-1 0 1]
Subir-	Triangular	Aumentar la frecuencia un valor de 3	[2 3 4]
Subir+	Triangular	Aumentar la frecuencia un valor de 7	[6 7 8]

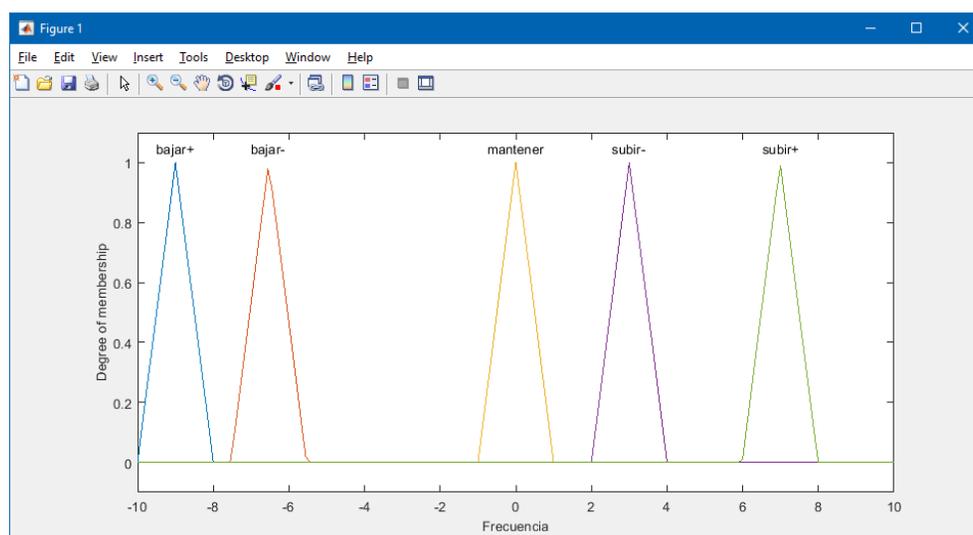


Figura 64. Funciones de membresía variable "frecuencia"

Para valorar la frecuencia de salida se establecen 5 funciones de membresía comprendidas en un rango de +/- 10, como se visualiza en la Figura 64. En la Tabla 3 se indica los parámetros y características de diseño de cada una de las funciones de membresía, los cuales son producto de la experimentación con la planta real.

5.8.3. Reglas de inferencia

Las reglas aplicadas requieren de un conocimiento de la planta a ser controlada, es decir, la forma en que ésta se comportaría en caso de realizar un cambio en alguna de

sus entradas. No existe una regla formal que indique el número de términos para cada variable lingüística que se debe usar, todo queda a discreción del diseñador del controlador que conoce el equipo, además de lo que se desea obtener como respuesta. Para el desarrollo de las reglas, se puede utilizar una o ambas variables de entrada, unidas mediante conectores lógicos AND y OR; cada regla tendrá como resultado una función de membresía de la salida.

Las reglas se basan en el lenguaje natural que puede manejar un operario de control de la planta: si el nivel se encuentre en el valor deseado, la frecuencia de salida del variador de velocidad no se deberá alterar; si el nivel se encuentra un poco bajo del valor deseado, habrá que subir la frecuencia un poco, y si se encontrara demasiado bajo, habrá que aumentar bastante la frecuencia. De igual manera se lo realizaría si el nivel se encontrara por encima del valor de consigna, disminuyendo un poco la frecuencia hasta que iguale el nivel a la consigna; y disminuyéndolo bastante si la diferencia fuera muy elevada. Cuando el nivel este cerca del valor deseado, si la variación en el error es negativa, se debe subir lentamente la frecuencia para evitar que por la inercia del proceso de llenado-vaciado de la planta se encuentre el nivel por debajo de la consigna, y al entrar en funcionamiento una regla inversa, provoque inestabilidad o bien oscilación alrededor de la consigna establecida. Todo este proceso lingüístico se transcribe en manera de reglas difusas hacia el Editor de Reglas del FLD (véase Figura 65) de la siguiente forma:

- Si el error es 'OK', entonces la frecuencia es 'Mantener'
- Si el error es 'Bajo+', entonces la frecuencia es 'Subir+'
- Si el error es 'Alto+', entonces la frecuencia es 'Bajar+'

- Si el error es 'OK' y la variación es 'Positiva', entonces la frecuencia es 'Bajar-'
- Si el error es 'OK' y la variación es 'Negativa', entonces la frecuencia es 'Subir-'
- Si el error es 'Bajo-' y la variación es 'Positiva', entonces la frecuencia es 'Bajar+'
- Si el error es 'Alto-' y la variación es 'Negativa', entonces la frecuencia es 'Subir+'

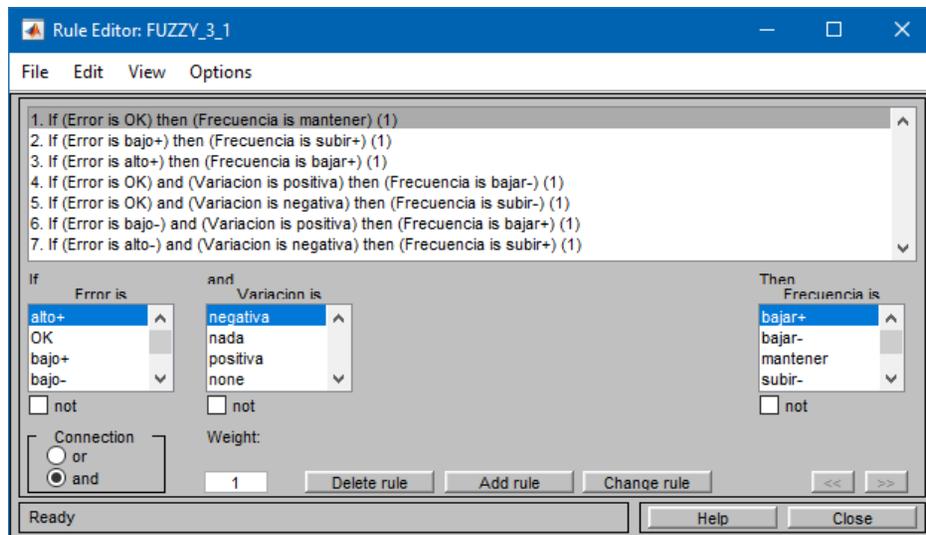


Figura 65. Reglas del sistema difuso desarrollado

Una vez desarrollado la estructura del sistema de inferencia difusa, se lo exporta hacia el workspace de Matlab para su implementación, como se observa en la Figura 66.

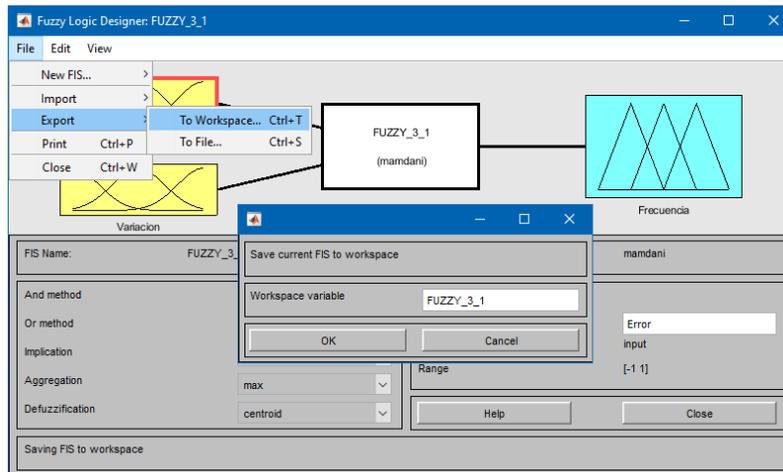


Figura 66. Exportación de la estructura difusa a Matlab

5.8.4. Puesta a prueba del controlador difuso

Para verificar el funcionamiento del controlador difuso desarrollado, se exportará al entorno de Simulink para visualizar su respuesta frente a la planta simulada. Para ello, dentro de una hoja de trabajo de Simulink se ingresa un bloque de tipo 'Fuzzy Logic Controller', desde la librería de Fuzzy Logic Toolbox, como se indica en la Figura 67.

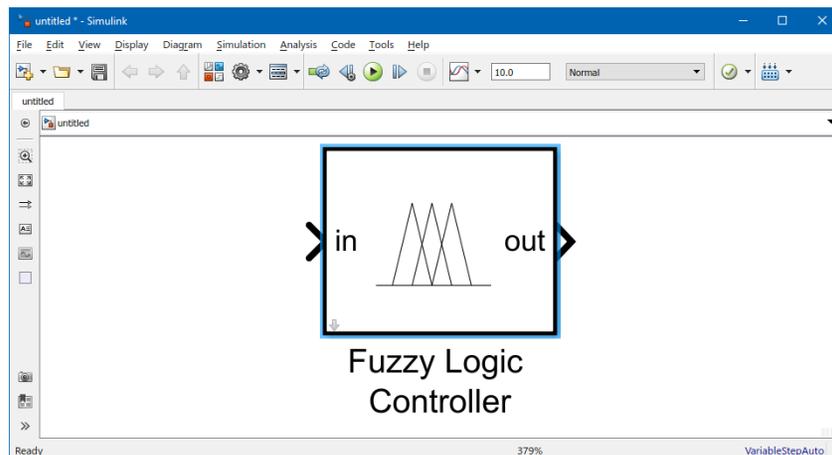


Figura 67. Bloque 'Fuzzy Logic Controller'

Dando doble clic sobre el bloque se despliega las propiedades (Figura 74), donde el único parámetro que se debe ingresar es el nombre de la variable FIS que se exportó anteriormente hacia el workspace.

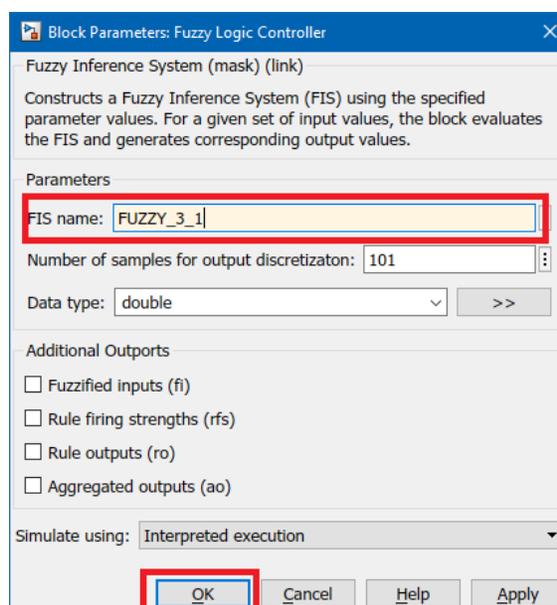


Figura 68. Propiedades del bloque 'Fuzzy Logic Controller'

Luego se procede a armar un modelo de control en lazo cerrado donde se tiene el ingreso del error y de su variación hacia el controlador, como se muestra en la Figura 69. Hay que tener en cuenta que, al ser un proceso discreto, es necesario usar bloques correspondientes a este tipo de dato. Otra consideración importante es asegurarse que los valores de entrada lleguen al controlador solamente con los rangos de valores establecidos en sus propias funciones de membresía, para evitar errores de cálculo debido a rangos no contemplados dentro del proceso difuso; para ello es necesario saturar las señales de entrada a los valores extremos correspondientes. También es necesario ingresar las variables en el orden que fueron declaradas, por ello el error debe ir en primer lugar y luego su variación.

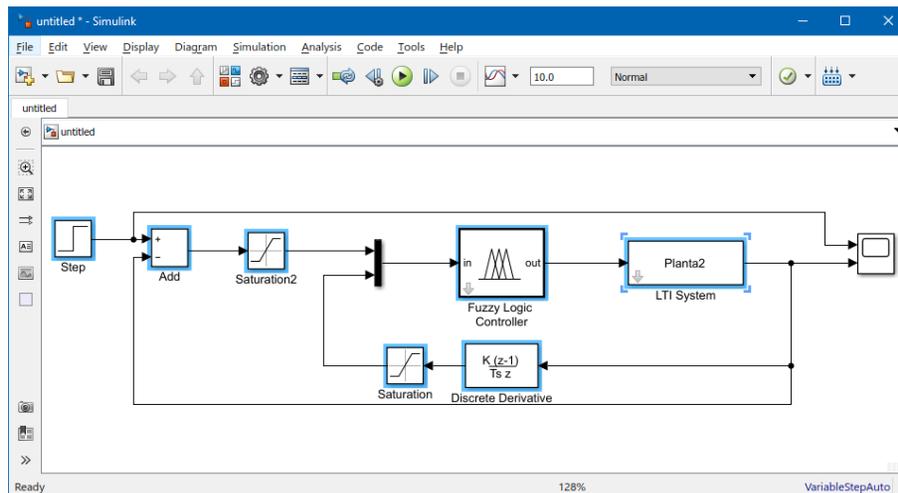


Figura 69. Diagrama del controlador difuso

En cuanto a la salida del controlador difuso, es necesario realizar un post-procesamiento de la señal obtenida, esto se debe al hecho que la señal obtenida por el proceso de inferencia difusa expresa la orden de “aumentar” o “disminuir” la velocidad de operación del variador de frecuencia, lo cual provocara una variación en el nivel de la planta; por lo que la propia señal de salida no es un indicativo de la velocidad exacta a la cual se debe colocar el actuador. Como se trata de una señal de tipo ‘mas’/’menos’, lo más conveniente es realizar una acumulación del valor de salida del controlador, lo que convertirá a la señal de salida en una consigna de frecuencia de entrada para el variador de velocidad.

Para la acumulación de la señal, la primera solución y la más obvia era la de realizar una integración de la señal de salida, como se visualiza en la Figura 70. Al momento de realizar pruebas de funcionamiento, se pudo observar que la velocidad de respuesta del bloque de integración era muy lenta, lo cual provocaba variaciones no deseables en la respuesta del controlador.

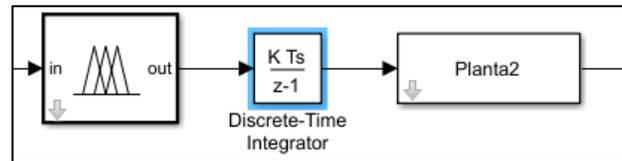


Figura 70. Acumulación por medio de integración.

Como segunda propuesta de solución del problema, se decidió emplear un sistema de memoria aditiva como el indicado en la Figura 71. Esto consiste en almacenar el valor de salida instantáneo del controlador en un tiempo t , y sumarlo al siguiente valor de proceso en $t + \Delta t$. Dicho resultado pasara a ser el nuevo valor acumulado que se sumara con siguiente valor de salida en $t + 2\Delta t$. Debido al proceso de acumulación, es indispensable que la señal de acumulación nunca esté por fuera del rango establecido para el actuador, ya que esto genera problemas con el variador de frecuencia, por lo cual esta señal también debe ser saturada.

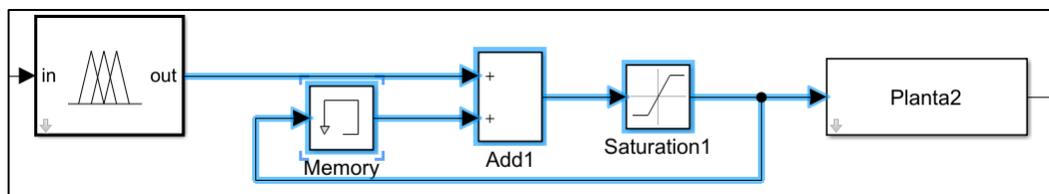


Figura 71. Acumulación por medio de memoria.

Una vez completado el diseño del diagrama, se procede a simular el comportamiento del controlador, primero con una señal de tipo escalón (Figura 72), y luego con una serie de consignas aleatorias (Figura 73). En estas gráficas, la línea verde (Step/Uniform Random Number) representa la consigna o setpoint deseado, y la línea roja (LTI System) representa la salida del modelo de planta generado.

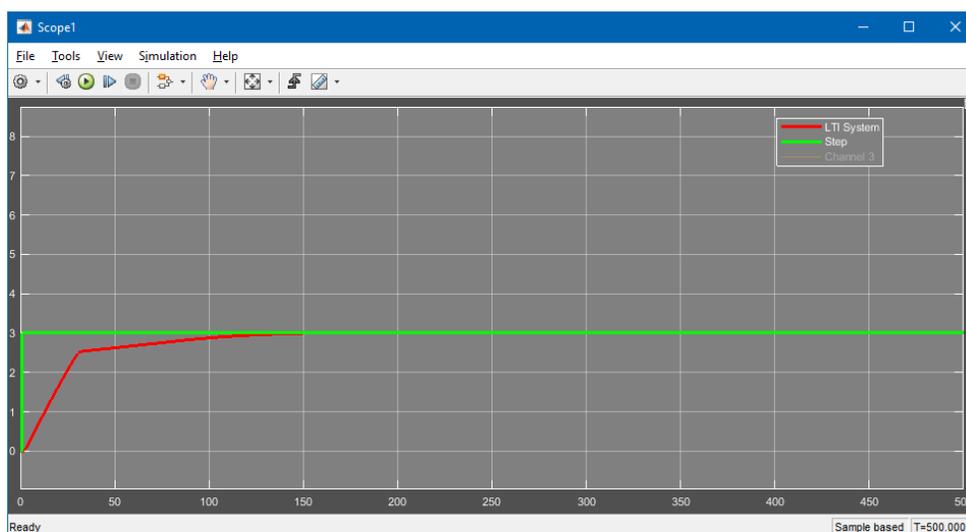


Figura 72. Respuesta del controlador difuso al escalón

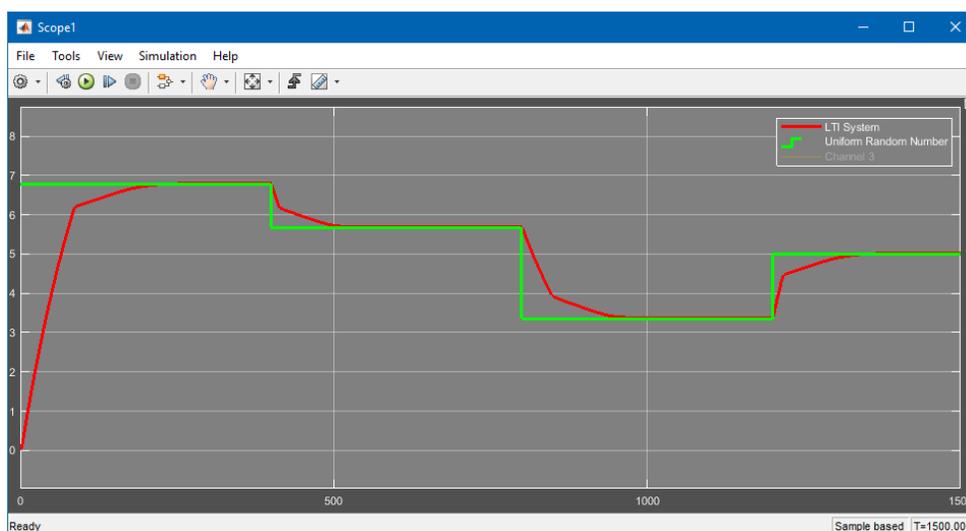


Figura 73. Respuesta del controlador difuso a valores de consigna aleatorios

Como se puede observar en ambas respuestas, el nivel de la planta crece rápidamente hasta acercarse al valor de consigna, donde reduce su velocidad para ajustarse suavemente al valor de setpoint. En la respuesta al escalón, el controlador provoca una señal de salida para que la planta se estabilice, lográndolo en un tiempo relativamente avanzado, lo que es dependiente del modelo de la planta. En cuanto a la respuesta de la

consigna variable, el controlador logra que la planta llegue al valor establecido en cada uno de los valores aleatorios.

5.9. Desarrollo del controlador Neuronal

Para desarrollar un controlador de tipo neural, que utiliza redes neuronales artificiales, es necesario realizar el entrenamiento de esta para que aprenda el comportamiento del proceso. El entrenamiento para la red neuronal se lo realiza mediante el software Matlab, a través de su herramienta Deep Learning Toolbox, el cual proporciona un marco de diseño e implementación de redes neuronales con algoritmos, modelos preestablecidos y aplicaciones. Entre sus principales usos se encuentra la clasificación y regresión de datos de imágenes, series temporales y texto. Para este caso, se usará la parte de Series Temporales y Sistemas de Control, donde se aplicará el Sistemas de control de redes neuronales para controlar sistemas no lineales utilizando redes neuronales de predicción de modelos, NARMA-L2 y referencia de modelos.

5.9.1. Datos de entrenamiento para la red

Para realizar el entrenamiento, primero tenemos que cargar los datos reunidos en el punto 5.6 como una estructura para el controlador. Para crear la estructura utilizamos el comando de Matlab 'struct', como se muestra en la Figura 74. Luego abrimos la variable, donde crearemos 4 diferentes campos, como se indica en la Tabla 4. Con clic derecho, seleccionamos 'New'; luego de ello cambiamos los nombres de las variables y se añade los valores correspondientes.

