



UNIVERSIDAD TÉCNICA DE AMBATO

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

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y COMUNICACIONES

Tema

**CONTROL NEURO-PID PARA UN SISTEMA DE TANQUES EN SERIE BAJO
LA NORMA IEC 61499**

Trabajo de Titulación Modalidad: Proyecto de Investigación, presentado previo a la obtención del título de Ingeniero en Electrónica y Comunicaciones.

ÁREA: Comunicaciones, Física y Electrónica.

LÍNEA DE INVESTIGACIÓN: Tecnologías de la comunicación
Y Sistemas electrónicos

AUTOR: José Raúl García Paredes

TUTOR: Ing. Mg. Paulina Ayala

Ambato – Ecuador

marzo - 2023

APROBACIÓN DEL TUTOR

En calidad de tutor del Trabajo de Titulación con el tema: CONTROL NEURO-PID PARA UN SISTEMA DE TANQUES EN SERIE BAJO LA NORMA IEC61499, desarrollado bajo la modalidad Proyecto de Investigación por el señor José Raúl García Paredes, estudiante de la Carrera de Ingeniería en Electrónica y Comunicaciones de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 15 del Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y el numeral 7.4 del respectivo instructivo.

Ambato, marzo 2023.

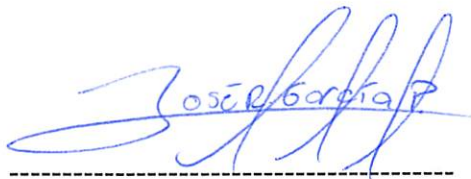
Ing. Mg. Paulina Ayala

TUTOR

AUTORÍA

El presente Proyecto de Investigación titulado CONTROL NEURO-PID PARA UN SISTEMA DE TANQUES EN SERIE BAJO LA NORMA IEC61499, es absolutamente original, auténtico y personal. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, marzo 2023



José Raúl García Paredes
C.C. 1804419255
AUTOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Titulación como un documento disponible para la lectura, consulta y procesos de investigación.

Cedo los derechos de mi Trabajo de Titulación en favor de la Universidad Técnica de Ambato, con fines de difusión pública. Además, autorizo su reproducción total o parcial dentro de las regulaciones de la institución.

Ambato, marzo 2023



José Raúl García Paredes
C.C. 1804419255
AUTOR

APROBACIÓN DEL TRIBUNAL DE GRADO

En calidad de par calificador del Informe Final del Trabajo de Titulación presentado por el señor José Raúl García Paredes, estudiante de la Carrera de Ingeniería en Electrónica y Comunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado CONTROL NEURO-PID PARA UN SISTEMA DE TANQUES EN SERIE BAJO LA NORMA IEC61499, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 17 del Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y al numeral 7.6 del respectivo instructivo. Para cuya constancia suscribimos, conjuntamente con la señora Presidenta del Tribunal.

Ambato, marzo 2023.

Ing. Pilar Urrutia, Mg

PRESIDENTA DEL TRIBUNAL

Ing. Franklin Salazar, Mg.
PROFESOR CALIFICADOR

Ing. Andrea Patricia Sánchez Zumba, Mg.
PROFESOR CALIFICADOR

DEDICATORIA

Esté camino fue muy difícil, en el cual me enamore del proceso, de mis resultados y esfuerzo que le dedique día tras día para convertirme en un profesional, más no de la forma en la que lo aprendí y en las cuales sacrifique muchas cosas y quizás la más importante de aquellas que fue mi salud mental, hoy al culminar este proceso se lo dedico a la persona que nunca dejo de tener fe en mí y no permitió que arrojara la toalla en más de una vez, mi amada e incondicional madre Mónica Paredes.

A su vez se la dedico a mi querido y amado Papá, José Alfredo García, que a su manera, me ha aconsejado y guiado, talvez con puntos de vista contrarios a mi criterio y forma de ver la vida, pero que siempre han aportado en mí un valor único y disruptivo.

A mi Abuelitos que agradezco a la vida por tenerlos aun junto a mi porque me han apoyado en cada paso que he dado en la vida.

José Raúl García Paredes

AGRADECIMIENTO

Estoy agradecido conmigo mismo por obtener la confianza y el valor de recuperar el tiempo perdido, para poder entregar este último esfuerzo, para mi orgullo y el de mi familia, pero hubo muchas personas que hicieron esto posible, a las cuales agradezco a continuación.

A toda mi familia por darme el apoyo necesario para continuar una carrera universitaria la cual es un privilegio acceder en Latinoamérica.

A los maestros que despertaron en mí el ingenio y retaron mis capacidades, las cuales fueron mejorando con el pasar de los años, hicieron que sea más fuerte y tenga un pensamiento crítico, analítico y la fiel creencia que un Ingeniero en Electrónica y Comunicaciones puede todo lo que se propone.