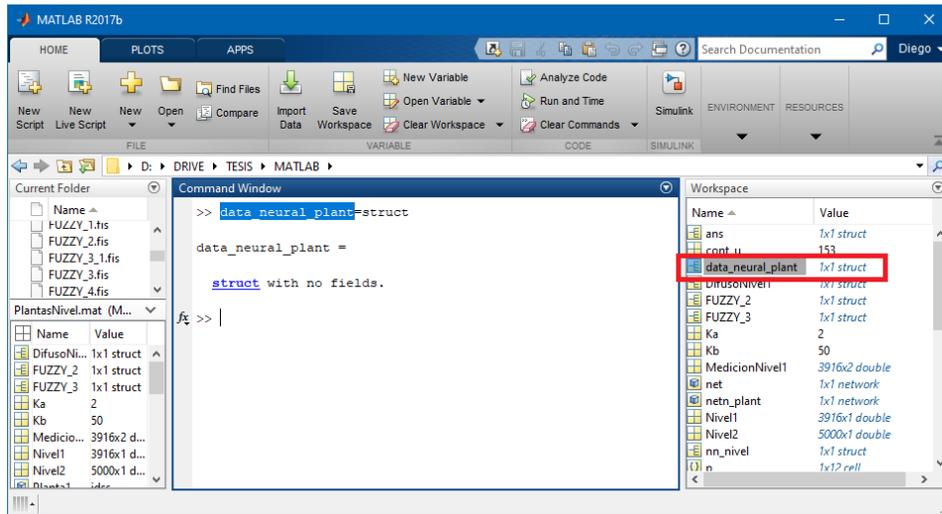


Figura 74. Creación de la estructura 'data_neural_plant'

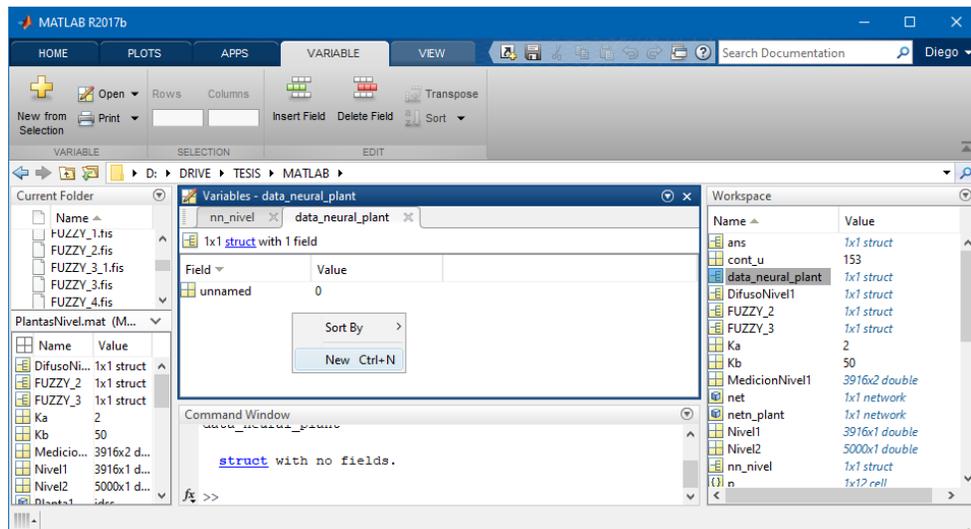


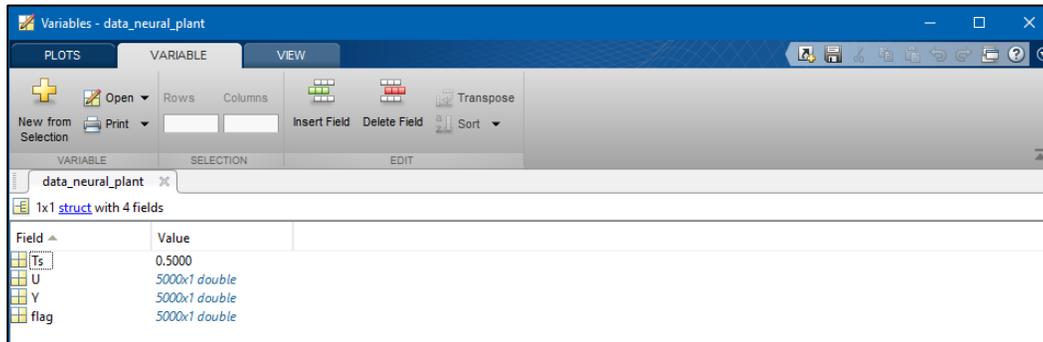
Figura 75. Creación de los campos en la estructura

Tabla 4

Descripción de los elementos de la estructura 'data_neural_plant'

Nombre	Tipo de dato	Descripción	Parámetros
Ts	Single	Tiempo de muestro de los datos	0.5
U	Array	Vector de los valores de entrada	Vector x5000
Y	Array	Vector de los valores de salida	Vector x5000
Flag	Array	Vector de bandera	Vector x5000 = 1

Una vez completado la estructura, obtendremos una forma como la que se indica en la Figura 77.



The screenshot shows the MATLAB Variables window for 'data_neural_plant'. The window has tabs for PLOTS, VARIABLE, and VIEW. The VARIABLE tab is active, showing a table with the following data:

Field	Value
Ts	0.5000
U	5000x1 double
Y	5000x1 double
flag	5000x1 double

Figura 76. Estructura 'data_neural_plant'

5.9.2. Entrenamiento del controlador

Para el desarrollo del controlador accedemos a una hoja de trabajo de Simulink, desde la cual ingresaremos un bloque de tipo 'NARMA-L2 Controller', el cual es parte de la librería de Neural Network Toolbox, como se indica en la Figura 77.

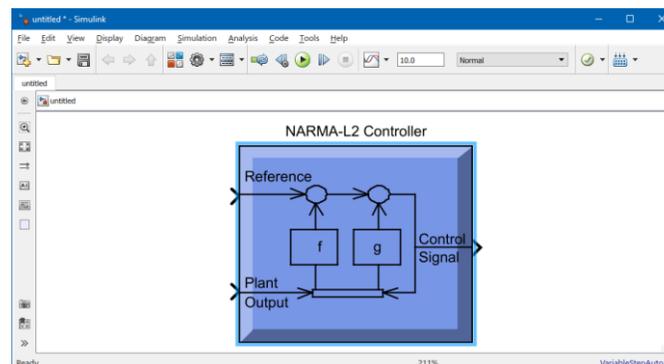
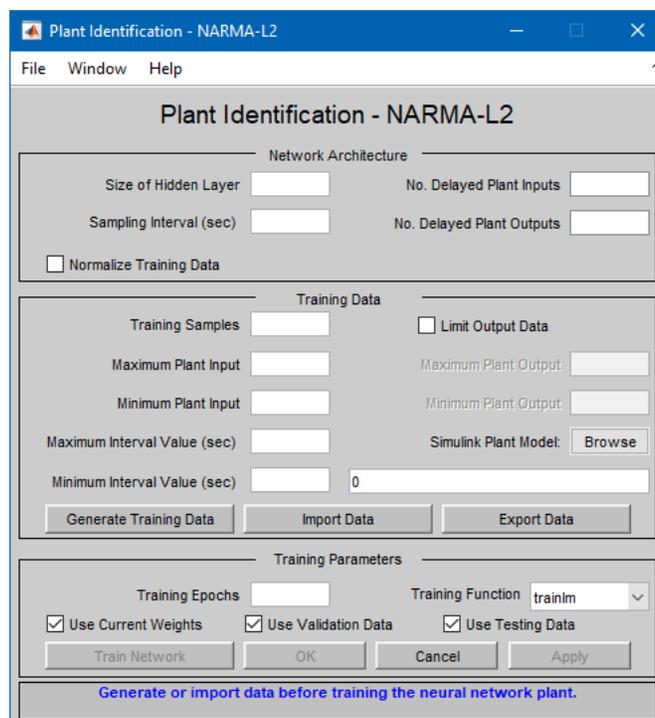


Figura 77. Bloque 'NARMA-L2 Controller'

Este neurocontrolador también es llamado control de linealización de realimentación; su idea central es transformar la dinámica de sistemas no lineales en dinámica lineal cancelando las no linealidades. Es un modelo estándar que se utiliza para representar

los sistemas no lineales generales de tiempo discreto es el modelo de promedio no lineal de movimiento autorregresivo (NARMA - nonlinear autoregressive-moving average). Dentro de su estructura, la red neuronal de dos capas de planta tiene una capa de entrada con una función de transferencia 'tansig'; además hay dos conjuntos de entradas al modelo de planta: valores retardados de la salida de la planta y valores retardados de la salida del controlador. La capa de salida tiene una función de transferencia 'purelin', y se puede establecer el tamaño de la capa oculta. (MathWorks, Inc, 2018)

Dando doble clic sobre el bloque se despliega las propiedades (Figura 78), además de la configuración para el entrenamiento de la red neuronal. El primer paso para ello es determinar el sistema a controlar.



The image shows a software dialog box titled "Plant Identification - NARMA-L2". It has a menu bar with "File", "Window", and "Help". The dialog is organized into three main sections:

- Network Architecture:** Contains input fields for "Size of Hidden Layer", "No. Delayed Plant Inputs", "Sampling Interval (sec)", and "No. Delayed Plant Outputs". There is also a checkbox for "Normalize Training Data".
- Training Data:** Contains input fields for "Training Samples", "Maximum Plant Input", "Minimum Plant Input", "Maximum Interval Value (sec)", and "Minimum Interval Value (sec)". It also has a checkbox for "Limit Output Data", input fields for "Maximum Plant Output" and "Minimum Plant Output", and a "Browse" button for "Simulink Plant Model". At the bottom of this section are buttons for "Generate Training Data", "Import Data", and "Export Data".
- Training Parameters:** Contains a "Training Epochs" field, a "Training Function" dropdown menu (set to "trainlm"), and three checked checkboxes: "Use Current Weights", "Use Validation Data", and "Use Testing Data". At the bottom are buttons for "Train Network", "OK", "Cancel", and "Apply".

At the very bottom of the dialog, there is a blue text link: "Generate or import data before training the neural network plant."

Figura 78. Propiedades del bloque 'NARMA-L2 Controller'

Para entender el funcionamiento de la ventana de entrenamiento, es necesario detallar los parámetros necesarios para el procedimiento:

- **Size of Hidden Layer:** define el tamaño de la segunda capa del modelo de planta red neuronal.
- **Sampling Interval:** intervalo de muestreo en el que se recogerán los datos del modelo de planta de Simulink.
- **No. Delayed Plant Inputs:** define cuántos retardos en la entrada de la planta se utilizarán para alimentar el modelo de red neuronal de la planta.
- **No. Delayed Plant Outputs:** define cuántos retardos en la salida de la planta se utilizarán para alimentar el modelo de red neuronal de la planta.
- **Training Samples:** define cuántos puntos de datos se generarán para el entrenamiento.
- **Maximum Plant Input:** define un límite superior en la entrada aleatoria de la planta.
- **Minimum Plant Input:** define un límite inferior en la entrada aleatoria de la planta.
- **Maximum Interval Value:** define un intervalo máximo durante el cual la entrada aleatoria de la planta permanecerá constante.
- **Minimum Interval Value:** define un intervalo mínimo durante el cual la entrada aleatoria de la planta permanecerá constante.
- **Training Epochs:** define el número de iteraciones para entrenar el modelo de planta de la red neural.

- **Training Function:** selecciona una función de entrenamiento para el modelo de planta de red neuronal.

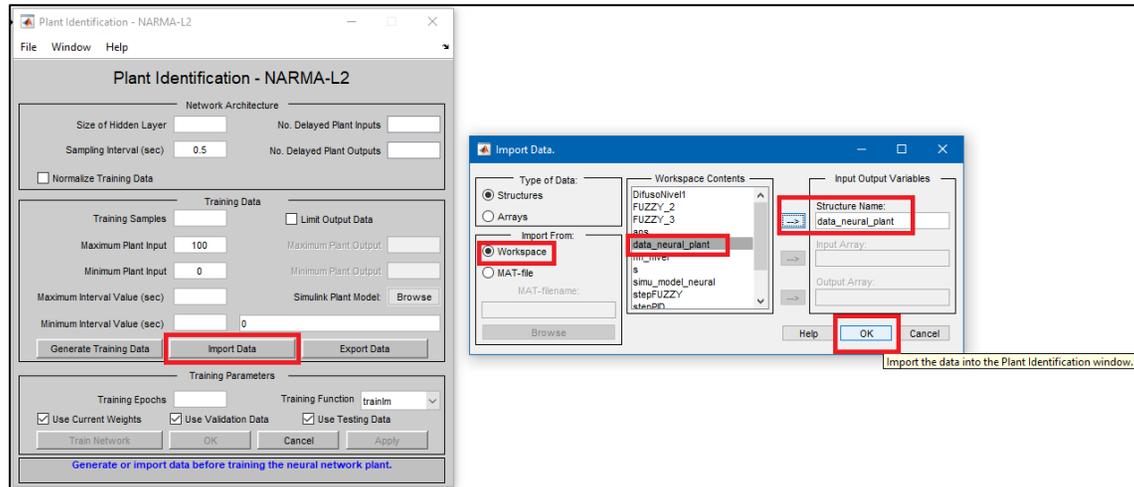


Figura 79. Importación de los datos de entrenamiento de la red neuronal de planta

Para el entrenamiento de la red neuronal existen dos posibilidades: se puede generar un modelo de planta en un archivo de Simulink en el cual esté presente un elemento de Input y otro de Output, a partir del cual se realizará la generación de datos de entrenamiento de manera automática según los parámetros establecidos en el área de 'Datos de Entrenamiento'; la segunda opción es cargar los datos recolectados del funcionamiento de la planta física. Aquí aplicaremos la segunda opción, cargando como datos de entrenamiento la estructura 'data_neural_plant', como se observa en la Figura 79. Los parámetros necesarios para este procedimiento es el tiempo de muestreo (Sampling Interval), y los máximos y mínimos valores de entradas de la planta (Maximum/minimum Plant Input).

Si todos los parámetros son ingresados correctamente, la aplicación los asimilará y presentará en pantalla una gráfica de los datos de entrada/salida del modelo de planta,

como se muestra en la Figura 80, los cuales son los mismo que los presentados en la Figura 41 y Figura 42, ya que proceden de las mismas mediciones. Lo único que se debe hacer es aceptar los datos mostrados.

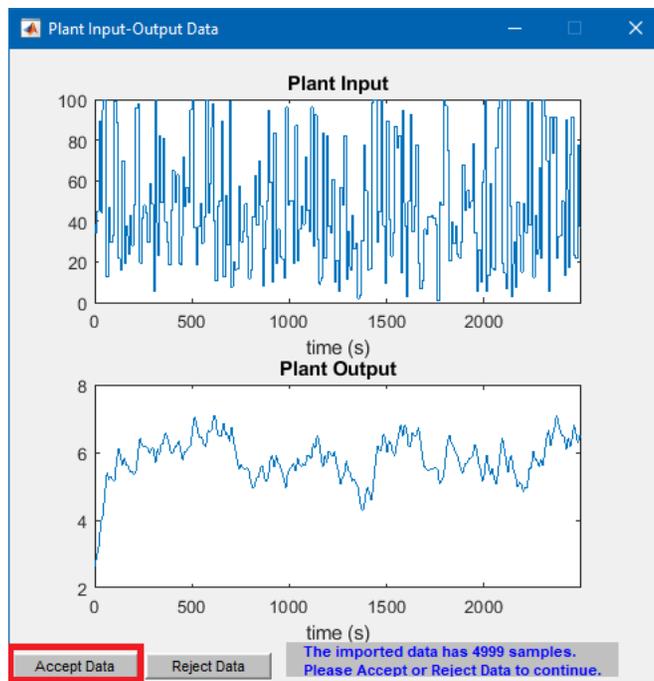


Figura 80. Importación de los datos de entrenamiento de la red neuronal de planta

Una vez que se tiene los datos de entrenamiento listos, se debe completar los parámetros de la estructura de la red, los cuales corresponden al número de neuronas de la capa oculta, así como el número de retardos de entrada y salida que se utilizaran. Según los estudios realizados en el tema de Redes Neuronales Artificiales, no existe un método claramente definido que indique con exactitud el número de neuronas que debería tener una red según la aplicación que se le vaya a dar, por lo cual esta decisión depende únicamente del desarrollador de la red, teniendo en cuenta que mientras mayor sea el número de neuronas y el número de retardos, mayor será el tiempo de entrenamiento, pero a su vez conlleva un mejor procesamiento de los datos; pero a su

vez si se excede el número de neuronas de la capa oculta, se pasa a no tener un entrenamiento óptimo. En esta aplicación, se ha decidido, luego de pruebas iniciales, los parámetros establecidos en la Figura 81.

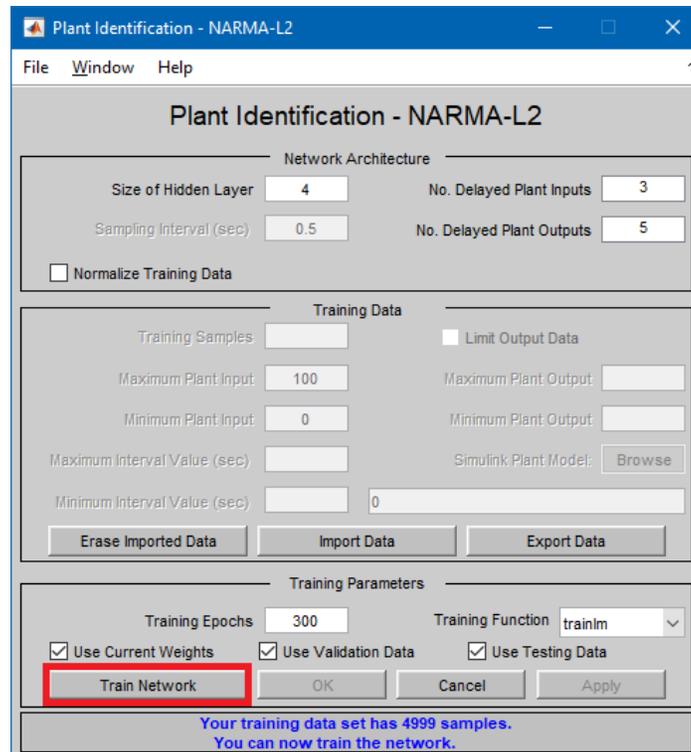


Figura 81. Parámetros de la arquitectura del controlador neuronal

Una vez configurado los parámetros, se procede al entrenamiento de la red. La estructura de datos cargada se utiliza en tres partes del proceso: entrenamiento, validación y pruebas; por defecto, el entrenamiento utiliza un 50% de los valores ingresados para su desarrollo, mientras que la validación y pruebas utiliza el porcentaje restante de los datos, divididos en partes iguales; aunque se puede variar el entrenamiento obviando la validación o la prueba. Una vez concluido el entrenamiento, Matlab muestra una ventana de resultados que expresa la viabilidad y el óptimo entrenamiento de la red neuronal. Aquí se muestran datos como las épocas de

entrenamiento, que son los periodos de tiempo del entrenamiento, tiempo utilizado para el entrenamiento, mejoramiento de éste, entre otros.

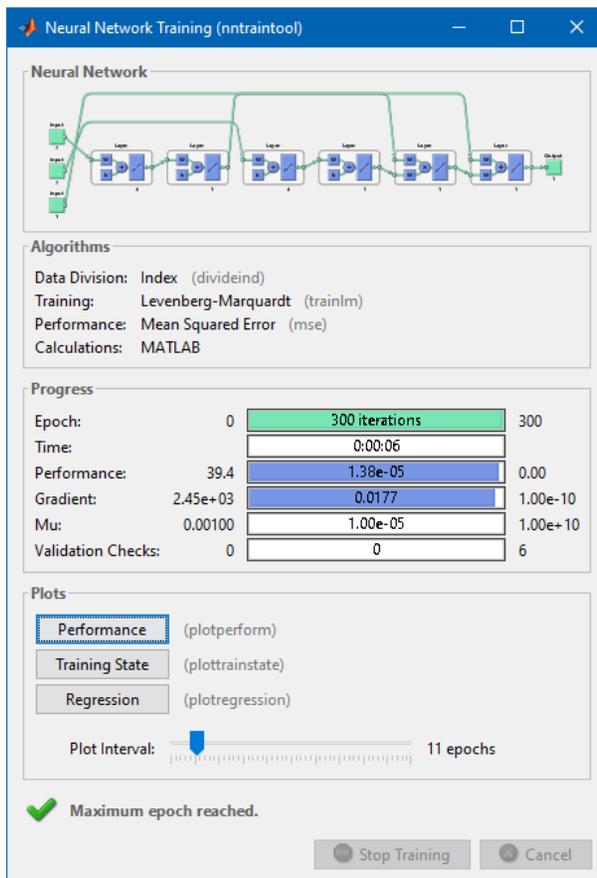


Figura 82. Resultados de entrenamiento de la red neuronal

La ventana nos da la posibilidad de observar un listado de gráficas que permiten analizar el comportamiento del entrenamiento de la red, midiéndolo mediante el rendimiento, la regresión de datos de salida y el estado de entrenamiento. La primera gráfica que se nos presenta es la del rendimiento de la red durante el entrenamiento (Figura 83), el cual se mide en términos del error; para este caso se visualiza que el error medio cuadrático se encuentra en el orden de 10^{-4} , lo cual es muy aceptable para este caso.

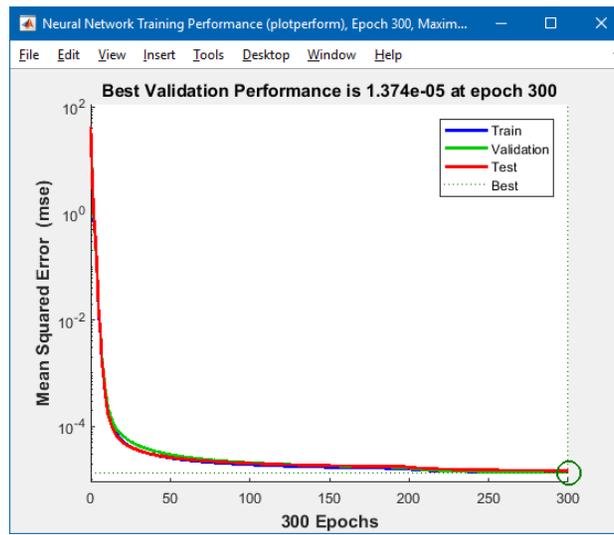


Figura 83. Tendencia de validación del entrenamiento

Una medida del ajuste que ha tenido el entrenamiento es el gráfico de regresión (Figura 84), en el que se muestran los resultados de la red con respecto a los objetivos de entrenamiento, validación y series de pruebas. Para un ajuste perfecto, los datos deben caer dentro de una línea de 45 grados, donde las salidas de la red son iguales a los objetivos.

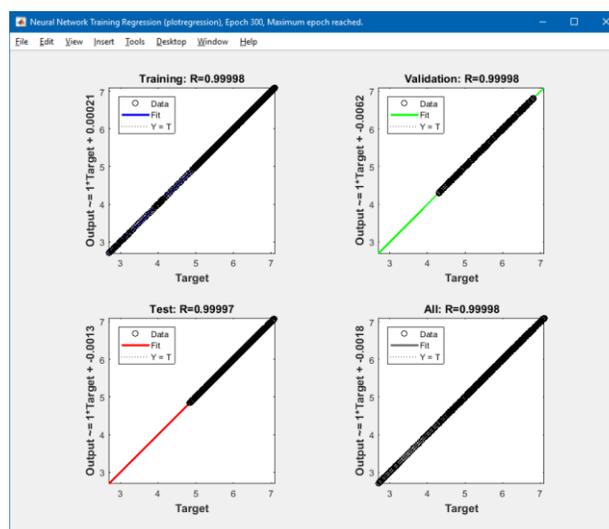


Figura 84. Diagrama de regresión del entrenamiento

Además de estas gráficas, también se presenta al usuario los resultados del controlador en sí, donde se observa, para cada una de las tres etapas del entrenamiento, los datos de entrada/salida con los que trabajo la etapa, así como la salida de la red neuronal de planta y el error que existe entre ésta y la salida real correspondiente a los datos de entrenamiento.