A mi tutora la Ing. Pauli que fue una verdadera guía y ayuda en este proceso. Ella siempre tuvo confianza en que lograría sacar adelante esta propuesta, gracias por guiarme en este proyecto.

A todos los que conforman el CBF Equilibrio, el cual me ayudo en el proceso de recuperar mi camino y aportarle valor y sentido a mi vida.

A mis amigos, a los que considero como familia y como mis hermanos, por estar ahí en los mejores y peores momentos (Los Gatos) y son la parte más valiosa que rescato de toda mi carrera Universitaria.

José Raúl García Paredes

ÍNDICE GENERAL

APROBACIÓN DEL TUTOR.....	ii
AUTORÍA.....	iii
DERECHOS DE AUTOR.....	iv
APROBACIÓN TRIBUNAL DE GRADO.....	v
DEDICATORIA	vi
AGRADECIMIENTO.....	vii
ÍNDICE GENERAL.....	viii
ÍNDICE DE TABLAS	xi
ÍNDICE DE FIGURAS.....	xii
RESUMEN EJECUTIVO	xv
ABSTRACT.....	xvi
INTRODUCCIÓN	17
CAPÍTULO I.....	19
MARCO TEÓRICO.....	19
1.1 Tema de Investigación.....	19
1.2 Antecedentes Investigativos	19
1.3 Contextualización del Problema.....	22
1.4 Fundamentación Teórica	23
1.4.1 Norma IEC-61499.....	23
1.4.2 Aspectos de la Norma IEC-61499.....	24
Aspectos de la Norma IEC-61499.....	24
1.4.3 La Norma IEC-499 y su estructura	25
1.4.4 Tipos de Modelos de la Norma IEC-61499	26
1.4.4.1 Modelo de bloque funcional (FB).....	26
1.4.4.2 Modelo de bloque Funcional de Interfaz de Servicio.....	28
1.4.4.3 Modelo de bloque de recurso.....	32
1.4.4.4 Modelo de bloque del dispositivo.....	33
1.4.4.5 Modelo de bloque del Sistema.....	34
1.4.4.6 Modelo de Bloque de aplicación y sub-aplicación.....	34
1.4.4.7 Herramientas de desarrollo	35
1.4.4.8 Plataformas de cumplimiento o ejecución	36
1.4.8.9 Redes Neuronales Artificiales.	38

1.4.8.10	Controlador PID.....	41
1.4.8.11	Control PID por medio de redes neuronales.....	41
1.4.8.12	Entrenamiento de una red neuronal.....	42
1.4.8.13	Propuesta de solución.....	43
1.5	Objetivos.....	43
1.5.8	Objetivo general.....	43
1.5.9	Objetivos Específicos.....	43
CAPÍTULO II.....		45
METODOLOGÍA.....		45
Materiales.....		45
Métodos.....		45
2.2.1	Modalidad de Investigación.....	45
2.2.2	Recolección de Información.....	46
2.2.3	Procesamiento y análisis de datos.....	46
2.2.4	Propuesta de solución.....	46
CAPÍTULO III.....		48
RESULTADOS Y DISCUSIÓN.....		48
3.1	Análisis y discusión de los resultados.....	48
3.2	Desarrollo de la propuesta.....	48
3.2.1	Descripción del módulo de nivel de tanques FESTO® MPS-PA, para la implementación del sistema de control.....	50
3.2.2	Diagrama de bloques del sistema de control.....	51
3.2.3	Diagrama PI&D de flujo de nivel.....	52
3.2.4	Selección de herramientas de software.....	53
3.2.4.1	Selección de la plataforma de desarrollo.....	53
3.2.4.2	Selección de la plataforma de composición (runtime).....	54
3.2.4.3	Selección de plataformas de creación para redes neuronales.....	54
3.2.4.4	Selección de elementos de hardware.....	55
3.2.4.5	Desarrollo del circuito de entradas y salidas del sistema de control. 56	
3.2.4.6	Esquema del Circuito de acondicionamiento de salida PWM.....	57
3.2.4.7	Diagrama del circuito de adaptación.....	57
3.2.4.8	Circuito de acondicionamiento para alimentación del procesador.....	58
3.2.4.9	Diagrama de bloques del sistema para la adquisición de datos....	58

3.2.4.10	Diseño de bloques funcionales en 4DIAC bajo la norma IEC-61499 59	
3.2.4.11	Bloque de función del controlador modelado de ejecución.....	61
3.2.4.12	Bloques de funciones	61
3.2.4.13	Aplicación de la norma IEC-61499 para crear el módulo PID.....	62
3.2.4.14	Gráfica de respuesta del proceso ante el impulso escalón	63
3.2.4.15	Implementación del algoritmo de control PID	64
3.2.4.16	Programación en Tía Portal	67
3.2.4.17	Diseño de la red neuronal	68
3.2.4.18	Datos de Aprendizaje para la red Neuronal	69
3.2.4.19	Implementación red neuronal	70
3.2.5	Pruebas de funcionamiento	77
CAPÍTULO IV		81
CONCLUSIONES Y RECOMENDACIONES		81
4.1	Conclusiones.....	81
4.2	Recomendaciones	82

ÍNDICE DE TABLAS

Tabla 1.	Estructura de la Norma	26
Tabla 2.	Eventos y datos de entrada del bloque funcional de interfaz de servicio bajo la norma IEC-61499.....	29
Tabla 3.	Eventos y datos de salida del bloque funcional de interfaz de servicio bajo la norma IEC-61499 [14].....	30
Tabla 4.	Diseño de conformación del Módulo de tanques FESTO® MPS-PA.....	51
Tabla 5.	Dispositivos controladores que trabajan con 4DIAC-IDE y FORTE.....	56
Tabla 6.	Valores de entradas y salidas del sistema.....	77
Tabla 7.	Valores establecidos y valores arrojados por la red neuronal.....	78

ÍNDICE DE FIGURAS

Figura 1.	Aspectos de la Norma IEC-61499.....	24
Figura 2.	Aspectos representados en la norma IEC-61499.	25
Figura 3.	Modelo de bloque (FB).....	27
Figura 4.	Estructura de bloque de función [11].....	28
Figura 5.	Estructura de un bloque funcional de interfaz de servicio SIFB [13]..	29
Figura 6.	Bloque funcional de interfaz SIFB Solicitante y Respondedor [14]....	30
Figura 7.	Paradigmas de comunicación bloques de Publish y Subscribe [14]. ...	31
Figura 8.	Interfaz de bloques de comunicación de Client-Server [14].....	32
Figura 9.	Interfaz del modelo de bloque de recurso [14].....	33
Figura 10.	Estructura del modelo del dispositivo [15].	33
Figura 11.	Interfaz de modelo del sistema bajo la norma IEC-61499 [16]	34
Figura 12.	Estructura de un modelo de aplicación y sub-aplicación [16].	35
Figura 13.	Entorno de 4DIC-IDE basada en la Norma IEC-61499[17].	36
Figura 14.	Plataforma FORTE y 4DIAC bajo la Norma IEC-61499 [18].	38
Figura 15.	Esquema de las redes neuronales [21].	39
Figura 16.	Tipos de funciones de transferencia [22].	40
Figura 17.	Clasificación de redes neuronales.	40
Figura 18.	Diagrama del sistema propuesto	49
Figura 19.	Módulo de tanques FESTO® MPS-PA.	50
Figura 20.	Diagrama de bloques del sistema de control.....	51

Figura 21.	Diagrama PI&D de flujo de nivel.	52
Figura 22.	Plataforma de desarrollo 4DIAC-IDE [26].	53
Figura 23.	Plataforma de composición 4DIAC-FORTE [27].	54
Figura 24.	Plataforma de creación de redes neuronales Python [28].	55
Figura 25.	Circuito de regulación PWM	57
Figura 26.	Circuito de adaptación de tensión de 0-5V	57
Figura 27.	Circuito de acondicionamiento para alimentación del procesador.	58
Figura 28.	Diagrama de bloques del sistema para la adquisición de datos.	59
Figura 29.	Gráfico del control de ejecución del controlador PID del sistema.	60
Figura 30.	Bloques de funciones desarrollados para lecturas y escrituras analógicas.	62
Figura 31.	Bloque PID.	63
Figura 32.	Bloque de la red neuronal	63
Figura 33.	Grafica con los datos de Arduino.	64
Figura 34.	Respuesta al impulso escalón en Arduino.	64
Figura 35.	Valores procesados mediante Matlab.	65
Figura 36.	Valores procesados mediante Matlab.	65
Figura 37.	Gráfica de la función de transferencia	66
Figura 38.	Grafica de la amplitud en función del tiempo.	66
Figura 39.	Grafica de respuesta mediante las constantes PID.	67
Figura 40.	Lectura del nivel del Tanque.	67