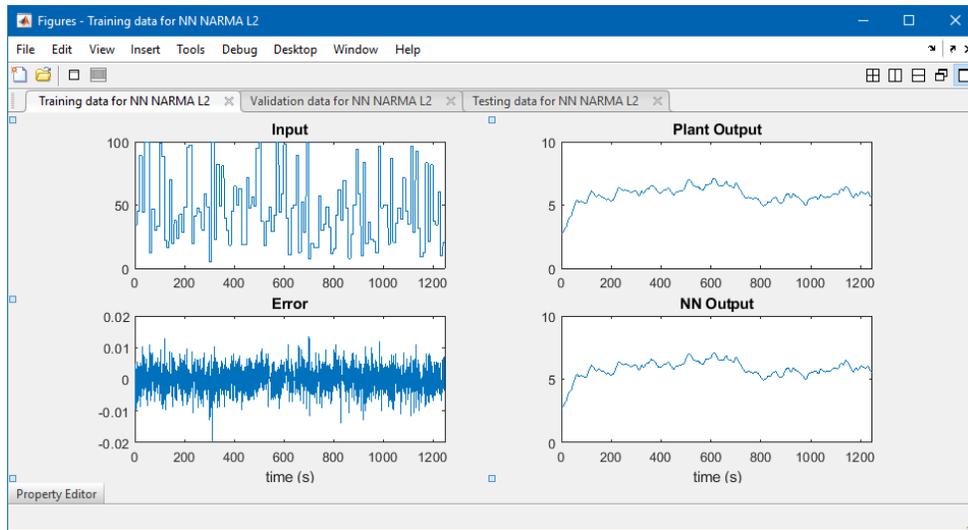


Figura 85. Resultado del entrenamiento de la red neuronal de planta

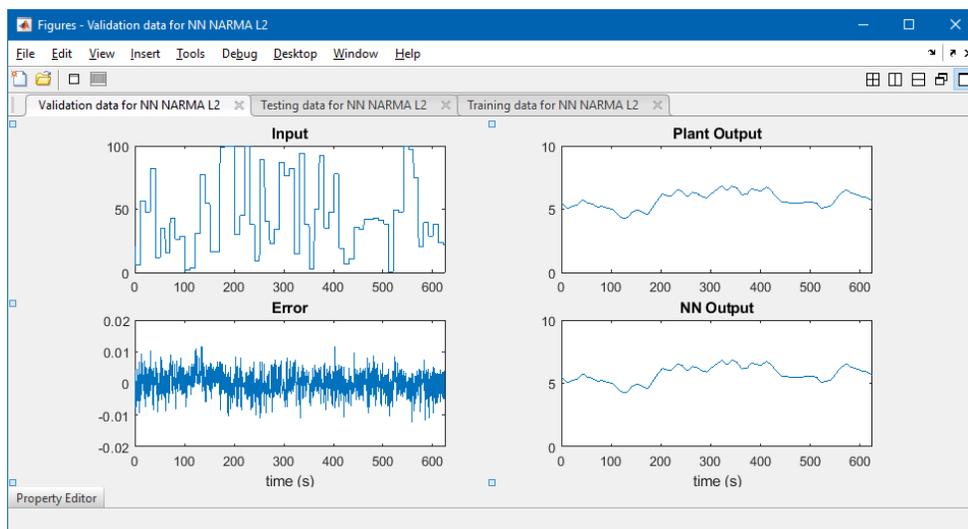


Figura 86. Resultado de validación de la red neuronal de planta

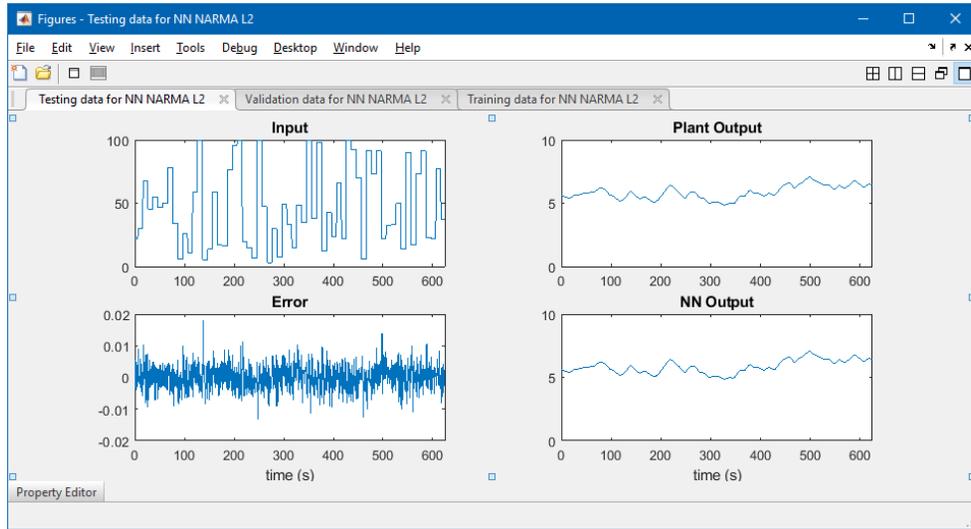


Figura 87. Resultado de pruebas de la red neuronal de planta

Como se puede observar, la red de planta se ajusta casi perfectamente a la respuesta de la planta real, obteniendo una variación de alrededor de 0.02, por lo que se puede tomar como válido el modelo neuronal. En la Figura 88 se puede apreciar un gráfico esquemático de la arquitectura de la red desarrollada, donde se puede visualizar las capas de entrada/salida, así como los retardos establecidos.

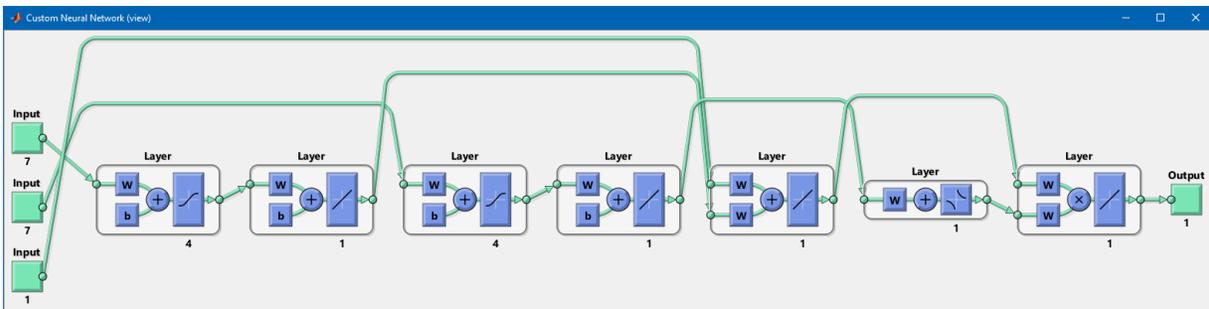


Figura 88. Diagrama de la arquitectura de la red.

5.9.3. Puesta a prueba del controlador neuronal

Para verificar el funcionamiento del controlador neuronal desarrollado, se lo implementará en el mismo entorno de Simulink para visualizar su respuesta frente al modelo de planta simulado. Para ello, el propio bloque del controlador nos indica los parámetros de entrada y salida que posee, por lo cual realizamos un diagrama como el mostrado en la Figura 89; no es necesario realizar ninguna otra configuración adicional.

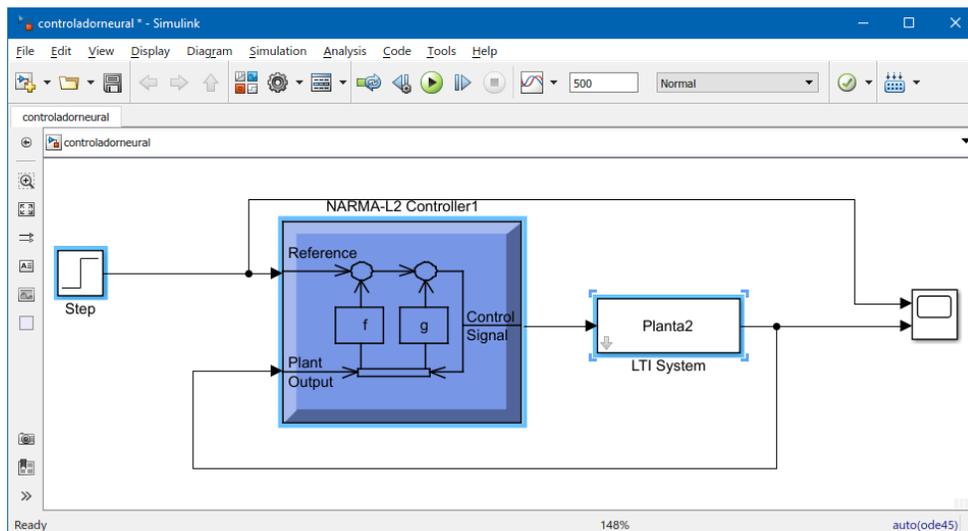


Figura 89. Diagrama del controlador neuronal.

Una vez completado el diagrama del bloque, se procede a simular el comportamiento del controlador, primero con una señal de tipo escalón (Figura 90), y luego con una serie de consignas aleatorias (Figura 91). En estas gráficas, la línea verde (Step/Uniform Random Number) representa la consigna o setpoint deseado, y la línea roja (LTI System) representa la salida del modelo de planta generado.

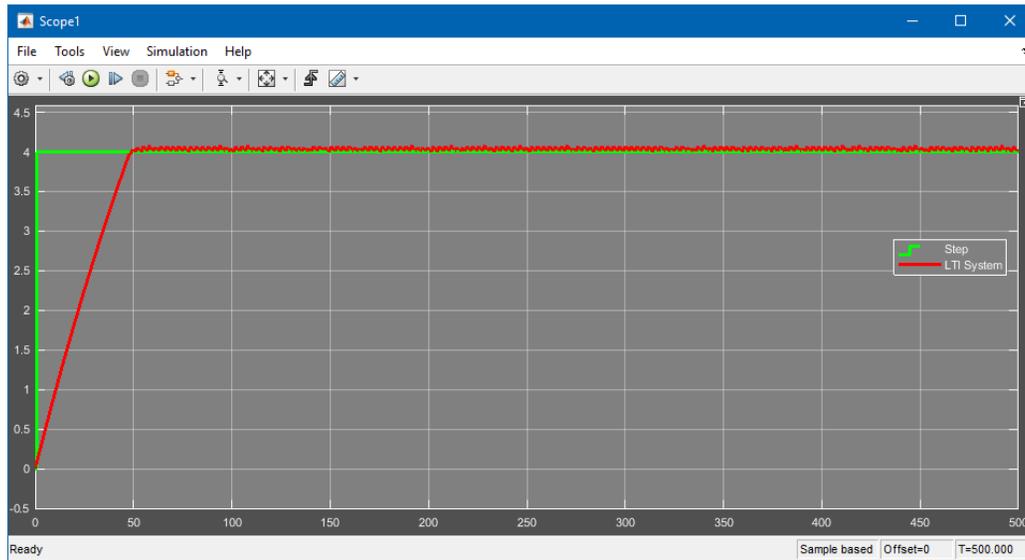


Figura 90. Respuesta del controlador neuronal al escalón

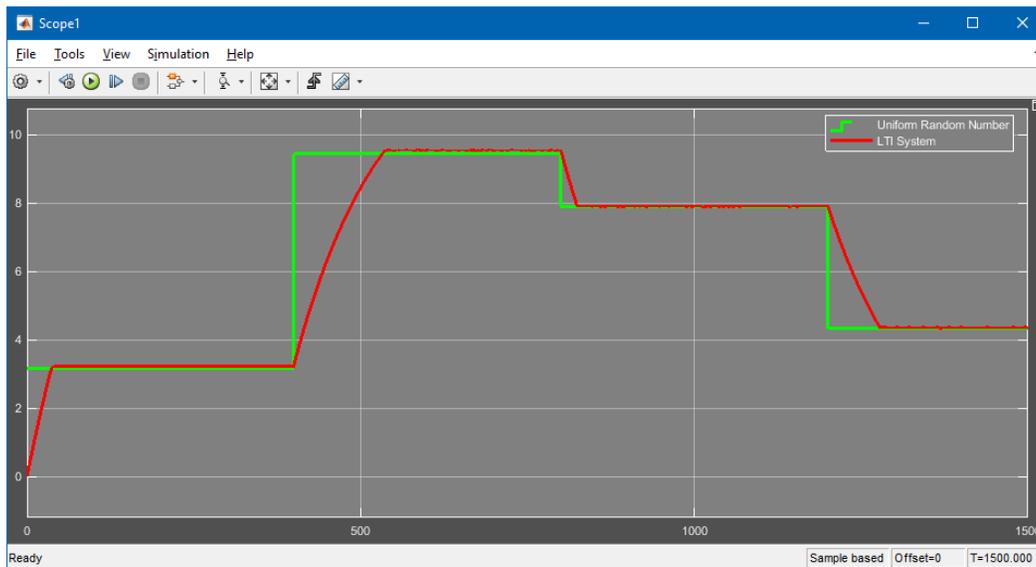


Figura 91. Respuesta del controlador neuronal a valores de consigna aleatorios

Como se puede observar en ambas respuestas, el nivel de la planta crece rápidamente hasta llegar al valor de consigna, donde reduce su velocidad drásticamente para ajustarse relativamente rápido al valor de setpoint. En la respuesta al escalón, el controlador provoca una señal de salida para que la planta se estabilice, lográndolo en un tiempo

relativamente corto. En cuanto a la respuesta de la consigna variable, el controlador logra que la planta llegue al valor establecido en cada uno de los valores aleatorios.

5.10. Diseño de la red industrial

Al hablar de sistemas distribuidos, hablamos de procesos que funcionan con un algoritmo de control de forma autónoma, lo que significa que cada proceso cuenta con su unidad de control, gestionada por un autómatas interconectado a ella. El modelo de planta industrial utilizado en este proyecto cuenta con su PLC para su funcionamiento, por lo cual es necesario añadir controlador que gestione el proceso de control de éste. Por lo tanto, se implementa una red Industrial que conste de: un controlador maestro, el cual enviará las señales de control y procesamiento hacia los equipos de campo; controladores esclavos que tendrán contacto directo con los sensores y actuadores de la planta; y un dispositivo de visualización e interacción, como un HMI, para el control del proceso por parte del usuario. Para esto, se montará una red industrial PROFIBUS y PROFINET entre la planta y los equipos desarrollados por (Amores & Quingatuña, 2017), que se basa en el uso de redes industriales para el control distribuido de sistemas industriales.

5.10.1. Desarrollo del proyecto mediante TIA PORTAL

Dentro del Software TIA PORTAL, se crea un nuevo proyecto, en el cual se añadirán los dispositivos antes mencionados para montar la red industrial presentada en la Figura 92, mientras que en la Tabla 5, se detalla las características de los equipos utilizados.

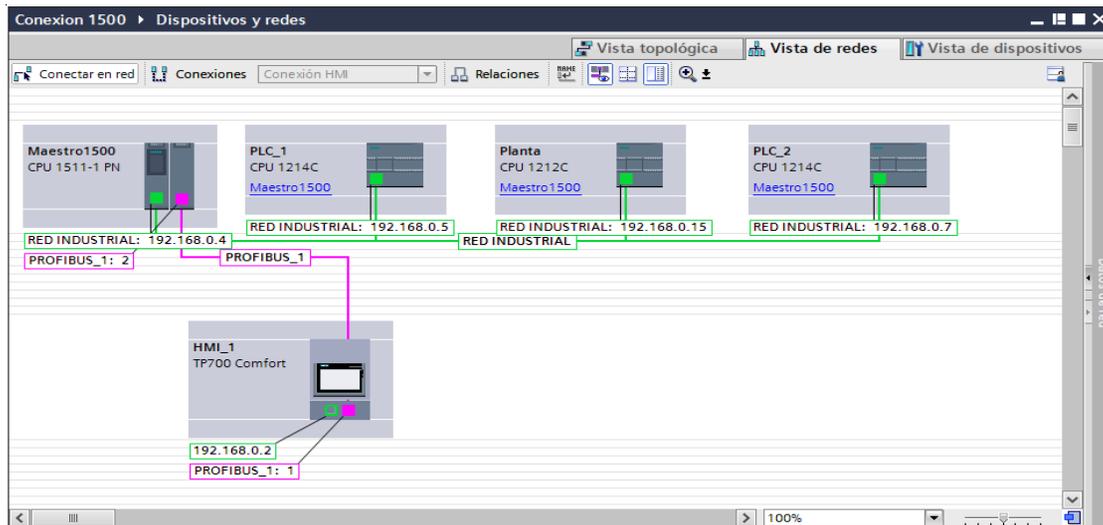


Figura 92. Red industrial del proyecto

Tabla 5

Descripción de los equipos que conforman la red industrial

Nombre	Tipo	Modelo
Maestro1500	Maestro	S7-1500 CPU 1511-1 PN
Planta	Esclavo	S7-1200 CPU 1212C AC/DC/Rly
PLC_1	Esclavo	S7-1200 CPU 1214C DC/DC/DC
PLC_2	Esclavo	S7-1200 CPU 1214C DC/DC/DC
HMI_1	Visualización	TP700 Comfort

El controlador maestro se comunica con el panel HMI a través de un enlace PROFIBUS mientras que los esclavos lo hacen a través de una red PROFINET. La ventaja de usar dicho protocolo dentro de los controladores SIEMENS es la posibilidad de aplicar la funcionalidad I-Device, que define al dispositivo maestro como 'Controlador IO' (input/output), y a los esclavos como 'Dispositivo IO'; esto facilita un intercambio de datos de forma determinista con un controlador IO, lo cual permite emplear la CPU como una unidad inteligente de preprocesamiento de procesos parciales, donde el I-Device está

integrado como dispositivo IO a un controlador IO de nivel superior. En la Figura 93 se esquematiza el proceso de la funcionalidad I-Device. Esta funcionalidad se la utiliza para repartir una tarea de automatización entre unidades parciales menores.

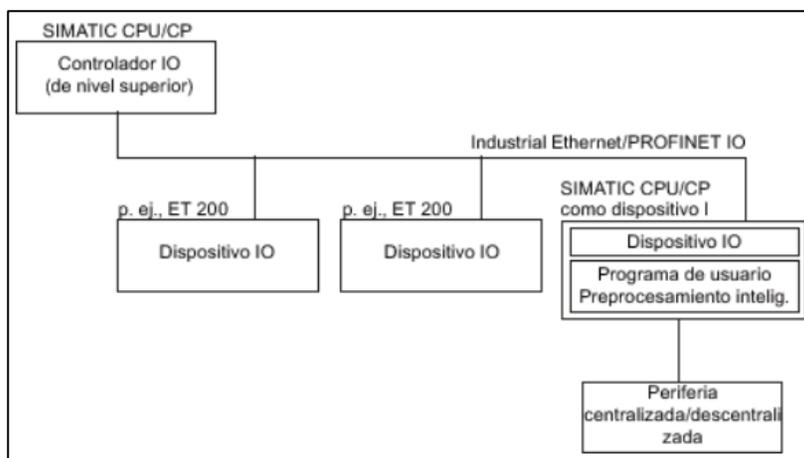


Figura 93. Esquema generalizado de la funcionalidad I-Device.

Fuente: (Siemens AG, 2016)

La configuración de I-Device se la realiza en la parte PROFINET de los que serán dispositivos IO, como se indica en la Figura 94. Una vez señalado, se asigna el controlador al cual estará ligado el dispositivo. Lo siguiente es delimitar las áreas de transferencia de los dispositivos. Consiste en establecer un rango de variables de tipo I del controlador IO, que se convertirán en variables de tipo O en el dispositivo IO, y viceversa. Por lo cual se generan dos áreas de transferencia por cada dispositivo: una que vaya de variables I→O y la segunda de O→I, como se indica en la Figura 95.

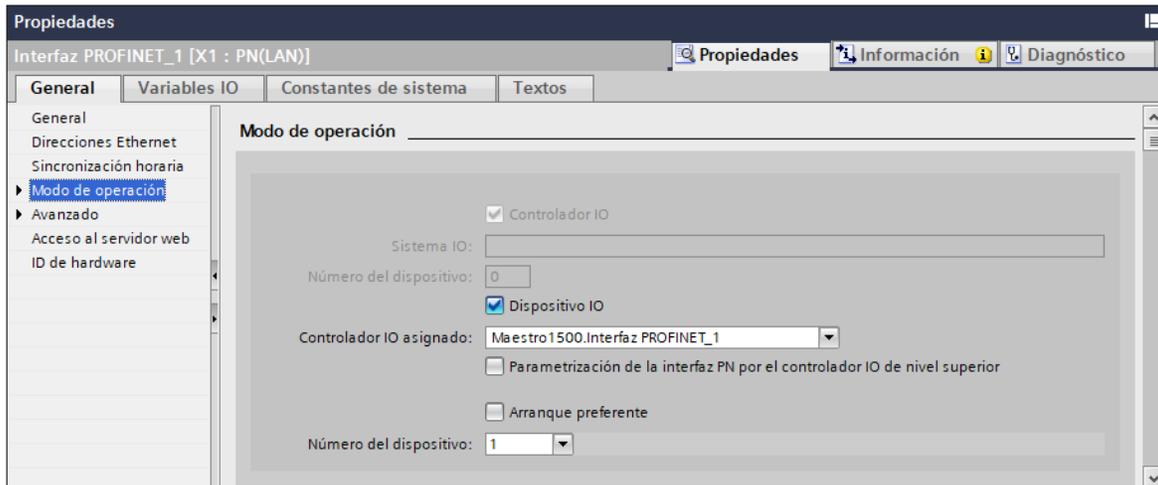


Figura 94. Configuración I-Device.