Figura 41.	Salida analógica para los valores de PWM.	68
Figura 42.	Esquema de la red neuronal aplicada al controlador del sistema.	69
Figura 43.	Esquema del control PID del módulo con la red neuronal.....	69
Figura 44.	Recolección de datos para aprendizaje para la predicción de Kp Ki Kd.....	70
Figura 45.	Código de la NN basado en Keras para la predicción de Kp Ki Kd	70
Figura 46.	Datos de medición de entrada y salida del sistema	71
Figura 47.	Importación de datos para crea la red neuronal	72
Figura 48.	Ventana de interfaz para crear una red neuronal.....	72
Figura 49.	Topología de la red neuronal.....	73
Figura 50.	Configuración de la red neuronal.....	73
Figura 51.	Entrenamiento de la neurona con 1000 interacciones.....	74
Figura 52.	Eficiencia y rendimiento de la neurona.....	75
Figura 53.	Esquema PCB del circuito aplicado al sistema de control.....	75
Figura 54.	Circuito ensamblado aplicado al sistema de control.....	76
Figura 55.	Programación referente al envió de PWM.....	76
Figura 56.	Prueba de funcionamiento de red neuronal.....	77
Figura 57.	Comprobación de rango de nivel de control	78
Figura 58.	Código de comando para pasar los datos generados.....	79

RESUMEN EJECUTIVO

El presente proyecto de titulación es un aporte en el área de investigación enfocado en el proceso industrial de reconfiguración dinámica de control automático, basado en el estándar IEC-61499 y multiagentes, utilizando un entorno de desarrollo denominado 4DIAC (4DIAC-IDE) y la herramienta de runtime FORTE (4DIAC-RTE), para el control de nivel de líquido en tanques, utilizando la tecnología actual como es el módulo FESTO® MPS-PA Compact Workstation del laboratorio de Procesos Industriales de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato, con el presente proyecto se desea implementar y diseñar el control Neuro-PID basado en redes neuronales, debido a las capacidades de retención de información y manejo de sistemas que ofrece en los procesos de control.

Palabras clave: Dinámica, multiagentes, 4DIAC, FORTE, Neuro-PID

ABSTRACT

This degree project is a contribution in the area of research focused on the industrial process of dynamic reconfiguration of automatic control, based on the IEC-61499 standard and multi-agent, using a development environment called 4DIAC (4DIAC-IDE) and the FORTE runtime tool (4DIAC-RTE), for the control of liquid level in tanks, using current technology such as the FESTO® MPS-PA Compact Workstation module of the Industrial Processes laboratory of the School of Systems, Electronics and Industrial Engineering of the Technical University of Ambato. With this project we want to implement and design the Neuro-PID control based on neural networks, due to the capabilities of information retention and management of systems that offers in the control processes.

Keywords: Dynamics, multiagent, 4DIAC, FORTE, Neuro-PID.

INTRODUCCIÓN

El presente proyecto de investigación denominado Control Neuro-PID para un sistema de tanques en serie, bajo la norma IEC 61499, ofrece un aporte en el área de investigación para sistemas de control industrial basado en inteligencia artificial mediante el uso y aplicación del estándar IEC 61499 para la automatización industrial, lo cual facilita la portabilidad, configurabilidad e interoperabilidad entre proveedores, así como la independencia de software y hardware, presentando un método eficiente en esfuerzos de ingeniería aplicada al diseño de control PID que se lleva a cabo en sistemas de control, permitiendo regular la sintonización de controles y optimizando el cálculo de valores adecuados de la salida de un sistema, por medio del diseño de redes neuronales con el objetivo de dar autonomía a cada parte constituida de un proceso presente en un control PID ideal, logrando demostrar dentro de los laboratorios de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato de manera práctica y a modo de prueba la capacidad de manejar un sistema de tanques en serie empleando computadoras industriales IPC. Donde la red neuronal será una herramienta eficiente para modelar un controlador PID autoajutable para el proceso de nivel de líquido en tanques, utilizando la tecnología actual que ofrece la Facultad como es el módulo FESTO® MPS-PA.

El desarrollo del proyecto de titulación y su aporte en la investigación se divide en cuatro capítulos que se detalla a continuación.

Capítulo I, se basa en la investigación aplicada desglosando cada tema referente al proyecto recolectando información basado en papers científicos, artículos académicos, investigación bibliográfica, libros, tesis y temas relacionados con el proyecto, así como información recolectada de fuentes bibliográficas que aporten con los requerimientos y especificaciones en el campo de los sistemas de automatización industrial además de los sistemas de control de variables complejas utilizando la norma IEC-61499. De esta manera se estipula el objetivo principal y objetivos específicos que permitan analizar y encontrar una solución al problema que presentan las nuevas exigencias de los sistemas industriales complejos.

Capítulo II, se describe la metodología aplicada en el proyecto de investigación, abarca la parte de métodos y materiales de software y hardware necesarios para su desarrollo

en base a la problemática planteada, definiendo la modalidad de investigación, recolección de información, el análisis y procesamiento de datos y la propuesta de solución para dar paso al desarrollo del trabajo.