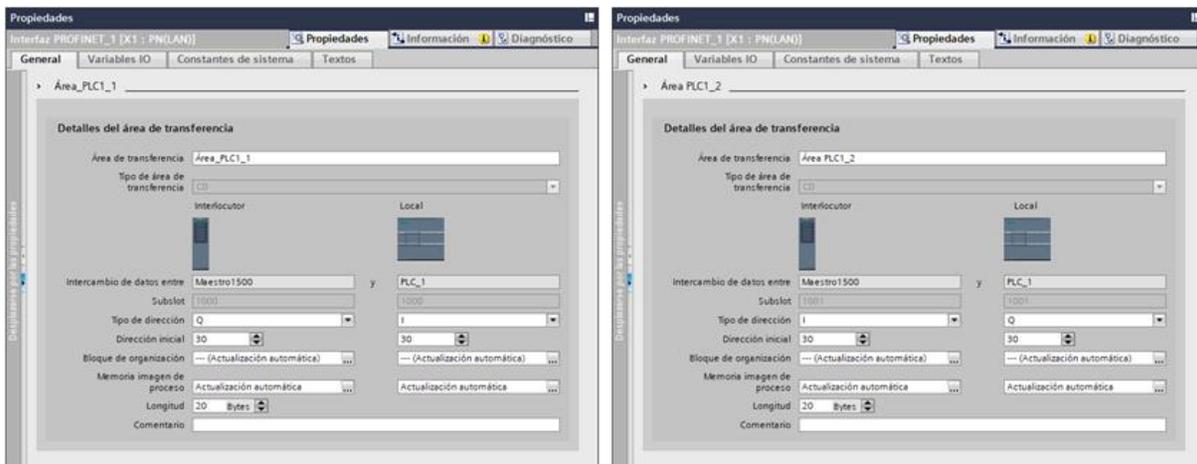


Figura 95. Configuración de las áreas de transferencia.

La programación del controlador maestro se divide en bloques de acuerdo con el tipo de controlador: para cada uno de los controles PID, difuso y neuronal existe un manejo de la bomba, así como del estado del controlador; y un bloque que se encarga de la recolección de datos de prueba mediante un Datalogging. En la Figura 96 se muestra el listado de los bloques de programación.

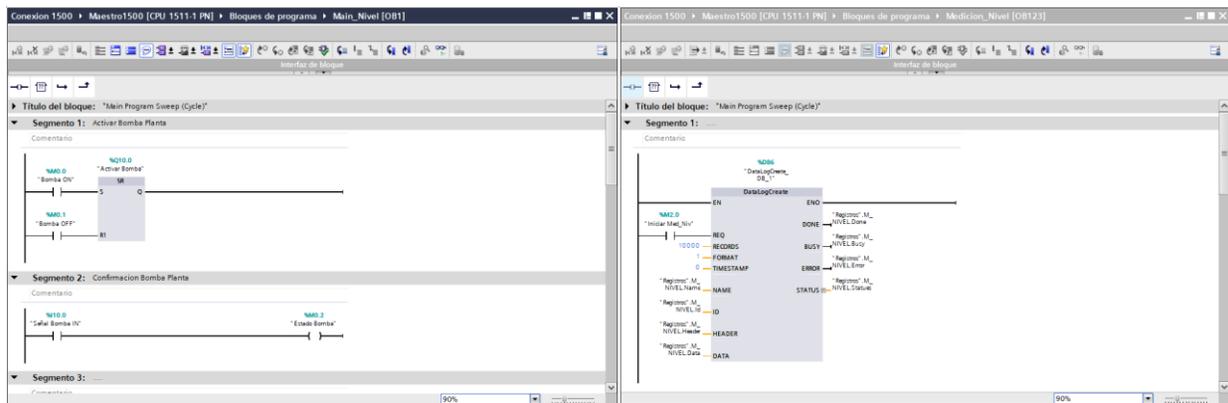


Figura 96. Bloques de programación para el Maestro1500

5.11. Implementación de los controladores a la planta física

Es necesario comprobar el comportamiento de los controladores desarrollados en los puntos anteriores aplicados en la planta física, para lo cual se los debe llevar del entorno de Simulink hacia el software de programación del controlador PLC. Para este proceso, se utilizó el producto Simulink PLC Coder.

5.11.1. Simulink® PLC Coder™

Simulink PLC Coder es un producto de Matlab que genera textos estructurados o SCL y diagramas de tipo Ladder independientes del hardware bajo norma IEC 61131-3 a partir de modelos de Simulink y funciones de MATLAB. El texto estructurado y los diagramas Ladder se generan en PLCopen XML y otros formatos de archivo compatibles con entornos de desarrollo integrados (IDEs) como Rockwell Automation Studio 5000, Siemens TIA Portal, entre otros. Con este producto se puede compilar y desarrollar la aplicación en numerosos dispositivos de control lógico programable (PLC) y controladores de automatización programables (PAC). (MathWorks, Inc, 2018). Entre sus principales características se encuentran:

- Generación automática de textos estructurados y diagramas de escalera IEC 61131-3.
- Soporte de MATLAB, incluyendo sentencias if-else, construcciones de bucle y operaciones matemáticas.
- Soporte de IDEs, incluyendo 3S-Smart Software Solutions CODESYS, Rockwell Automation Studio 5000, Siemens TIA Portal, OMRON Sysmac Studio y PLCopen XML.
- Compatibilidad con Simulink, incluidos los subsistemas reutilizables, los bloques de controladores PID y las tablas de búsqueda.
- Soporte para múltiples tipos de datos, incluyendo booleano, entero, enumerado y coma flotante, así como vectores, matrices, buses y parámetros sintonizables.
- Creación de bancos de pruebas. (MathWorks, Inc, 2018)

5.11.2. Migración de los controladores hacia el PLC

Con el diagrama de bloques de los controladores listo, se procede a crear un subsistema con los elementos que conforman la parte de control, como se muestra en la Figura 97. Con ello se obtiene un nuevo bloque el cual contiene todo el proceso seleccionado; si accedemos al éste, podemos cambiar el nombre de las variables de entrada y salida para que se relacionen mejor con sus contrapartes físicas, como se indica en la Figura 98. Para este caso específico, se realizará la migración del controlador difuso.

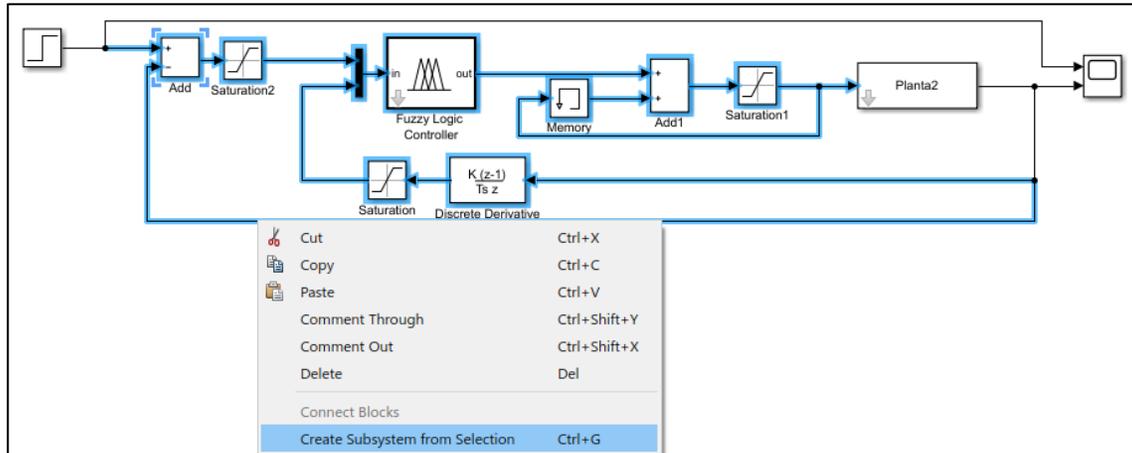


Figura 97. Creación del bloque de subsistema

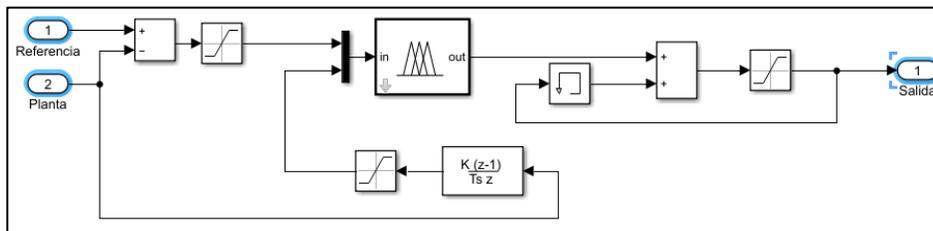


Figura 98. Cambio de nombre de las variables del controlador

Lo siguiente es modificar el parámetro 'Treat as atomic unit' del subsistema creado, como se muestra en la Figura 99. Este es un paso necesario para la utilización de PLC Coder, ya que hace que Simulink trate el subsistema como una unidad al determinar el orden de ejecución de los métodos de bloques.

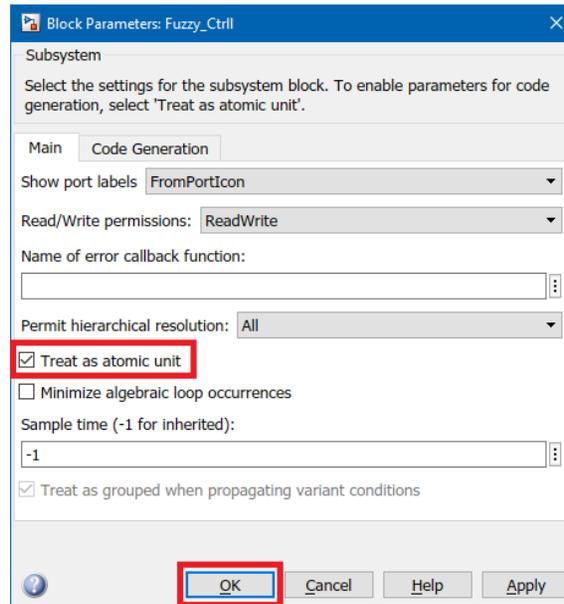


Figura 99. Parámetros del bloque de subsistema

Luego de esto, se realiza la configuración del PLC Coder desde la ventana de Configuración de Parámetros del Modelo, como se indica en la Figura 100, en donde se seleccionará el entorno de desarrollo integrado IDE al cual se exportará el archivo SCL; para este caso, como se trata de un PLC S7-1200, se trabaja con TIA PORTAL.

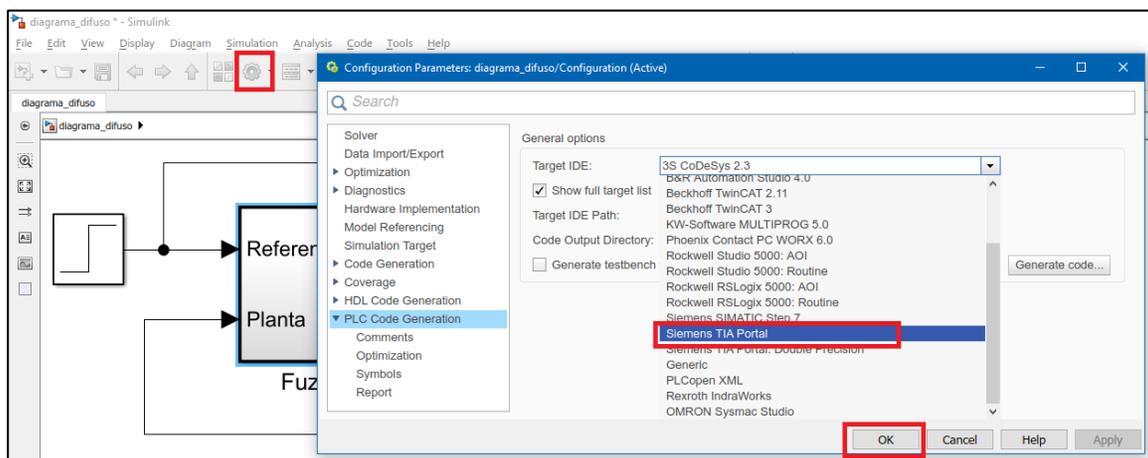


Figura 100. Parámetros del PLC Coder.

Con la configuración ya establecida, se procede a generar la codificación del archivo. Una vez que se haya completado el proceso, aparecerá el Visor de Diagnostico confirmando la generación del archivo, o a su vez, errores encontrados durante el proceso de compilación, como se indica en la Figura 101. Para evitar errores, es importante no usar bloques de funciones continuas (ubicadas en las librerías como “Continuous”), ya que PLC Coder no los soporta.

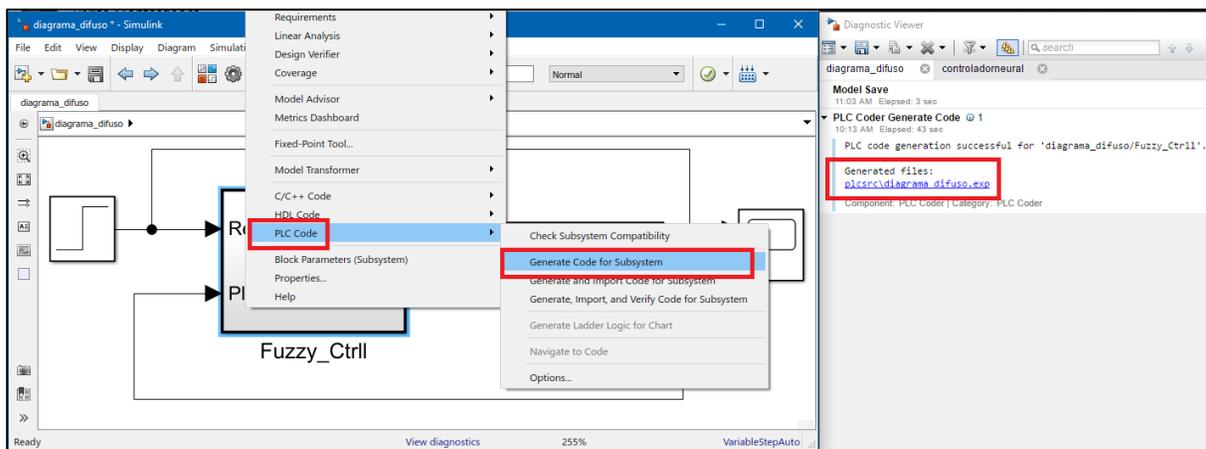
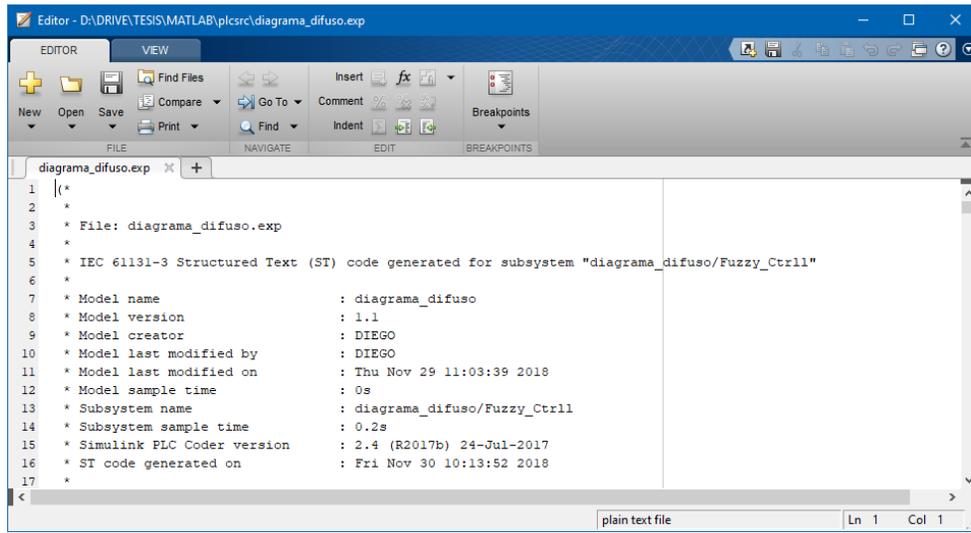


Figura 101. Generación del código estructurado.

Al abrir el archivo generado, se muestra un formato de texto plano según estándares IEC 61131-3, como se indica en la Figura 102, donde lo único que debemos hacer es guardar el archivo con una extensión SCL.



```

1 |(*
2 |*
3 |* File: diagrama_difuso.exp
4 |*
5 |* IEC 61131-3 Structured Text (ST) code generated for subsystem "diagrama_difuso/Fuzzy_Ctrl1"
6 |*
7 |* Model name           : diagrama_difuso
8 |* Model version        : 1.1
9 |* Model creator         : DIEGO
10 |* Model last modified by : DIEGO
11 |* Model last modified on  : Thu Nov 29 11:03:39 2018
12 |* Model sample time     : 0s
13 |* Subsystem name        : diagrama_difuso/Fuzzy_Ctrl1
14 |* Subsystem sample time : 0.2s
15 |* Simulink PLC Coder version : 2.4 (R2017b) 24-Jul-2017
16 |* ST code generated on   : Fri Nov 30 10:13:52 2018
17 |*

```

Figura 102. Código estructurado del controlador.

Con el archivo SCL, lo siguiente es importarlo desde el proyecto de TIA PORTAL, específicamente desde el área de Fuentes Externas. Aquí se selecciona la ruta del archivo SCL. Cuando el archivo se encuentre en el proyecto, se tiene que generar los bloques de programación contenidos en el archivo, como se ve en la Figura 103.

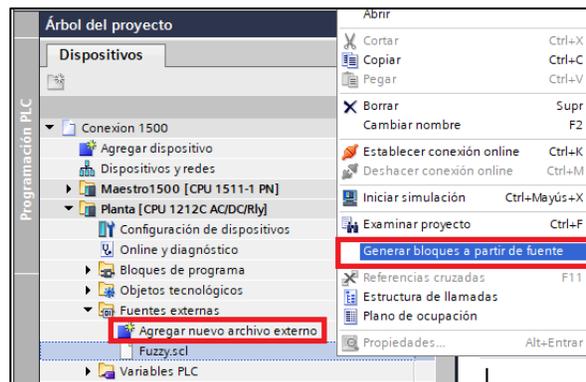


Figura 103. Generación automática de los bloques.

Con esta acción, el software comienza a leer el archivo, y compila los bloques de programa que se encuentran codificados; los bloques resultantes son bloques de función (FB) y funciones (FC), ya que el subsistema generado en Simulink requiere de sus

propias funciones para operar, las cuales también se encuentran en el archivo SCL. En la Figura 104 se puede observar el resultado de la generación.

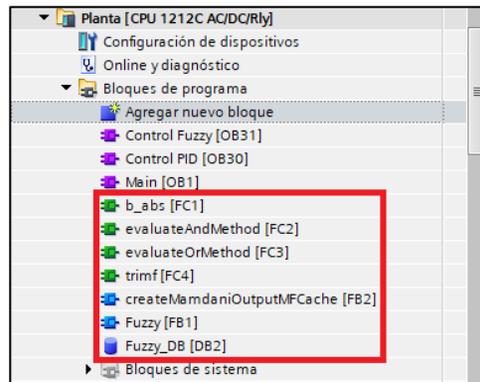


Figura 104. Bloques de programa generados.

Finalmente, solo queda añadir dentro de los bloques de programa ordinarios, el bloque de función correspondiente, como se indica en la Figura 105.

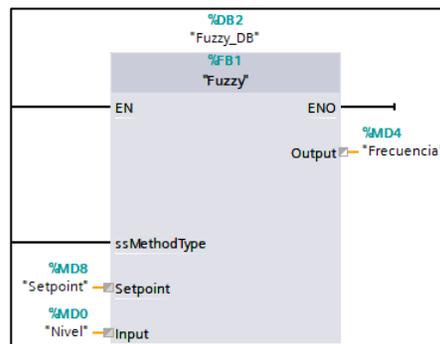


Figura 105. Llamada de la función generada.

De la misma manera, es necesario llevar cada uno de los controladores desarrollados en el ambiente de Simulink hacia el PLC siguiendo el mismo procedimiento, con lo cual al final se tendrán los tres tipos de control implementados en el mismo procesador.

5.12. Resultados de la implementación

Para la evaluación del comportamiento de los controladores desarrollados, se realiza un análisis de su comportamiento frente a una señal de entrada de tipo escalón, con lo cual se obtendrá los valores de respuesta en función del tiempo. El objetivo de los controladores es lograr un control “robusto” que tenga la característica de mantener la variable de proceso muy estable a pesar de los cambios de carga, producir una rápida respuesta ante los cambios de setpoint, una mínima oscilación sin importar el tiempo de cambio y un mínimo error entre el valor de setpoint y variable de proceso en el tiempo. El análisis consta de dos partes: la evaluación de su respuesta simulada y la obtenida de la implementación en la planta física. La planta física, en su variable de nivel, cuenta con una escala de 0 a 10 litros, pero debido al propio funcionamiento en vacío de la planta, el nivel se establece en alrededor de dos, por lo que el rango de pruebas para el sistema será de 3 a 9 litros en intervalos de 1 litro. Para la parte de simulación, los datos serán recolectados por Simulink y enviados hacia el área de Matlab para su análisis; para la parte de datos físicos, se utilizará la misma programación de Datalog para almacenar los datos de funcionamiento en un intervalo de muestreo de 0.1 segundos.

5.12.1. Función STEPINFO

Para obtener los parámetros que caracterizan la respuesta de los controladores, nuevamente se utiliza el software Matlab, ya que nos permite utilizar la función ‘stepinfo’, la cual nos presenta las características de respuesta al escalón de un sistema físico, como se observa en la Figura 106. Específicamente, esta función requiere de una curva

de entrada $y(t)$, que será la obtenida del nivel de la panta a diferentes consignas, y nos devuelve parámetros de respuesta como:

- **Tiempo de subida (RiseTime - RT):** Tiempo que tarda la respuesta en subir del 10% al 90% de la respuesta en estado estacionario.
- **Tiempo de estabilización (SettlingTime - ST):** Tiempo que tarda el error $|y(t) - y_{final}|$ entre la respuesta $y(t)$ y la respuesta de estado estacionario y_{final} en caer dentro del 2% de y_{final} .
- **Asentamiento Mínimo (SettlingMin - Smin):** Valor mínimo de $y(t)$ una vez que la respuesta ha aumentado.
- **Asentamiento Máximo (SettlingMax - Smax):** Valor máximo de $y(t)$ una vez que la respuesta ha aumentado.
- **Sobre impulso (Overshoot - Os):** Porcentaje de sobre impulso, en relación con y_{final} .
- **Sub impulso (Undershoot - Us):** Porcentaje de sub-impulso, en relación con y_{final} .
- **Pico (Peak - Pk):** Valor absoluto pico de $y(t)$.
- **Tiempo de pico (PeakTime - PT):** Tiempo en el que se produce el valor pico.

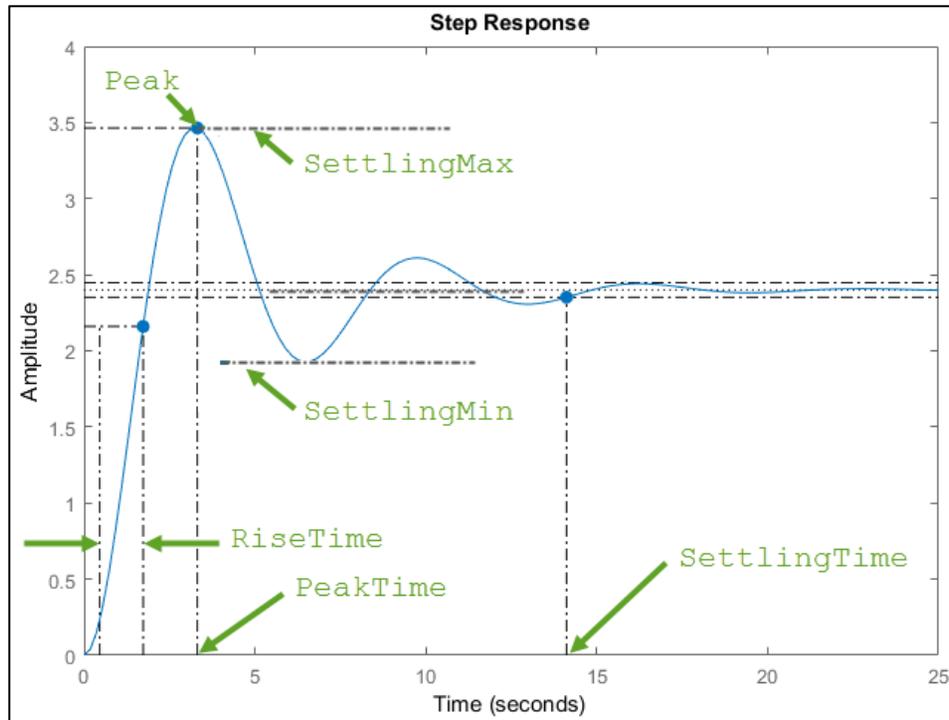


Figura 106. Parámetros de la instrucción 'stepinfo'

5.12.2. Resultados de los controladores simulados

En la Figura 107 se puede observar que el controlador PID tiene un control muy suavizado en el crecimiento de la variable, lo que se traduce en un elevado tiempo de subida y de estabilización; por otro lado, los controladores difuso y neuronal provocan un crecimiento máximo de la variable. En la parte difusa, se observa un cambio en la pendiente de crecimiento alrededor de 2.5 litros, desde el cual el nivel crece más lentamente, mientras que el neuronal continua con la misma pendiente hasta llegar al valor de consigna y se estabiliza con una serie de oscilaciones. En la Tabla 6 se tiene el resultado de la función "stepinfo" para cada uno de los controladores, donde la estabilización más rápida la tuvo el controlador neuronal, pero a su vez también tuvo el mayor porcentaje de sobre impulso, contrario al difuso que tuvo el menor.

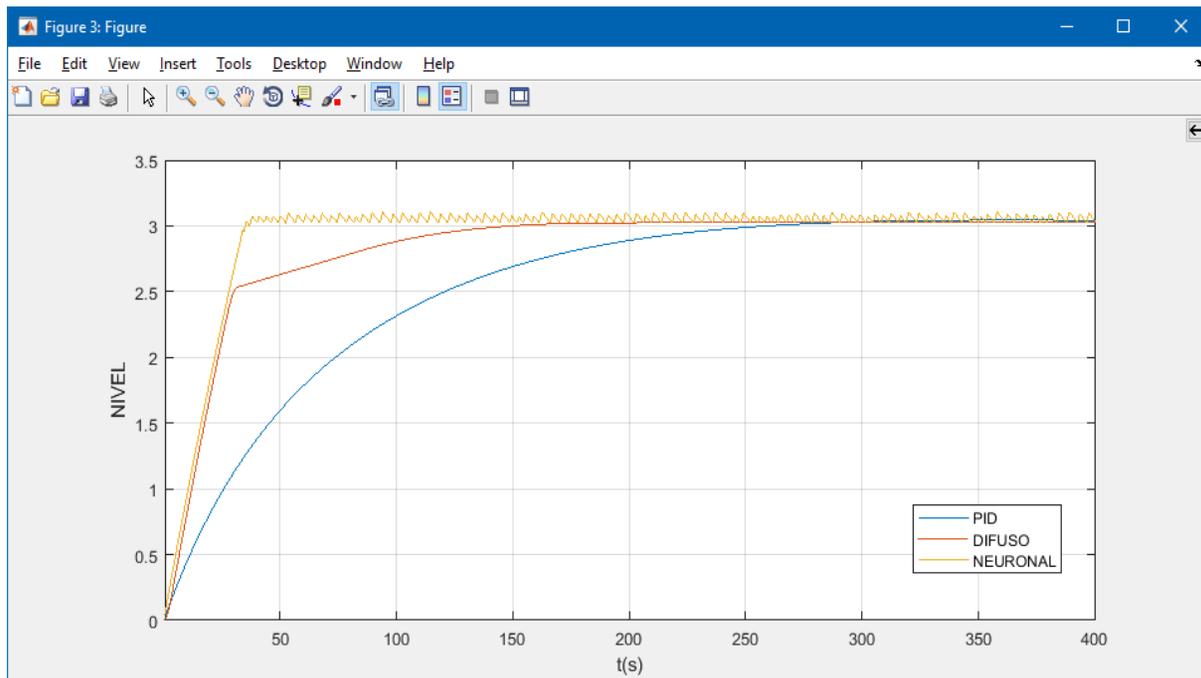


Figura 107. Respuesta de la simulación a un escalón de 3

Tabla 6

Respuesta de los controladores simulados para un Setpoint de 3.

Parámetro	PID	Difuso	Neuronal
RT	145,1984	58,8022	27,4638
ST	186,9686	92,5862	32,4482
Smin	2,7007	2,7005	2,7060
Smax	3,0447	3,0243	3,1062
Os	1,4895	0,8102	3,5413
Us	0,0000	0,0000	0,0000
Pk	3,0447	3,0243	3,1062
PT	363,0000	226,2000	94,0500

En la Figura 108 se puede observar que el controlador PID provoca un crecimiento más veloz de la variable, lo que se traduce en un tiempo de subida y de estabilización reducidos comparados al setpoint de 3; por otro lado, los controladores difuso y neuronal

mantienen su pendiente de crecimiento de la variable. En la parte difusa, se vuelve a observar un cambio en la pendiente de crecimiento alrededor de 0.5 litros antes del valor de consigna, desde el cual el nivel crece a menor velocidad, mientras que el neuronal continua con la misma pendiente hasta llegar al valor de consigna y se estabiliza con una serie de oscilaciones. En la Tabla 7 se tiene el resultado de la función “stepinfo”, donde la estabilización más rápida la tuvo el controlador neuronal, pero a su vez también tuvo el mayor porcentaje de sobre impulso, contrario al difuso que tuvo el menor.

Tabla 7

Respuesta de los controladores simulados para un Setpoint de 4.

Parámetro	PID	Difuso	Neuronal
RT	145,1984	56,2182	38,4221
ST	186,9686	100,1883	45,3404
Smin	3,6009	3,6007	3,6058
Smax	4,0596	4,0243	4,0788
Os	1,4895	0,6077	1,9691
Us	0	0	0
Pk	4,0596	4,0243	4,0788
PT	363	243,9	426

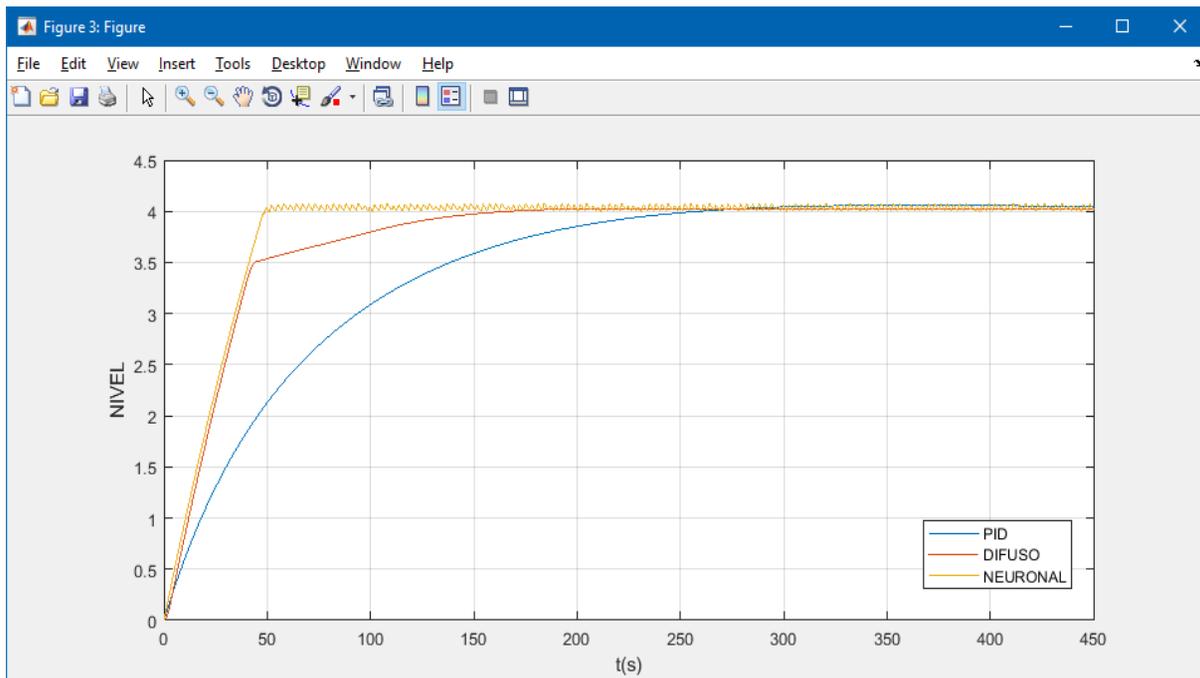


Figura 108. Respuesta de la simulación a un escalón de 4

En la Figura 109 se puede observar que el controlador PID provoca un crecimiento proporcional de la variable comparados al setpoint de 4, con un tiempo de subida y de estabilización similares; por otro lado, los controladores difuso y neuronal mantienen un comportamiento similar a las consignas anteriores. En la parte difusa, se vuelve a observar el cambio en la pendiente de crecimiento alrededor de 0.5 litros antes del valor de consigna, y el controlador neuronal continua con la misma pendiente hasta llegar al valor de consigna y se estabiliza con una serie de oscilaciones. En la Tabla 8 se tiene el resultado de la función “stepinfo”, donde la estabilización más rápida la tuvo el controlador neuronal, el mayor porcentaje de sobre impulso lo tuvo el PID, contrario al difuso que tuvo el menor.

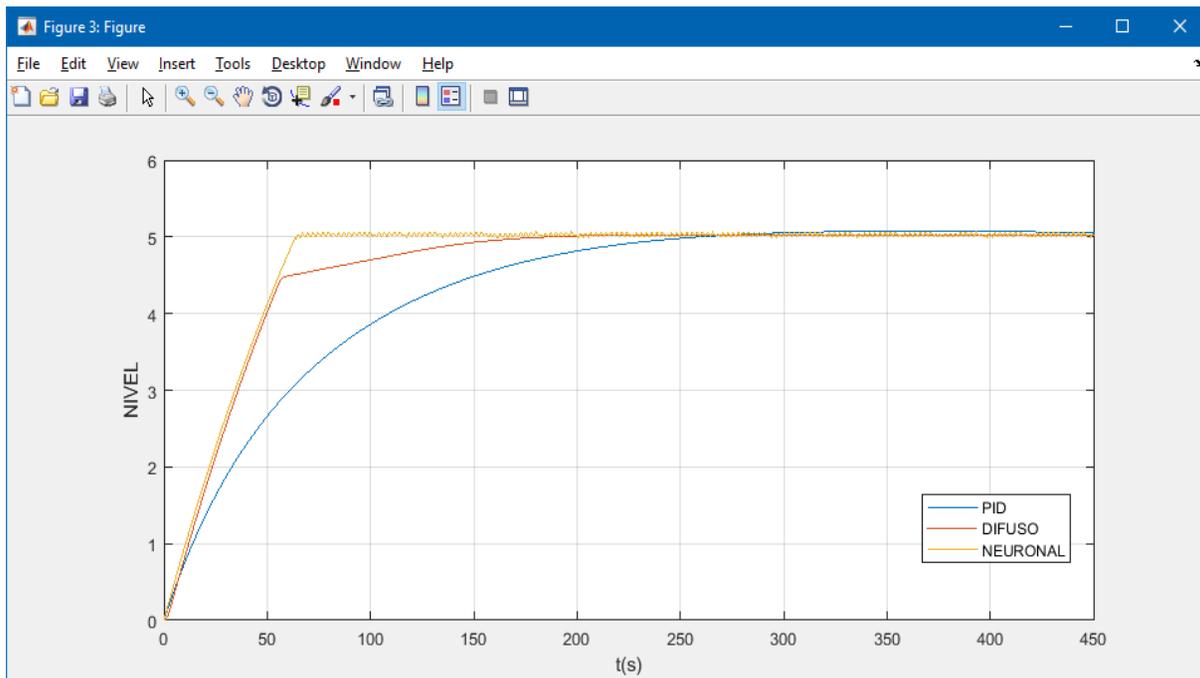


Figura 109. Respuesta de la simulación a un escalón de 5

Tabla 8

Respuesta de los controladores simulados para un Setpoint de 5.

Parámetro	PID	Difuso	Neuronal
RT	145,1984	55,3989	50,5851
ST	186,9686	109,5095	59,6446
Smin	4,5011	4,5004	4,5048
Smax	5,0745	5,0243	5,0689
Os	1,4895	0,4861	1,378
Us	0	0	0
Pk	5,0745	5,0243	5,0689
PT	363	262,8	496,5

En la Figura 110 se puede observar que el controlador PID provoca un crecimiento proporcional de la variable comparados al setpoint de 4, con un tiempo de subida y de estabilización más reducidos; por otro lado, los controladores difuso y neuronal

mantienen un comportamiento similar a las consignas anteriores. En la parte difusa, se vuelve a observar el cambio en la pendiente de crecimiento alrededor de 0.5 litros antes del valor de consigna, y el controlador neuronal continua con la misma pendiente hasta llegar al valor de consigna y se estabiliza con una serie de oscilaciones. En la Tabla 9 se tiene el resultado de la función “stepinfo”, donde la estabilización más rápida la tuvo el controlador neuronal y la más lenta el PID; el mayor porcentaje de sobre impulso lo tuvo el PID, contrario al difuso que tuvo el menor.

Tabla 9

Respuesta de los controladores simulados para un Setpoint de 6.

Parámetro	PID	Difuso	Neuronal
RT	145,1984	64,4453	64,2256
ST	186,9686	120,1586	75,7087
Smin	5,4014	5,4007	5,4047
Smax	6,0894	6,0243	6,0441
Os	1,4895	0,4051	0,7349
Us	0	0	0
Pk	6,0894	6,0243	6,0441
PT	363	283,05	144

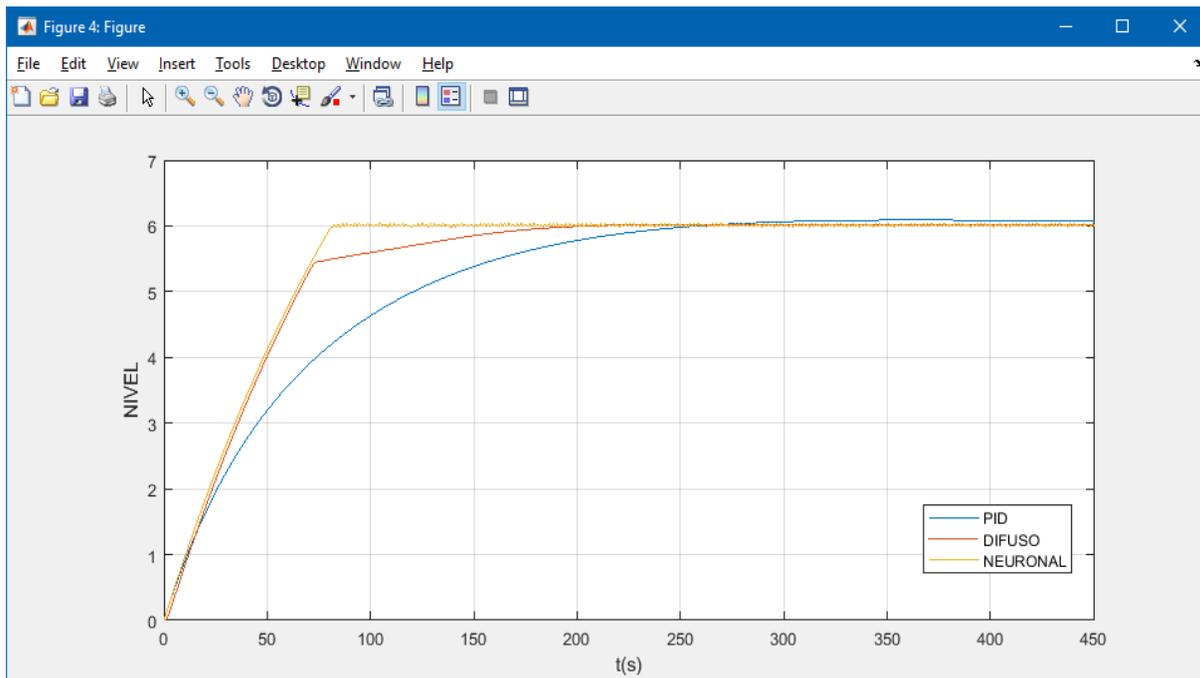


Figura 110. Respuesta de la simulación a un escalón de 6

En la Figura 111 se puede observar que el controlador PID provoca un crecimiento proporcional de la variable comparados a las consignas anteriores, con un tiempo de subida y de estabilización más reducidos; por otro lado, los controladores difuso y neuronal mantienen un comportamiento similar a los anteriores gráficos. En la parte difusa, se vuelve a observa el cambio en la pendiente de crecimiento alrededor de 0.5 litros antes del valor de consigna, y el controlador neuronal continua con la misma pendiente hasta llegar al valor de consigna y se estabiliza con una serie de oscilaciones. En la Tabla 10 se tiene el resultado de la función “stepinfo”, donde la estabilización más rápida la tuvo el controlador neuronal y la más lenta el PID; el mayor porcentaje de sobre impulso lo tuvo el PID, contrario al difuso que tuvo el menor; los parámetros difusos y neuronales son similares.

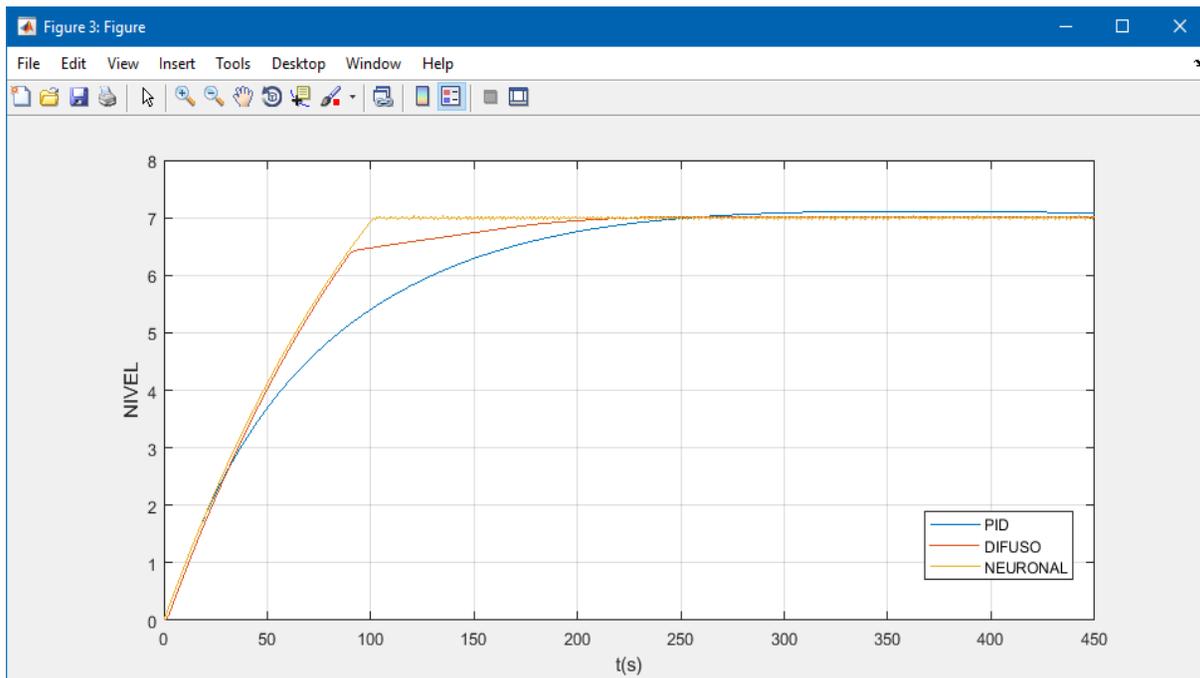


Figura 111. Respuesta de la simulación a un escalón de 7

Tabla 10

Respuesta de los controladores simulados para un Setpoint de 7.

Parámetro	PID	Difuso	Neuronal
RT	143,3733	79,7218	79,7209
ST	185,0477	132,6553	94,0272
Smin	6,3015	6,3015	6,3012
Smax	7,1115	7,0243	7,0302
Os	1,5928	0,3472	0,4315
Us	0	0	0
Pk	7,1115	7,0243	7,0302
PT	357,9	305,1	352,5

En la Figura 112 se puede observar que el controlador PID se empieza a ajustar y aproximar a los otros dos controladores, con un tiempo de subida y de estabilización más reducidos; por otro lado, los controladores difuso y neuronal mantienen su

comportamiento. En la parte difusa, se vuelve a observar el cambio en la pendiente de crecimiento alrededor de 0.5 litros antes del valor de consigna, y el controlador neuronal continua con la misma pendiente hasta llegar al valor de consigna y se estabiliza con una serie de oscilaciones. En la Tabla 11 se tiene el resultado de la función “stepinfo”, donde la estabilización más rápida la tuvo el controlador neuronal y la más lenta el PID; el mayor porcentaje de sobre impulso lo tuvo el PID, contrario al difuso que tuvo el menor; los parámetros difusos y neuronales son similares.

Tabla 11

Respuesta de los controladores simulados para un Setpoint de 8.