Capítulo III, se desarrolla el proyecto investigativo basado en el cumplimiento de cada objetivo planteado, enfocado a las necesidades presentes en el trabajo de investigación como funcionalidades requeridas para satisfacer cada etapa del proyecto de manera óptima, dando cumplimiento al sistema aplicado como apoyo en el área de sistemas industriales en el campo de control, automatización y procesamiento de información mediante el uso de la norma IEC-61499. Se diseña y valida el sistema basándose en simulaciones, diseño, pruebas de funcionamiento de control y resultados del proyecto.

Capítulo IV, se describe las conclusiones y recomendaciones evaluadas del proyecto, basado en los resultado conseguidos y alcanzados por parte de la investigación que demanda el área estudiada.

CAPÍTULO I

MARCO TEÓRICO

1.1 Tema de Investigación

Control Neuro-PID para un sistema de tanques en serie bajo la norma IEC-61499.

1.2 Antecedentes Investigativos

Debido a que se trata de un tema extenso y de interés mundial se lograron encontrar varias investigaciones de las cuales se enuncian las siguientes:

En el año 2021, Daewom Park, Tien Loc Le, Nguyen Vu Quynh, Ngo Kim Long y Sung Kyung Hong publican el artículo científico, “Online Tuning of PID Controller Using a Multilayer Fuzzy Neural Network Design for Quadcopter Attitude Tracking Control”. En este estudio, se presenta un control PID adaptativo que utiliza un diseño de red neuronal difusa multicapa para el control de una cuadricóptero. El controlador consta de una estructura multicapa para mejorar la capacidad de aprendizaje y la flexibilidad de una red neuronal difusa. Las leyes de adaptación para actualizar los parámetros de la red en línea se derivaron utilizando el método de descenso de gradiente. Los resultados del control del cuadricóptero usando el controlador PID-MFNN propuesto, se realizó usando el simulador de robótica Gazebo y ROS el cual utiliza el lenguaje de programación JavaScript. En este estudio, se utilizó el cuadricóptero IRIS de 3D Robotics sin sensores adicionales, que se simularon en el terreno básico como un entorno del mundo real para verificar el sistema de control diseñado. Dando como resultado un método eficaz para obtener ganancias adecuadas para los controladores PID sin necesidad de un modelo de sistema matemático o cálculos complicados. Además, se proporcionó una nueva estructura multicapa para mejorar la capacidad de aprendizaje y la flexibilidad de la FNN utilizada. Se concluye que el diseño puede ser utilizado en cuadricóptero real y se plante como trabajo futuro la aplicación de algoritmos óptimos para lograr parámetros iniciales adecuados [1].

En el año 2020, Johannes Gunther, Elias Reichensdofer, Patrick M.Pilarski. Klaus Diepold publican el artículo científico “Interpretable PID parameter tuning for control engineering using general dynamic neural networks: An extensive comparison”. En

este artículo, se examinó la utilidad de extender los controladores PID con redes neuronales recurrentes, mostrando que los controladores PID GDNN funcionan bien en una variedad de sistemas de control. Para hacerlo, se utilizaron cuatro sistemas de referencia que representan las referencias de ingeniería de control más comunes. Para los experimentos se utilizaron Intel Core i5-4570 en el entrenamiento y en la ejecución de los 4 sistemas se utilizó una Raspberry Pi 3 Model A+. Los sistemas fueron sistemas de tanques, Péndulo Invertido, Sistema LTI con retardo de entrada, Bucle de convección térmica caótica. Para cada sistema se realizó una ejecución de los mismos de 50 veces donde en cada escenario el control fue el mejor. Los 4 sistemas diferentes se realizaron pruebas con y sin ruido del sensor así, como con y sin perturbaciones, lo que da como resultado 16 experimentos en total. Estos escenarios cubren los desafíos más importantes en la ingeniería de control. Por lo tanto, este estudio es único en su extensión, ya que los documentos anteriores solo utilizaron un tipo de sistema de referencia. El enfoque basado en redes neuronales supera a un controlador PID estándar en 15 de 16 escenarios y supera a un controlador basado en modelos en 13 de 16 escenarios. Los resultados muestran el potencial de ampliar los sistemas existentes mediante el aprendizaje automático en general y las redes neuronales en particular. Además, se mantiene el diseño y la integración de la red neuronal simples para permitir una fácil adopción de nuestra técnica. Se realizó un análisis detallado para un escenario representativo, destacando el rendimiento de control superior del enfoque sobre el controlador PID tradicional y el retroceso basado en modelos [2].