Parámetro	PID	Difuso	Neuronal
RT	139,5891	97,6129	97,6129
ST	179,6076	147,8672	115,3378
Smin	7,2021	7,2052	7,2049
Smax	8,1569	8,0243	8,0392
Os	1,9613	0,3038	0,4901
Us	0	0	0
Pk	8,1569	8,0243	8,0392
PT	343,65	330	237

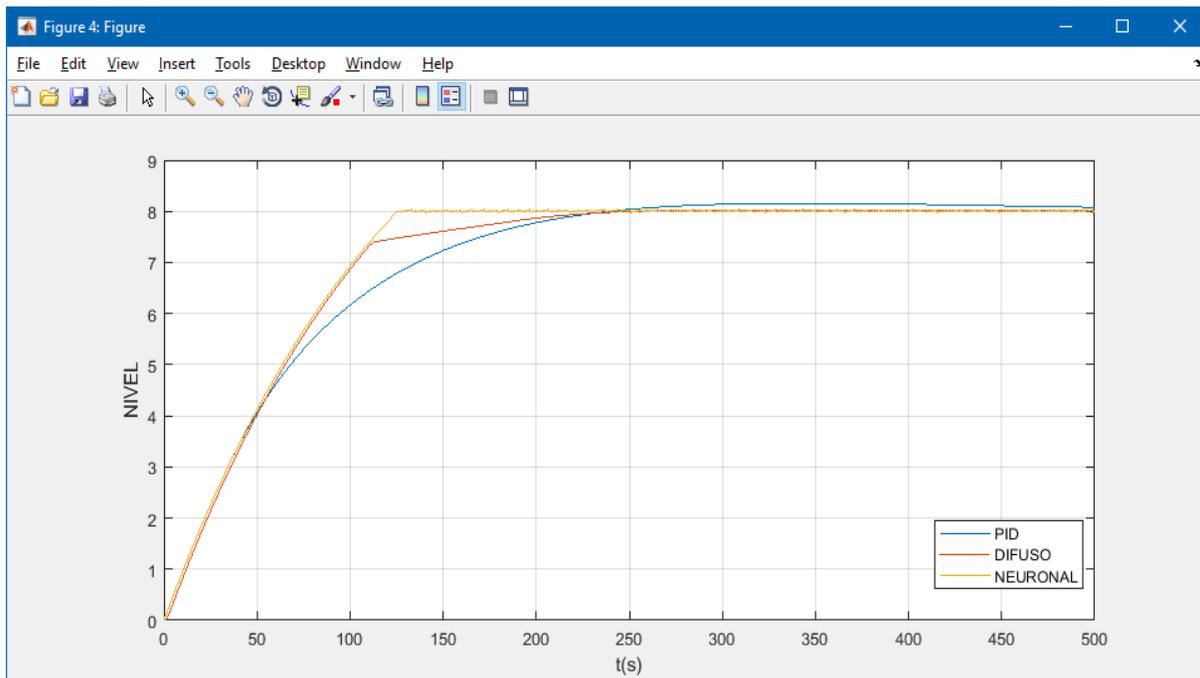


Figura 112. Respuesta de la simulación a un escalón de 8

En la Figura 113 se puede observar que el controlador PID se empieza a ajustar y aproximar a los otros dos controladores, con un tiempo de subida y de estabilización más reducidos; por otro lado, los controladores difuso y neuronal mantienen su comportamiento. En la parte difusa, se vuelve a observar el cambio en la pendiente de crecimiento alrededor de 0.5 litros antes del valor de consigna, y el controlador neuronal continua con la misma pendiente hasta llegar al valor de consigna y se estabiliza con una serie de oscilaciones. En la Tabla 12 se tiene el resultado de la función “stepinfo”, donde la estabilización más rápida la tuvo el controlador neuronal y la más lenta el PID; el mayor porcentaje de sobre impulso lo tuvo el PID, contrario al difuso que tuvo el menor; los parámetros difusos y neuronales son similares.

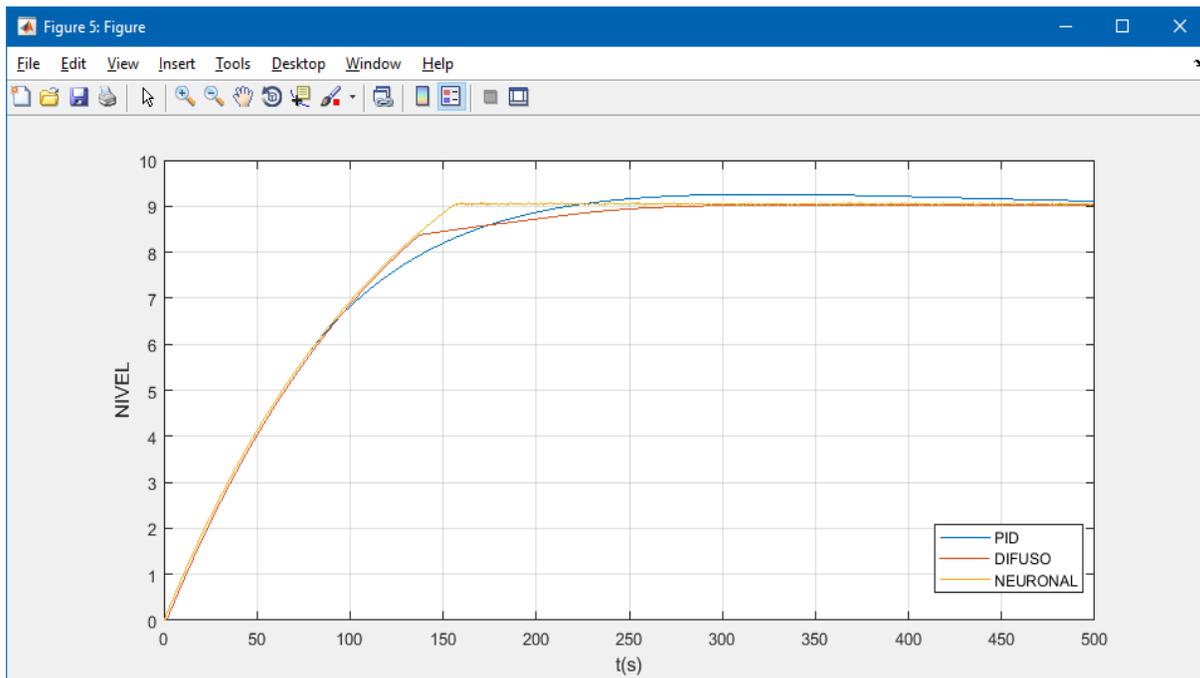


Figura 113. Respuesta de la simulación a un escalón de 9

Tabla 12

Respuesta de los controladores simulados para un Setpoint de 9.

Parámetro	PID	Difuso	Neuronal
RT	135,6488	118,7211	118,7211
ST	171,6933	167,0983	140,8125
Smin	8,103	8,1049	8,1046
Smax	9,2603	9,0243	9,0801
Os	2,8921	0,2701	0,8896
Us	0	0	0
Pk	9,2603	9,0243	9,0801
PT	321	358,65	252

a. Resumen de los controladores simulados

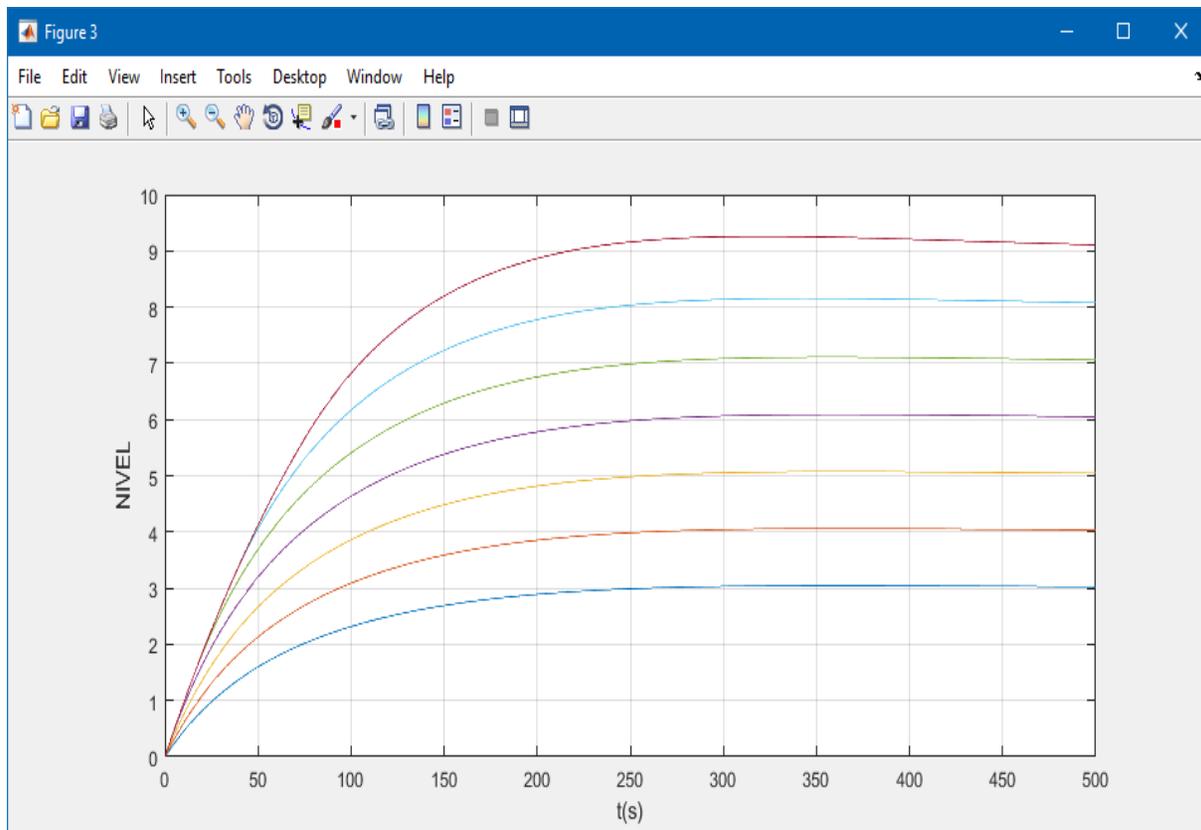


Figura 114. Respuesta del controlador PID simulado a los escalones

Como resumen de la Figura 114, se puede observar que el control PID toma un gran tiempo de subida, y por ende de establecimiento, lo cual podemos atribuirlo a las constantes de parametrización, ya que, con el fin de no obtener un pico demasiado elevado, reduce el tiempo de subida de la planta. Como principal característica de este controlador, se menciona que su porcentaje de sobre impulso promedio es de 1.77%, lo cual puede considerarse como aceptable. A medida que el setpoint va creciendo, la diferencia que existe en el tiempo de establecimiento con los otros dos tipos de controladores se fue reduciendo notablemente.

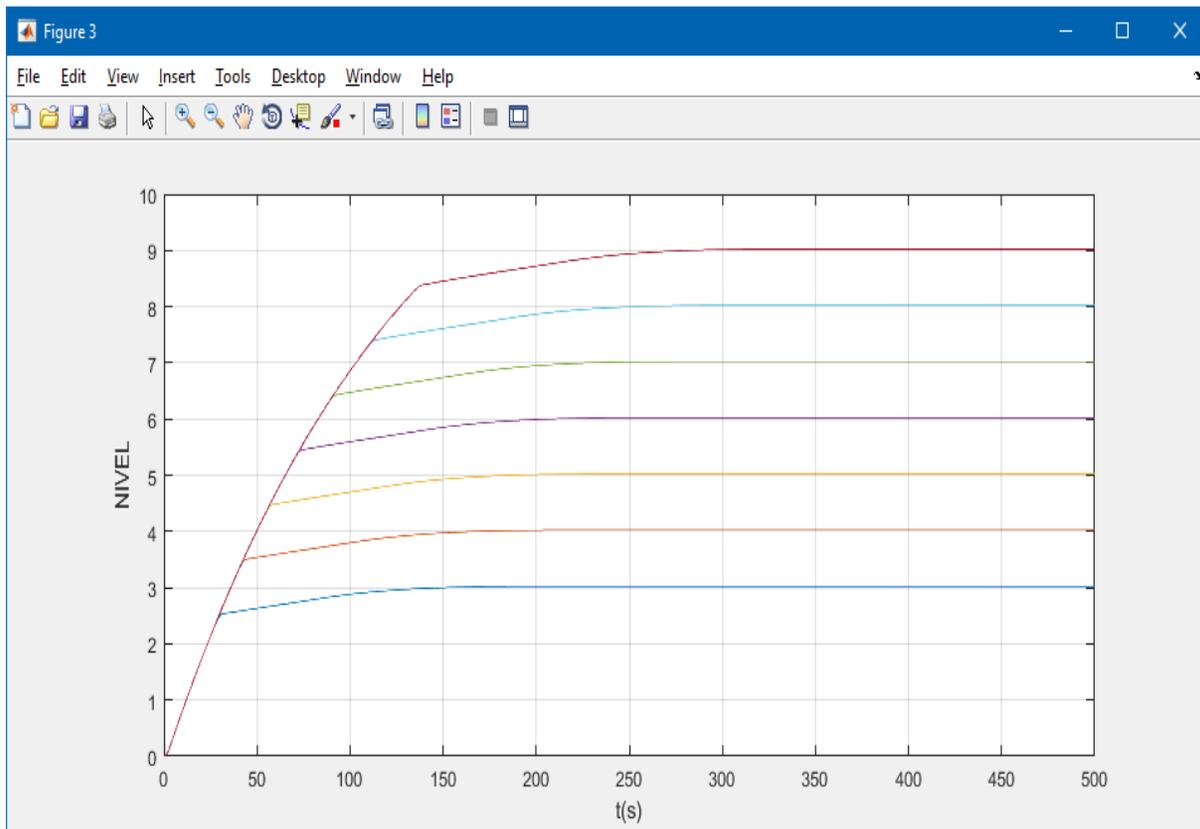


Figura 115. Respuesta del controlador difuso simulado a los escalones

En lo que respecta al controlador difuso, se obtiene un tipo de respuesta muy notorio y diferenciado a los otros dos, como se puede observar en la Figura 115. El tiempo de subida fue mucho más rápido comparado al controlador PID, así mismo su tiempo de establecimiento, además de mantener constante la pendiente de crecimiento de la variable controlada sea cual sea el valor de consigna; pero su característica principal es que, poco antes de llegar a dicho valor, reduce el crecimiento de la variable, provocando una nueva recta de aproximación y por ende suavizando el establecimiento de la misma, derivando en ser el controlador con la respuesta que posee el menor porcentaje de sobre impulso, de 0.46%. Este comportamiento no lineal es producto del conjunto de reglas de inferencia establecidas en el desarrollo del controlador difuso.

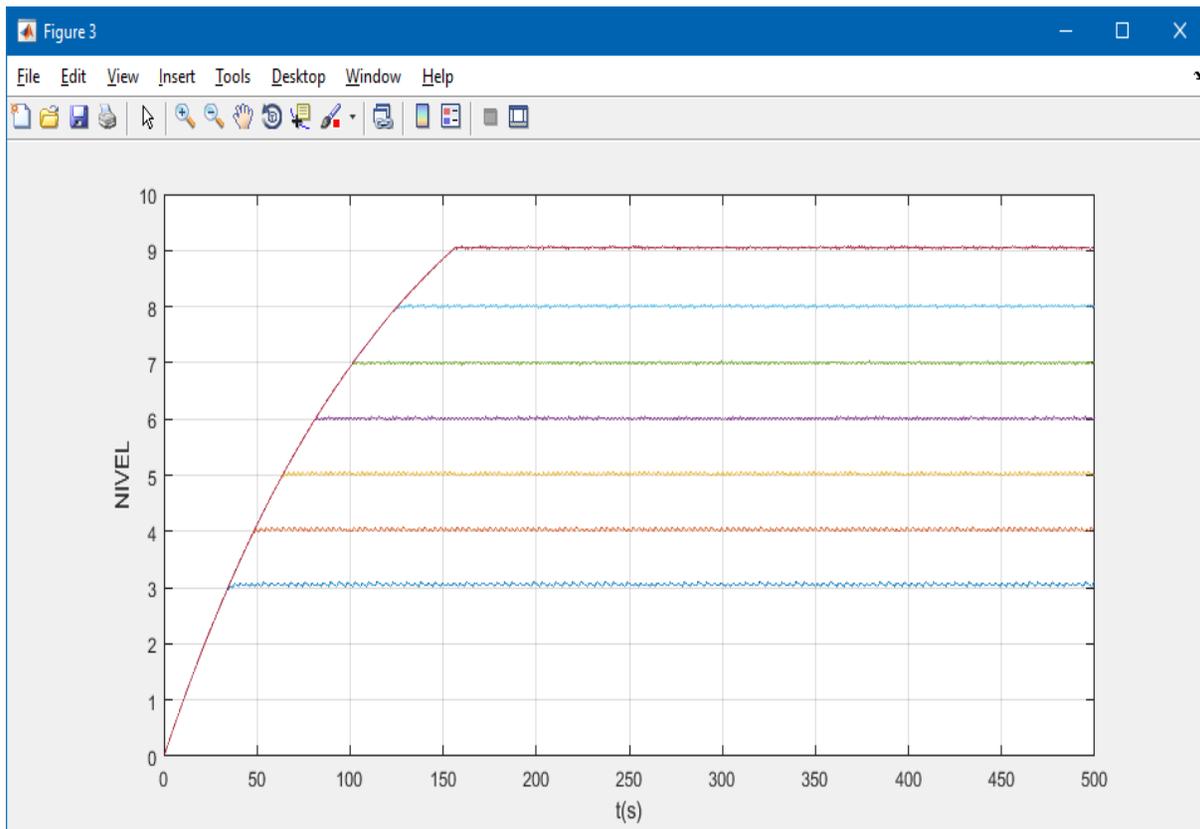


Figura 116. Respuesta del controlador neuronal simulado a los escalones

Finalmente, lo que corresponde al controlador neuronal, resumido en la Figura 116, al igual que los otros dos controladores, muestra un patrón de comportamiento constante para cada una de las diferentes consignas. En general, este controlador obtuvo el tiempo de establecimiento más rápido en comparación a los otros dos, aunque el tiempo de subida es equiparable al del control difuso. Su principal característica es que el cambio en el crecimiento de la variable una vez llegado al setpoint es muy veloz, provocando el que el porcentaje de sobre impulso sea solamente del 1.34%, menor que el obtenido por el control PID. Otra característica importante es que la variable controlada se mantiene oscilando alrededor del nivel de consigna, pero su amplitud es realmente reducida, como se puede notar en cada uno de los porcentajes de sobre impulso de las respuestas.

5.12.3. Resultados de los controladores implementados

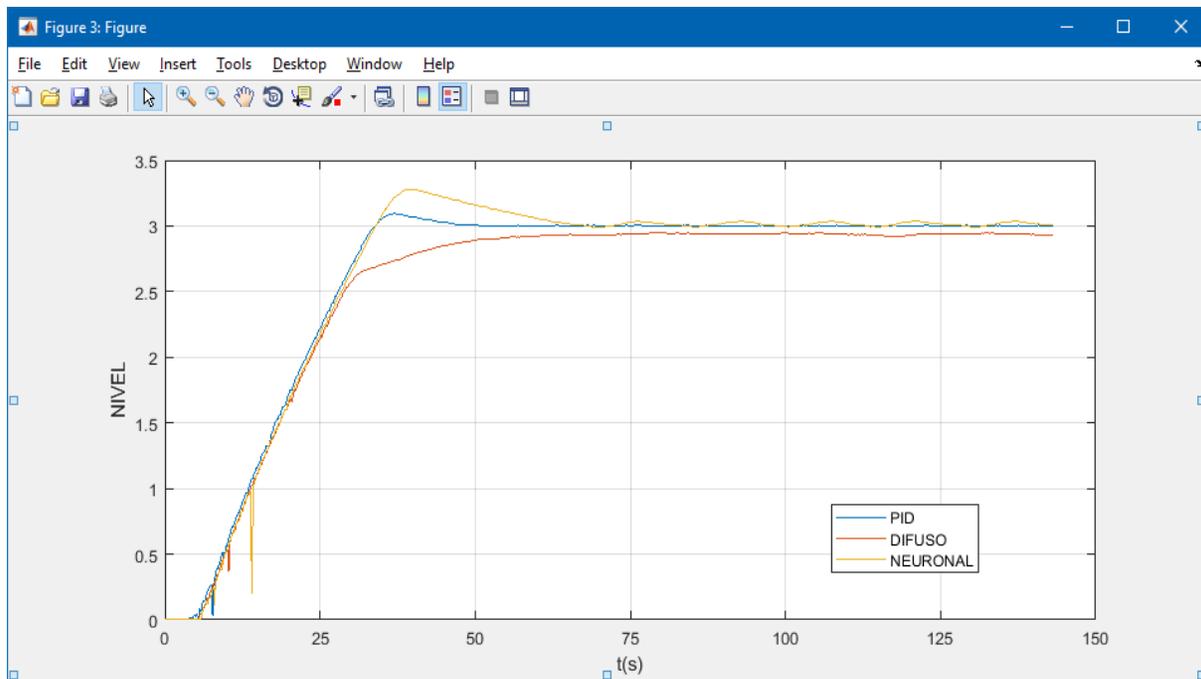


Figura 117. Respuesta de la planta a un escalón de 3

Tabla 13

Respuesta de la planta para un Setpoint de 3.

Parámetro	PID	Difuso	Neuronal
RT	22,0697	26,3685	22,2915
ST	30,0442	34,5321	30,5862
Smin	2,7053	2,7035	2,702
Smax	3,0981	2,9495	3,2809
Os	3,2704	0	9,3629
Us	0	0	0
Pk	3,0981	2,9495	3,2809
PT	36,9	78,8	39,1

En la Figura 117 se puede observar que el controlador PID provoca un crecimiento máximo de la variable, lo que se traduce en un tiempo de subida y de estabilización

reducidos, con un ligero sobre impulso; en la parte difusa, se observa un cambio en la pendiente de crecimiento alrededor de 2.5 litros, desde el cual el nivel crece más lentamente, mientras que el neuronal continua con la misma pendiente de crecimiento luego de alcanzado el valor de consigna y se estabiliza después de un considerable sobre impulso con una serie de oscilaciones. En la Tabla 13 se tiene el resultado de la función “stepinfo” para cada uno de los controladores, donde la estabilización más rápida la tuvo el controlador PID, el mayor porcentaje de sobre impulso le corresponde al controlador neuronal, mientras que el difuso no presenta ningún sobre impulso.