En el año 2019, Antonio Marino and Filippo Neri publica el artículo científico ‘PID Tuning with Neural Networks’. En este artículo, se investigó empíricamente los perceptrones multicapa recursivos para sintetizar un controlador PID y sus ganancias mediante la determinación de las acciones proporcional, integral y derivativa. Se reporta sobre la red neuronal recursiva para ajustar el controlador PID en un sistema de prueba desarrollado en Matlab usando Symulink y comparando el rendimiento de la arquitectura con los métodos de Zigler y Nichols. También informa sobre una investigación empírica preliminar de la nueva metodología. En el desarrollo se utilizó un control PID convencional y red neuronal PIDNN que hace uso de tres tipos de neuronas: neurona P, neurona I y neurona D que realizan las tres acciones de control fundamentales. La red está construida con tres niveles. El nivel de entrada está

formado por dos neuronas que toman como entrada la salida de la planta y la entrada de referencia. El segundo nivel está formado por las neuronas P, I y D. Y el tercer nivel combina los tres componentes de la acción de control en la acción de control que se aplicará a la planta. Dando resultados empíricos muestran que el método PIDNN es muy prometedor en términos de un aprendizaje más rápido de las ganancias del controlador con respecto a métodos más tradicionales como ZN y ES [3].

En el año 2018, Ali Zribi, Mohamed Chourou y Mohamed Djemel publican el artículo científico “A New Neural Network Controller Design for Nonlinear Processes”. En este artículo, se propuso un método de ajuste adaptativo del controlador de red neuronal para procesos no lineales el cual consta de una red con tres entradas, diez neuronas ocultas y una sola salida por medio de la herramienta Matlab. Donde se utiliza el método descenso de gradiente para ajustar los parámetros del PID. Los resultados de la simulación muestran que el controlador PIDNN propuesto es significativamente mejor que los otros métodos considerados. El controlador puede lograr mejores resultados de control con menos períodos de muestreo y mejora la estabilidad para condiciones variables del sistema y perturbaciones externas [4].

En el año 2018, Alka Agrawal, Vishal Goyal y Puneet Mishra publican el artículo científico “Adaptive Control of a Nonlinear Surge Tank-Level System Using Neural Network-Based PID Controller”. En este artículo, muestra el diseño un controlador PID de red neuronal basado en la técnica de red neuronal multicapa para controlar el nivel de líquido en un sistema de tanque de compensación no lineal. El controlador Neuronal-PID funciona con un algoritmo de retropropagación y los pesos se actualizan de acuerdo con la regla de aprendizaje basada en el descenso del gradiente el cual se simuló por medio de la herramienta Matlab. Se realizan varios estudios de simulación para comprobar el rendimiento del controlador NN-PID y compararlo con el controlador PID convencional. Al comparar los índices de desempeño del controlador NN-PID y el controlador PID convencional, se encuentra que para el caso nominal el NN-PID tiene una integral de error cuadrático (ISE) de 32.965 que es 97.35% menor que el controlador PID convencional que tiene ISE de 1.246E3. Del mismo modo, al comparar el ISE de ambos controladores durante la perturbación y las variaciones paramétricas, Sobre la base de los resultados comparativos, se puede concluir que el controlador Neuronal-PID propuesto ofrece un mejor rendimiento y se cree que es más

útil para ser utilizado en los procesos industriales que tienen dinámicas no lineales e inciertas [5].

En el año 2018, Geetha G publican el artículo científico “MODELLING AND SIMULATION OF NEURAL NETWORK BASED INTELLIGENT PID CONTROLLER FOR PRESSURE CONTROL USING BACK PROPAGATION ALGORIT”. En este artículo, el control NEURO-PID utilizando un algoritmo back propagation atrás basado en una red neuronal. El modelo de red neuronal fue diseñado para ajustar el parámetro para los controladores PID del modelo de planta. El controlador NEURO-PID tiene la capacidad de adaptarse rápidamente a los cambios como su entrada y también brinda un mejor tiempo de estabilización, menos sobre impulso y desviación mínima en el punto de ajuste en comparación con el controlador PID convencional, para realizar la simulación utilizaron la herramienta de Simulink. El sistema consta de dos tanques cilíndricos en posición horizontal. Uno de los tanques se mantiene a un nivel más alto que el otro. Para el cálculo de la presión de los tanques utilizan un transmisor de presión mide la presión del tanque y da una señal de corriente de salida 4-20 mA que se convertirá a 0-5 voltios usando un convertidor de corriente a voltaje. El sistema de control calcula el error entre el punto de ajuste y la variable medida y el controlador calcula la acción de control que se le da a la planta a través del DAC. La salida del DAC está en el rango de 0-5 V y se le da al convertidor de voltaje a corriente que da una salida en 4-20 mA. La señal de corriente se envía al convertidor de presión, que pasa al convertidor de corriente a presión, que proporciona una salida de 3 a 15 psi de presión. Para el control NUERO-PID consta por el controlador PID convencional y la red neuronal back propagation donde la red consta de tres capas en la entrada ingresan los parámetros rink, yout y error y en la salida de la red se devuelve los parámetros kp, ki y kd. Dando como resultado de la simulación del sistema de control de presión en un tanque mediante el uso de un controlador PID basado en la red neuronal BP muestran que puede controlar mejor la robustez y también reducir el tiempo de asentamiento [6].