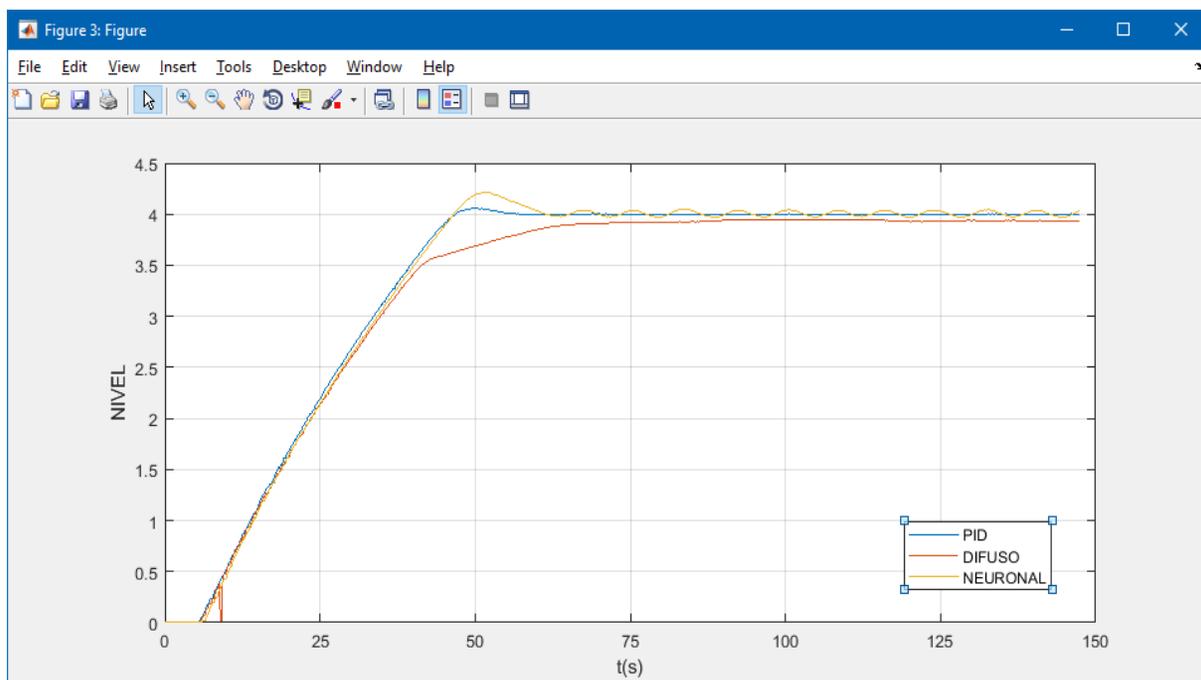


Figura 118. Respuesta de la planta a un escalón de 4

En la Figura 118 se puede observar que el controlador PID provoca un crecimiento máximo de la variable, lo que se traduce en un tiempo de subida y de estabilización reducidos, con un menor sobre impulso; en la parte difusa, se observa un cambio en la pendiente de crecimiento alrededor de 3.5 litros, desde el cual el nivel crece más

lentamente, mientras que el neuronal continua con la misma pendiente de crecimiento luego de alcanzado el valor de consigna y se estabiliza con un menor sobre impulso y una serie de oscilaciones. En la Tabla 14 se tiene el resultado de la función "stepinfo" para cada uno de los controladores, donde la estabilización más rápida la tuvo el controlador PID, el mayor porcentaje de sobre impulso le corresponde al controlador neuronal, mientras que el difuso no presento ningún sobre impulso.

Tabla 14

Respuesta de la planta para un Setpoint de 4.

Parámetro	PID	Difuso	Neuronal
RT	31,9795	35,7942	32,0509
ST	40,8467	45,1087	41,4072
Smin	3,6047	3,605	3,6061
Smax	4,0607	3,952	4,2169
Os	1,5171	0	5,4233
Us	0	0	0
Pk	4,067	3,952	4,2169
PT	49,6	96,6	51,6

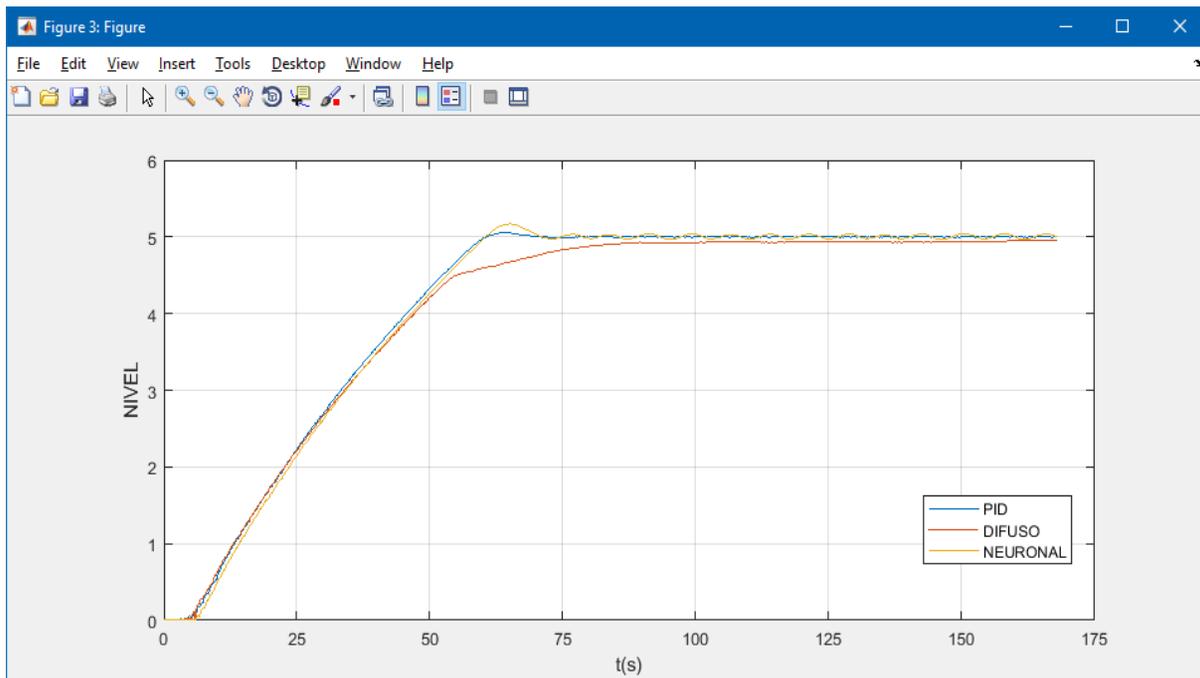


Figura 119. Respuesta de la planta a un escalón de 5

En la Figura 119 se puede observar que el controlador PID provoca un crecimiento máximo de la variable, lo que se traduce en un tiempo de subida y de estabilización reducidos, con un menor sobre impulso; en la parte difusa, se observa un cambio en la pendiente de crecimiento alrededor de 0.5 litros antes del valor de setpoint, desde el cual el nivel crece más lentamente, mientras que el neuronal continua con la misma pendiente de crecimiento luego de alcanzado el valor de consigna y se estabiliza con un menor sobre impulso y una serie de oscilaciones. En la Tabla 15 se tiene el resultado de la función “stepinfo” para cada uno de los controladores, donde la estabilización más rápida la tuvo el controlador PID, el mayor porcentaje de sobre impulso le corresponde al controlador neuronal, mientras que el difuso no presenta ningún sobre impulso.

Tabla 15

Respuesta de la planta para un Setpoint de 5.

Parámetro	PID	Difuso	Neuronal
RT	42,7873	45,905	43,1148
ST	52,4222	55,1238	53,2163
Smin	4,5065	4,5039	4,5058
Smax	5,057	4,9549	5,1711
Os	1,141	0	3,423
Us	0	0	0
Pk	5,057	4,9549	5,1711
PT	64,7	161,7	65,1

En la Figura 120 se puede observar que el controlador PID provoca un crecimiento máximo de la variable, al igual que los otros dos; en la parte difusa, se observa una ligera reducción en la velocidad de crecimiento de la variable, además de un cambio en la pendiente de crecimiento alrededor de 0.5 litros antes del valor de setpoint, desde el cual el nivel crece más lentamente, mientras que el neuronal continua con la misma pendiente de crecimiento hasta el valor de consigna y se estabiliza con un menor sobre impulso y una serie de oscilaciones. En la Tabla 16 se tiene el resultado de la función “stepinfo” para cada uno de los controladores, donde la estabilización más rápida la tuvo el controlador PID, el mayor porcentaje de sobre impulso le corresponde al controlador neuronal, mientras que el difuso no presento ningún sobre impulso.

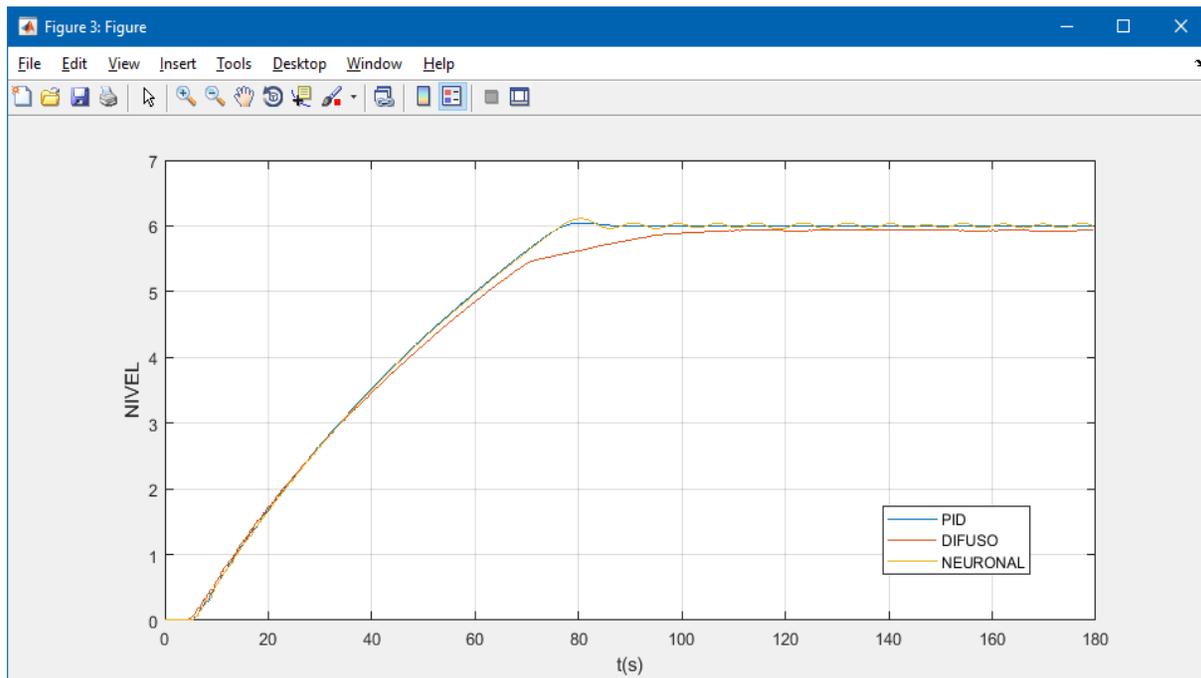


Figura 120. Respuesta de la planta a un escalón de 6

Tabla 16

Respuesta de la planta para un Setpoint de 6.

Parámetro	PID	Difuso	Neuronal
RT	55,9514	59,4519	56,0629
ST	66,3488	69,4	66,67
Smin	5,4015	54,0502	5,4008
Smax	6,0447	5,939	6,113
Os	0,7449	0	1,8835
Us	0	0	0
Pk	6,0447	5,939	6,113
PT	80	134,1	80,6

En la Figura 121 se puede observar que el controlador PID provoca un crecimiento máximo de la variable, al igual que en el neuronal; en la parte difusa, se observa un cambio en la pendiente de crecimiento alrededor de 0.5 litros antes del valor de setpoint,

desde el cual el nivel crece más lentamente, mientras que el neuronal alcanza el valor de consigna y se estabiliza con un menor sobre impulso y una serie de oscilaciones. En la Tabla 17 se tiene el resultado de la función “stepinfo” para cada uno de los controladores, donde la estabilización más rápida la tuvo el controlador neuronal, así como también el mayor porcentaje de sobre impulso, mientras que el difuso no presento ningún sobre impulso.

Tabla 17

Respuesta de la planta para un Setpoint de 7.

Parámetro	PID	Difuso	Neuronal
RT	70,558	75,9662	70,554
ST	81,8106	86,4611	81,7235
Smin	6,3052	6,302	6,3056
Smax	7,0414	6,9611	7,0959
Os	0,5918	0	1,3704
Us	0	0	0
Pk	7,0414	6,9611	7,0959
PT	99,1	179,7	98,3

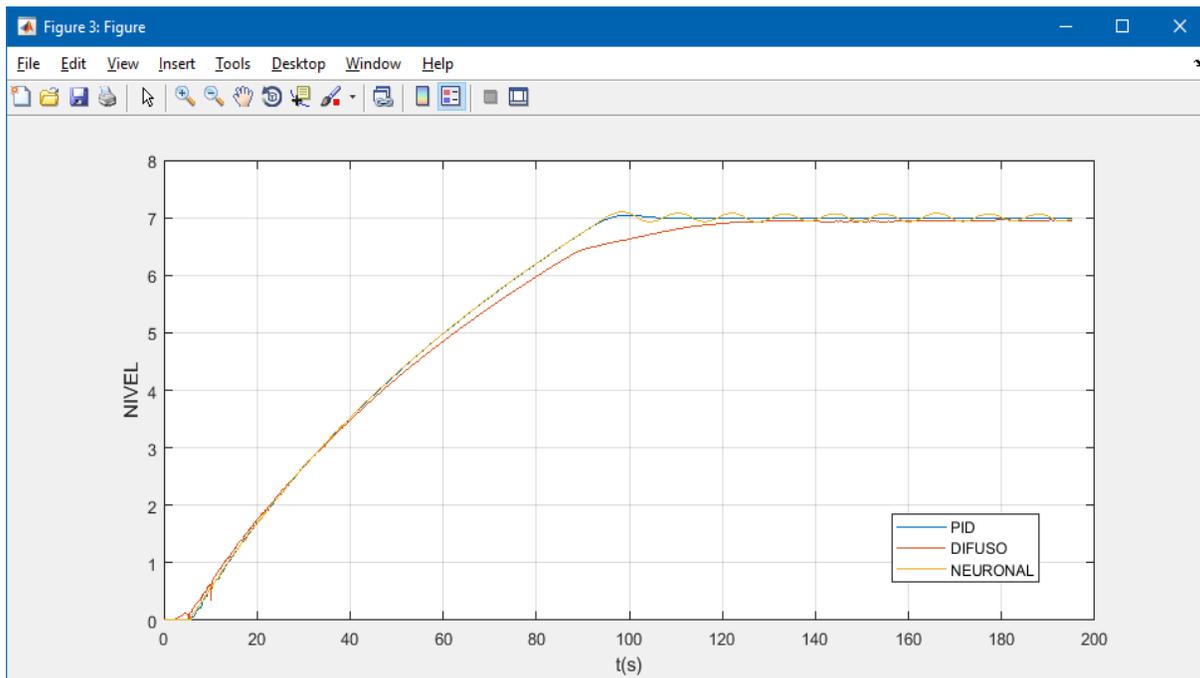


Figura 121. Respuesta de la planta a un escalón de 7

En la Figura 122 se puede observar que el controlador PID provoca un crecimiento máximo de la variable, al igual que en el neuronal; en la parte difusa, se observa un comportamiento más lento, además el punto de cambio en la pendiente de crecimiento alrededor se hace menos notorio, mientras que el neuronal alcanza el valor de consigna y se estabiliza con un menor sobre impulso y una serie de oscilaciones. En la Tabla 18 se tiene el resultado de la función “stepinfo” para cada uno de los controladores, donde la estabilización más rápida la tuvo el controlador neuronal, así como también el mayor porcentaje de sobre impulso, mientras que el difuso no presento ningún sobre impulso.

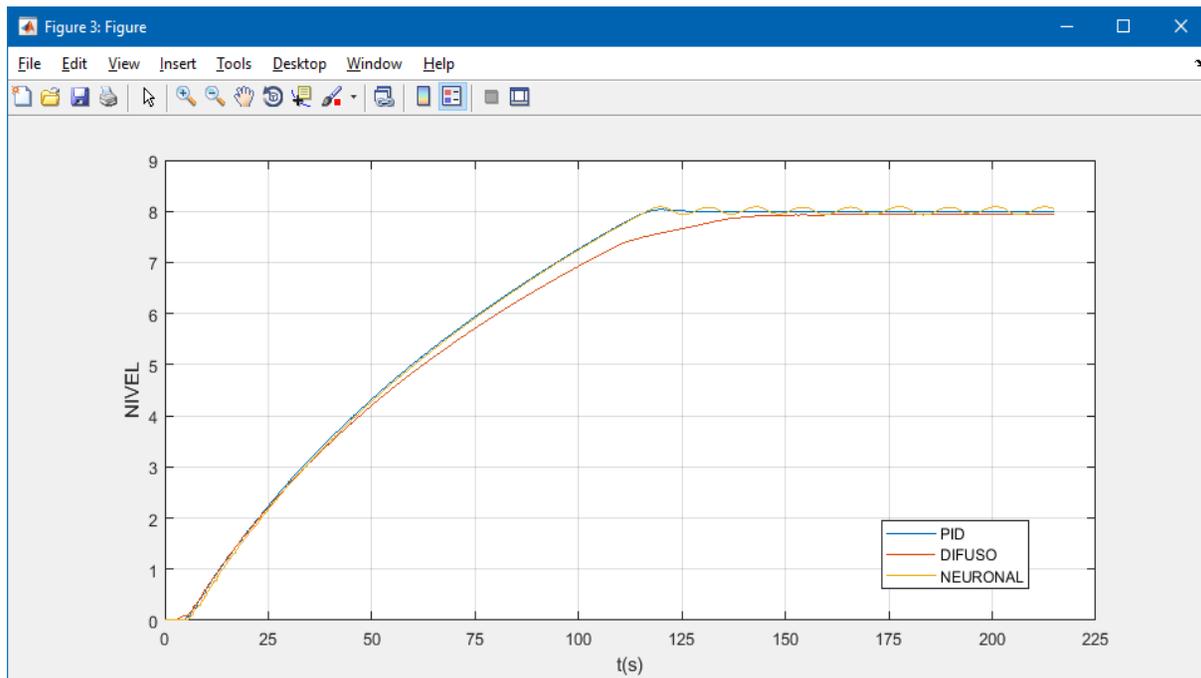


Figura 122. Respuesta de la planta a un escalón de 8

Tabla 18

Respuesta de la planta para un Setpoint de 8.

Parámetro	PID	Difuso	Neuronal
RT	87,1909	94,9358	86,5549
ST	98,6821	106,306	99,0962
Smin	7,2009	7,2035	7,2002
Smax	8,0356	7,9484	8,0919
Os	0,4451	0	1,1492
Us	0	0	0
Pk	8,0356	7,9484	8,0919
PT	120,5	172,6	177,4

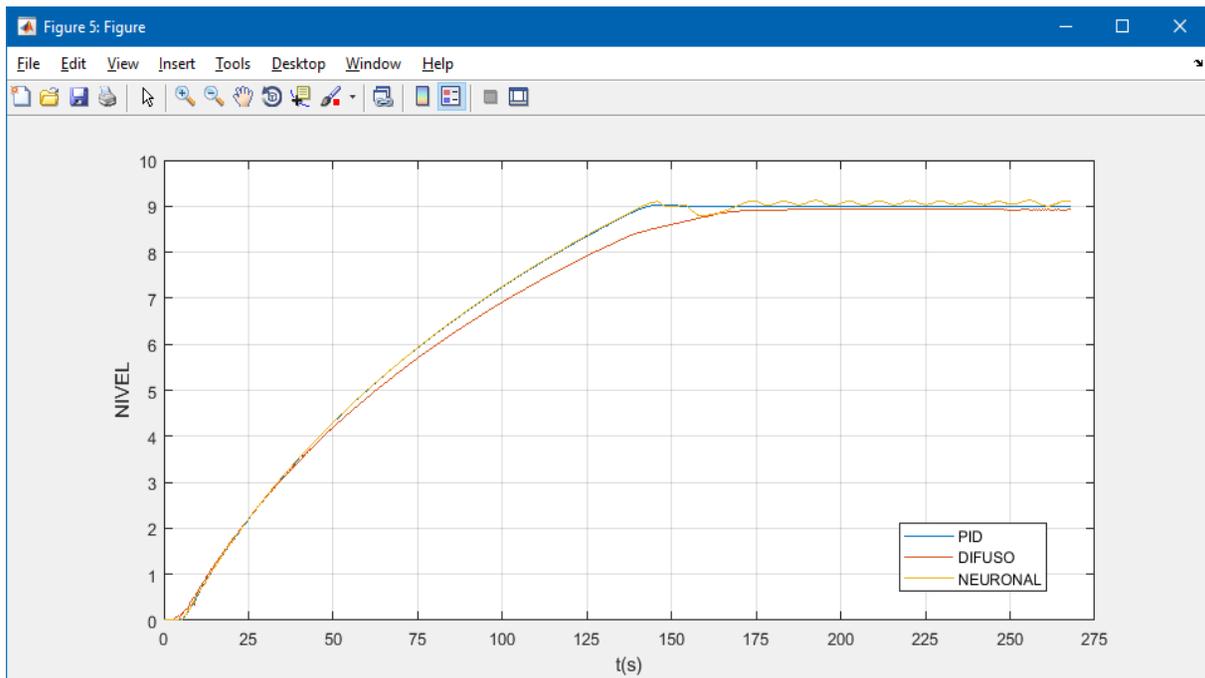


Figura 123. Respuesta de la planta a un escalón de 9

En la Figura 123 se puede observar que el controlador PID provoca un crecimiento máximo de la variable, al igual que en el neuronal; en la parte difusa, se observa un comportamiento más lento, además el punto de cambio en la pendiente de crecimiento alrededor se hace menos notorio, mientras que el neuronal alcanza el valor de consigna y se estabiliza con un menor sobre impulso y una serie de oscilaciones. En la Tabla 19 se tiene el resultado de la función “stepinfo” para cada uno de los controladores, donde la estabilización más rápida la tuvo el controlador neuronal, así como también el mayor porcentaje de sobre impulso, mientras que el difuso no presento ningún sobre impulso.

Tabla 19

Respuesta de la planta para un Setpoint de 9.

Parámetro	PID	Difuso	Neuronal
RT	106,2593	117,6899	105,6517
ST	119,0485	130,0275	118,5954
Smin	8,1025	8,1032	8,1003
Smax	9,032	8,944	9,1366
Os	0,3553	0	1,5181
Us	0	0	0
Pk	9,032	8,944	9,1366
PT	147,5	213,7	255,4

a. Resumen de los controladores implementados

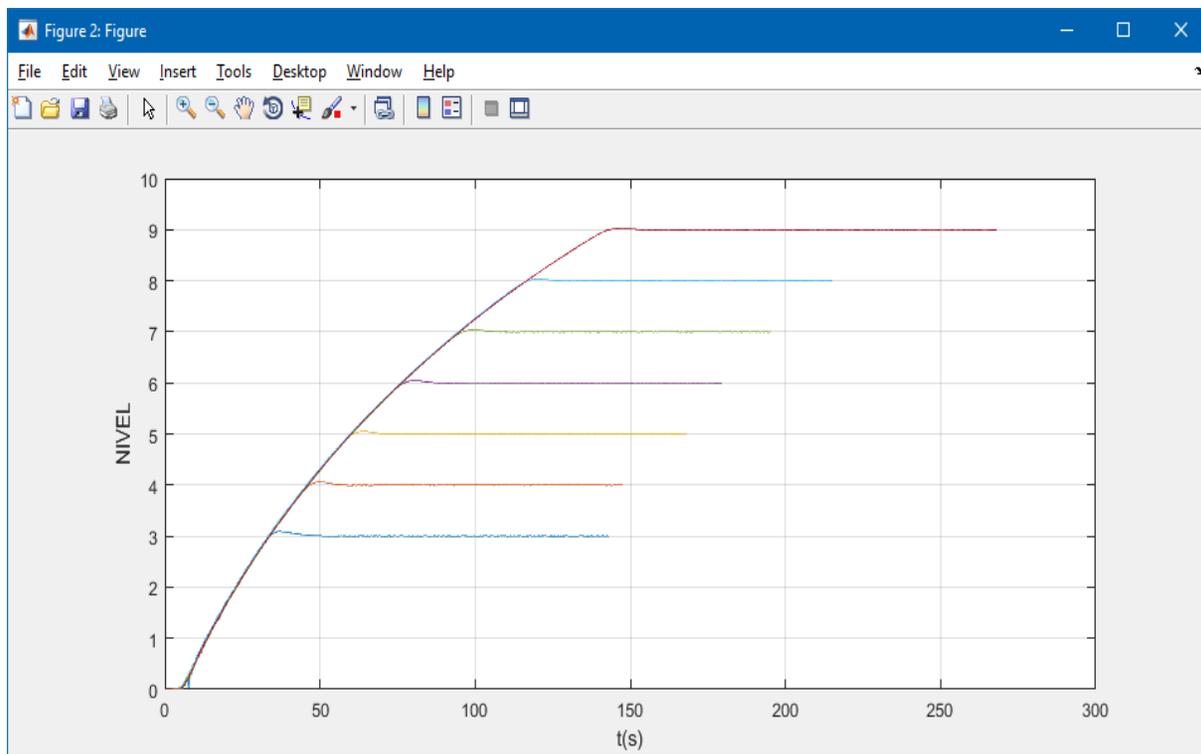


Figura 124. Respuesta del controlador PID implementado a los escalones

Comenzando por el controlador PID, podemos notar en la Figura 124 una clara diferencia en comportamiento, comparado con la Figura 114 que corresponde a la simulación del mismo. La pendiente de crecimiento de la variable controlada es constante en cada una de las consignas establecidas, por lo que su tiempo de subida y de establecimiento en la parte real de la planta mejora considerablemente. Su porcentaje de sobre impulso promedio se ubica aproximadamente en 1.15%, con un comportamiento de tipo inverso, ya que mientras mayor es el setpoint, menor sobre impulso se obtiene. Su tiempo de establecimiento es más rápido que el controlador difuso, aunque se encuentra igualado con el controlador neuronal.

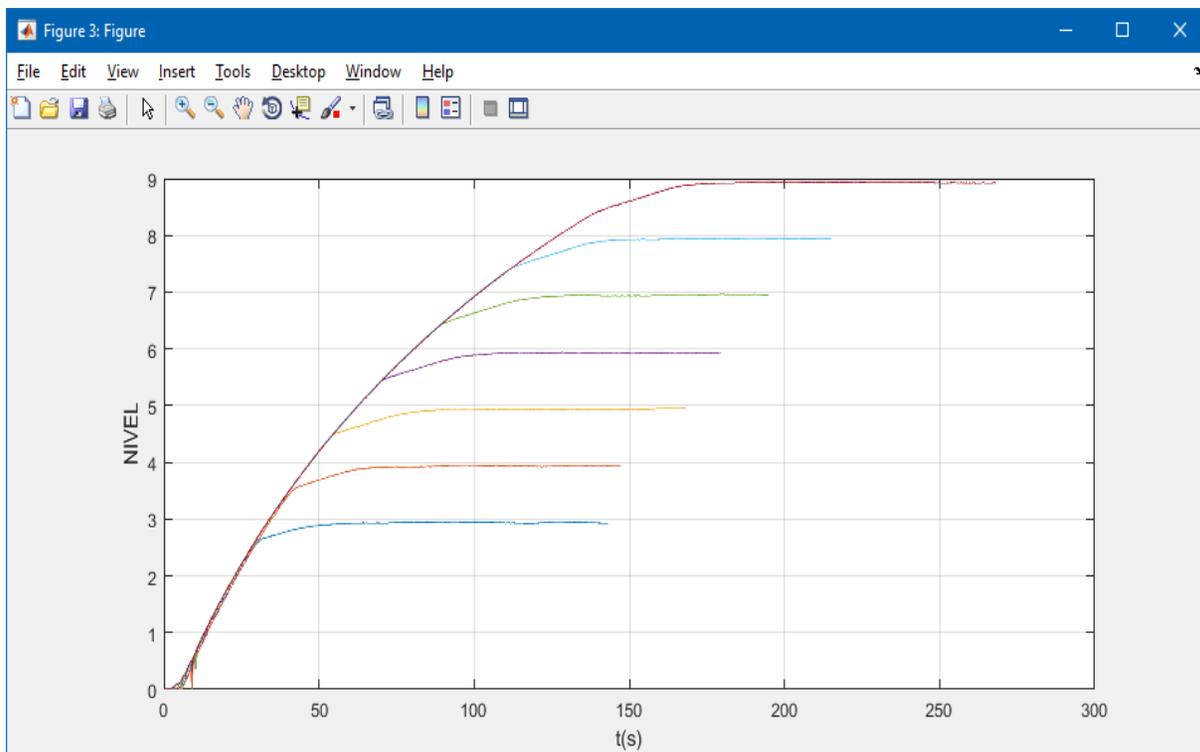


Figura 125. Respuesta del controlador difuso implementado a los escalones

El comportamiento del controlador difuso, mostrado en la Figura 125, se ajusta al obtenido en la parte de simulación: la pendiente de crecimiento de la variable controlada es constante para cada setpoint colocado, además de presentarse también el cambio de pendiente una vez que el valor de la variable esta próxima a la consigna establecida. En la parte real, este controlador presenta una característica muy llamativa, ya que a ningún momento el valor real de la variable supera la consigna, lo que se traduce en la eliminación de cualquier tipo de sobre impulso sobre la planta real. Su tiempo de establecimiento es mayor debido a que tarda más tiempo en alcanzar un estado estacionario por el hecho de no sobrepasar el valor de setpoint.

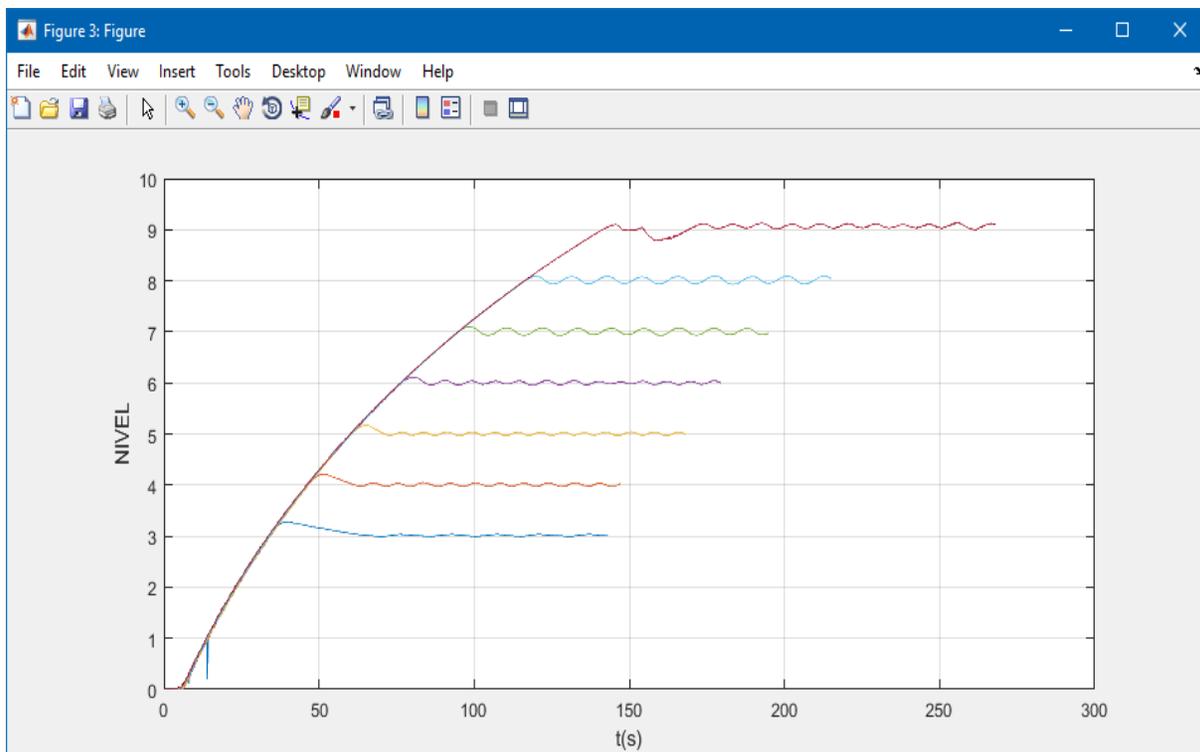


Figura 126. Respuesta del controlador neuronal implementado a los escalones

El comportamiento del controlador neuronal, mostrado en la Figura 126 , tiene ligeras variaciones en comparación a su contraparte simulada. En primer lugar, presenta una serie de sobre impulsos que se ven reducidos a medida que el nivel deseado aumenta, lo cual indica un mejor control para niveles altos del tanque; además mantiene presenta las oscilaciones alrededor del nivel de consigna establecido, de un periodo más amplio a comparación de los resultados simulados, esto debido a inercia del líquido impulsado por la bomba hidráulica. El porcentaje de sobre impulso promedio es el más elevado de los 3, aproximadamente un 3.44%, sin embargo, sigue siendo aceptable para este caso en específico.

CAPÍTULO VI

6. CONCLUSIONES Y RECOMENDACIONES

6.1. Conclusiones

- Los controladores PID basan su funcionamiento en la calibración y sintonización de los parámetros proporcional (k_p), integral (k_i) y derivativo (k_d), además de trabajar en diferentes combinaciones utilizando estas tres funcionalidades según lo requiera el proceso a controlar, por lo cual es necesario un conocimiento profundo de éste para lograr una sintonía adecuada; el controlador difuso depende de las funciones de membresía asignadas a cada una de las variables del proceso junto con la serie de reglas de inferencia difusa para la obtención del valor de salida, además del hecho que para su desarrollo no se requiere un conocimiento matemático del modelo a controlar; mientras que el controlador neuronal depende de los valores de entrenamiento obtenidos del proceso, junto con la cantidad de neuronas dentro de las capas ocultas, que permite obtener un comportamiento similar al proceso.
- Cada uno de los controladores PID, difuso y neuronal, fueron desarrollados utilizando la herramienta matemática Matlab y Simulink, con ayuda de las Toolbox Control System, Fuzzy Logic, Deep Learning Toolbox y Simulink PLC Coder; los cuales nos permiten modelar el proceso a controlar, generar los controladores, configurarlos de una manera intuitiva y simularlos para verificar su correcto funcionamiento, además de facilitar el traslado de los mismos de un entorno de

simulación a uno físico a través de la generación de código SCL para su implementación en un PLC, ahorrando tiempo de programación.

- En las mediciones realizadas a la variable de nivel de la planta industrial del Laboratorio de Hidrónica y Neutrónica, se observó que el control PID se adapta de manera aceptable al control del proceso, con un porcentaje de sobre impulso de 1.15%; por su parte el controlador difuso provocó dos tipos de pendiente de crecimiento de la variable: uno acelerado y cerca del valor de setpoint uno más lento, provocando una respuesta que no presenta porcentaje de sobre impulso, aunque su tiempo de establecimiento se aumenta; mientras que el controlador neuronal presento un porcentaje de sobre impulso del 3.44%, con valores elevados a consignas bajas, pero a valores altos de setpoint, su porcentaje de sobre impulso y su tiempo de establecimiento mejoran a comparación del controlador PID. Finalmente, la principal ventaja que presentan los controles inteligentes (difuso y neuronal) es la posibilidad de trabajar con sistemas MIMO, lo cual no puede ser realizado con un control PID regular como, por ejemplo, el control de una válvula proporcional hidráulica.

6.2. Recomendaciones

- Modificar las reglas de inferencia del control difuso, junto con el aumento de nuevas variables de entrada para modificar el comportamiento de controlador y adaptarlo para mejorar su respuesta hacia el proceso industrial, con el fin de

reducir el error en estado estable de la variable, así como un tiempo de establecimiento más rápido.

- Utilizar mayores datos de muestreo en el control neuronal, en un rango de respuesta más amplio para la obtención del modelo más cercano al físico; además, el tipo de red neuronal utilizado es uno predefinido por el entorno de Simulink, así que queda la posibilidad de mejorar el desempeño del controlador modificando la arquitectura de la red neuronal para añadir más variables de entrada.
- Cada uno de los controladores tienen sus ventajas respecto a los otros, por lo cual se recomienda analizar formas híbridas de éstos, como los son un controlador Neuro-PID, donde las constantes del PID provienen de una red neuronal establecida; un controlador PID difuso donde las constantes provienen del proceso de inferencia difusa del sistema, o un controlador Neuro-Fuzzy, donde los parámetros de las funciones de membresía difusas se adaptan según la salida de una red neuronal.

BIBLIOGRAFÍA

- Alomía, L., & Bastidas, C. (2013). *Caracterización y evaluación del comportamiento de controladores inteligentes difusos y controladores PID para aplicaciones industriales*. Tesis de Maestría, Escuela Politécnica Nacional, Quito.
- Amores, A., & Quingatuña, O. (2017). *Diseño e implementación de un sistema SCADA, utilizando redes de comunicación industrial para el control distributivo de sistemas hidráulicos y neumáticos en el Laboratorio de Hidrónica y Neutrónica de la Universidad de las Fuerzas Armadas ESPE*. Trabajo de Titulación.
- Ardila, P. (2010). *Estado Actual y Futuro de la Ingeniería de Control*. Universidad Pontificia Bolivariana. Obtenido de <https://revistas.upb.edu.co/index.php/puente/article/viewFile/7287/6656>
- Automación Micromecánica. (2018). *Controlador Lógico Programable (PLC)*. Recuperado el 17 de Noviembre de 2018, de <http://www.microautomacion.com/capacitacion/Manual061ControladorLgicoProgramablePLC.pdf>
- Banks, W., & Hayward, G. (2002). *Fuzzy Logic in Embedded Microcomputers and Control Systems*. Ontario: Byte Craft Limited.
- Castro, M., Díaz, G., Sebastián, R., Mur, F., Sancristobal, E., Sempere, V., & Silvestre, J. (2012). *Comunicaciones Industriales: Sistemas distribuidos y Aplicaciones*. Madrid: Editorial UNED.
- Creus, A. (2011). *Instrumentación Industrial* (Octava ed.). Barcelona: Marcombo.
- Espina, J., García, J., & Larrañaga, J. (2011). *Herramientas de redes neuronales para ingeniería de procesos industriales*. Obtenido de https://www.researchgate.net/publication/268433885_Herramientas_de_redes_neuronales_para_ingenieria_de_procesos_industriales
- Fundación BBVA. (3 de Enero de 2013). *El inventor de la lógica difusa Lotfi Zadeh, Premio Fundación BBVA Fronteras del Conocimiento por hacer posible que ordenadores y máquinas funcionen y decidan como los humanos*. Recuperado el 4 de Enero de 2019, de Premios Fronteras del Conocimiento: <https://www.fbbva.es/noticias/zadeh-premio-fronteras-del-conocimiento-tecnologias-la-informacion-la-comunicacion/>

- Goebel , G. (01 de Junio de 2003). *An Introduction To Fuzzy Control Systems*. Obtenido de FAQs.ORG: <http://www.faqs.org/docs/fuzzy/>
- Guano, M., & Romero, D. (2014). *Diseño e Implementación de un Sistema de Monitoreo y Control Local y Remoto, utilizando los Protocolos Industriales Profibus DP y Profinet con PLC's S7-1200*. Proyecto de Titulación, Latacunga.
- Guzmán, D., & Castaño, V. (2006). La lógica difusa en ingeniería: Principios, aplicaciones y futuro. *Ciencia y Tecnología*.
- Intrave. (2017). *Aplicaciones del Control Avanzado de Procesos Industriales*. Obtenido de <http://intrave.com/aplicaciones-del-control-avanzado-de-procesos-industriales/>
- Kant, K. (Abril de 2011). *Computer-Based Industrial Control* (Segunda ed.). New Delhi: PHI Learning. Obtenido de http://www.dsea.unipi.it/Members/balestrinow/CP/file/introd_APC_SPC.pdf
- Kaschel, H., & Pinto, E. (s.f.). *Analisis del estado del arte de los buses de campo aplicados al control de procesos industriales*. Recuperado el 15 de Noviembre de 2018, de <http://ingeborda.com.ar/biblioteca/Biblioteca%20Internet/Articulos%20Tecnicos%20de%20Consulta/Instalaciones%20Electricas%20Industriales/Buses%20Industriales.pdf>
- Kuo, B. (2009). *Sistemas de Control Automatico*. Mexico D.F.: Prentice Hall.
- MathWorks, Inc. (2018). *Design NARMA-L2 Neural Controller in Simulink*. Obtenido de Documentation: <https://la.mathworks.com/help/deeplearning/ug/design-narma-l2-neural-controller-in-simulink.html>
- MathWorks, Inc. (Septiembre de 2018). *User's Guide*. Obtenido de Simulink® PLC Coder™: https://la.mathworks.com/help/pdf_doc/plccoder/plc_st.pdf
- MathWorks, Inc. (2018). *Foundations of Fuzzy Logic*. Obtenido de Documentation: <https://la.mathworks.com/help/fuzzy/foundations-of-fuzzy-logic.html>
- MathWorks, Inc. (2018). *MATLAB Product Description*. Obtenido de MATLAB Documentation: https://la.mathworks.com/help/matlab/learn_matlab/product-description.html?lang=en
- MathWorks, Inc. (2018). *Simulink*. Obtenido de Simulink Documentation: <https://la.mathworks.com/help/simulink/index.html>

- MathWorks, Inc. (2018). *What Is Fuzzy Logic?* Obtenido de Documentation: <https://la.mathworks.com/help/fuzzy/what-is-fuzzy-logic.html>
- Morales, C., & Moreno, J. (2017). *Identificación de un sistema de control de nivel y flujo utilizando redes neuronales artificiales*. Trabajo de grado, Universidad Distrital Francisco José de Caldas.
- Moreno, G. (2005). *Diseño e implementación de un sistema basado en una red neuronal no supervisada para el control de movimiento de un robot móvil*. Proyecto de grado, Escuela Politécnica del Ejército, Quito.
- Muñoz, P., & Pérez, C. (2016). *Diseño e implementación de una estación de trabajo compacta para la regulación de variables de control: nivel, caudal, presión y temperatura en lazo cerrado para el Laboratorio de Hidrónica y Neutrónica de la Universidad de las Fuerzas Armadas Extensión L*. Trabajo de titulación, Universidad de las Fuerzas Armadas, Latacunga.
- Ogata, K. (2010). *Ingeniería de control moderna*. Madrid: Pearson Educación.
- Pineda, M., & Vivas, A. (2008). *Control de un modelo aerodinámico aplicando sistemas difusos*. Proyecto de titulación, Escuela Politécnica Nacional.
- Ponce, P. (2010). *Inteligencia Artificial con Aplicaciones a la Ingeniería*. Mexico D.F.: Alfaomega.
- Rodríguez, A. (2008). *Comunicaciones Industriales*. Barcelona: Marcombo.
- Ruiz, H. (2007). *Control Difuso vs. Control PID: Análisis y Simulación Numérica*. Tesis de Grado, Universidad Autónoma del Estado De Hidalgo, Pachuca de Soto.
- Salas, R. (2009). *Banco de laboratorio para la realización de prácticas de control PID*. Monografía, Universidad Tecnológica de Bolívar, Cartagena de Indias. Obtenido de <http://biblioteca.unitecnologica.edu.co/notas/tesis/0053687.pdf>
- Siemens AG. (2006). *Fuzzy Control User Manual*. Obtenido de SIMATIC S7: https://cache.industry.siemens.com/dl/files/654/1135654/att_24995/v1/Fuzzy_e.pdf
- Siemens AG. (Septiembre de 2016). *S7-1200 Programmable controller. System Manual*. Obtenido de SIMATIC: https://cache.industry.siemens.com/dl/files/593/109741593/att_895681/v1/s71200_system_manual_en-US_en-US.pdf

Siemens AG. (Septiembre de 2016). *S7-1500 CPU 1511-1PN Manual*. Obtenido de SIMATIC:

https://cache.industry.siemens.com/dl/files/492/68020492/att_895876/v1/s71500_cpu1511_1_pn_manual_en-US_en-US.pdf

Universidad de Oviedo. (2006). *Comunicaciones Industriales*. Obtenido de Ingeniería Electrónica y Automática:

<http://isa.uniovi.es/docencia/iea/teoria/comunicacionesindustrialesdocumento.pdf>

Vera, C. (2009). *Sistemas Avanzados de Control*. Pamplona: Universidad Nacional Abierta y a Distancia - UNAD.

ANEXOS



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA ELECTROMECAÁNICA

CERTIFICACIÓN

Se certifica que el presente trabajo fue desarrollado por el señor Pacheco Taco, Diego Ernesto.

En la ciudad de Latacunga, a los veinte y cinco días del mes de enero de 2019.

Aprobado por:

Ing. Wilson Sánchez
DIRECTOR DEL PROYECTO

Ing. Katya Torres
DIRECTORA DE CARRERA

Dr. Rodrigo Vaza
SECRETARIO ACADÉMICO