1.3 Contextualización del Problema

La arquitectura actual del sistema de automatización industrial ha hecho un trabajo al hacer avanzar la industria hasta donde está hoy, pero la tecnología patentada, cerrada

y centralizada hace difícil que se puedan optar tecnologías avanzadas de manera rápida y eficiente. Por lo cual la industria está pasando de sistemas centralizados a un paradigma más distribuido. Los sistemas actuales con una inteligencia central que controla todo se transforman en sistema distribuidos. En los sistemas distribuidos, las partes individuales tienen inteligencia y pueden comunicarse entre sí, por lo que el sistema actúa como un todo. La norma IEC61499 define un lenguaje de modelado de dominio para desarrollar soluciones de control distribuidas [7].

El no aplicar estándares abiertos de automatización industrial es costoso en el mundo actual. Hoy en día, los fabricantes industriales están adentrados en sistemas abiertos, pero tienen que adoptar un gran cambio con implementaciones con código y portabilidad de funciones, para que sus usuarios finales no tengan que soportar gastos innecesarios de ingeniería y demoras en la implementación de diseños de plantas, lo que provoca pérdidas de tiempo, eficiencia para ambas partes [7].

La adopción del estándar IEC61499 para la automatización industrial puede liberar los esfuerzos de ingeniería y permitir una gestión fácilmente lo que abre el mundo de la automatización a las infinitas posibilidades de la Industria 4.0. Donde para lograr los beneficios requiere que la industria aproveche los conceptos, la tecnología y la arquitectura de la Industria 4.0 e IoT con estándares abiertos [8].

En conclusión, la industria ecuatoriana requiere la aplicación de nuevas tecnologías como la inteligencia artificial conjuntamente con estándares abiertos que permiten una mayor adaptabilidad en procesos, permitiendo a la industria un mejoramiento constante donde, se logrará una optimización de materias primas, ahorro energético y una alta calidad de los productos.

1.4 Fundamentación Teórica

1.4.1 Norma IEC-61499

La norma o estándar IEC-61499 es una arquitectura enfocada en el desarrollo de aplicaciones de control industrial y de ingeniería en lógica descentralizada que se compone con el manejo de bloques funcionales (FB), considerados como la base de los modelos estructurales que se componen de entradas, salidas y características de función interna, de tal manera que se comporta como los diagramas clásicos

estructurando y programando las entradas con sus definiciones para obtener las salidas respectivas, para modelar aplicaciones complejas, el estándar propone el conocimiento de ingeniería en el desarrollo de nuevos métodos en diseño de bloques y modelos. Otra ventaja que ofrece la norma es el método en que ejecuta los bloques funcionales, en el cual permanecen en estado de reposo hasta recibir una señal de entrada para su posterior ejecución aprovechando de mejor manera la portabilidad, y reutilización de los bloques funcionales, que se ejecutan uno a uno cada vez que se requiere solventar una necesidad, pudiendo definirlo como una unidad que no depende del resto siendo solamente libre [9].

1.4.2 Aspectos de la Norma IEC-61499

El objetivo principal del estándar es conseguir un control distribuido en cada parte del proceso de un sistema que se adecue a nuevos componentes y tecnologías, como sistemas embebidos computadoras industriales y redes avanzadas, que permitan hacer los sistemas más flexibles mediante la descentralización de la lógica de control e implementación de sistemas distribuidos de inteligencia artificial, aprovechando la reutilización de código de datos y poder diseñar un mismo software para dos máquinas con diferentes especificaciones de fabricación, por ello el estándar IEC-61499 se basa en el estudio de tres aspectos clave a considerar [10].



Figura 1. Aspectos de la Norma IEC-61499

En la siguiente figura 2, se visualizan los aspectos representados por la norma IEC-61499.

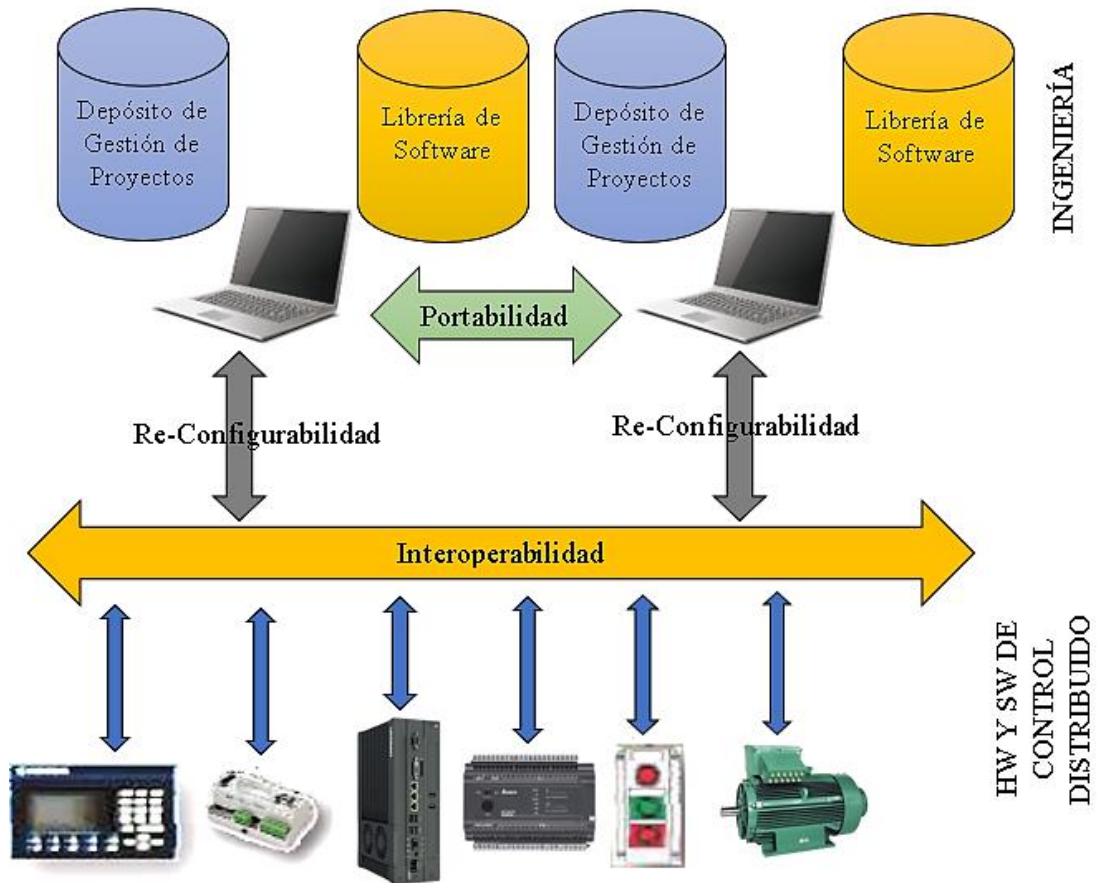


Figura 2. Aspectos representados en la norma IEC-61499.

1.4.3 La Norma IEC-499 y su estructura

La norma o estándar se establece en cuatro partes de distribución, cada una tiene sus objetivos establecidos para ser cumplidos y cada parte es independiente, a continuación, se muestran estas partes en la tabla 1.

Tabla 1. Estructura de la Norma

Estructura	Objetivos
Parte 1	<ul style="list-style-type: none"> • Define la arquitectura para los bloques funcionales, como son el en desarrollo, reutilización e implementación. • Define los bloques funcionales básico, compuesto entre otros, como aplicaciones señal, datos, interfaz, etc.
Parte 2	<ul style="list-style-type: none"> • Define los requisitos necesarios que deben cumplir los bloques funcionales en la parte de desarrollo para estructurar de manera ideal la arquitectura propuesta, estos requisitos no abarcan os aspectos referentes a la norma presentes en la figura 1. • Define los esquemas XML entre las herramientas de desarrollo y los esquemas o ficheros
Parte 3	<ul style="list-style-type: none"> • Instructivo presentado de manera anímica en el cual se muestran las funcionalidades de la norma y sus aplicaciones en un vasto conjunto de dominios.
Parte 4	<ul style="list-style-type: none"> • Define perfiles que permitan complicar de forma ordenada y objetiva los aspectos de la norma IEC-61499, dando cumplimiento a estos requisitos en los dispositivos y herramientas usadas para el desarrollo.

Elaborado por: El investigador [11].

1.4.4 Tipos de Modelos de la Norma IEC-61499

Para comprender los conceptos definidos de esta norma, se desarrolla la parte 1 definida anteriormente en cual abarca las bases de la misma, enfocando el estudio referente a la investigación del proyecto como son los bloques funcionales.

1.4.4.1 Modelo de bloque funcional (FB)

El bloque funcional se define como una estructura independiente frente de otros bloques, que posee sus respectivas funcionalidades, así como variables de ejecución, actuando como una unidad computacional con sus propios algoritmos los cuales pueden ser: IL, LT, ST, SFC, FBD, así como otros lenguajes de alto nivel como: C,

C++, Java y Delphi, cada bloque se representa mediante una interfaz como se observa en la figura 3.

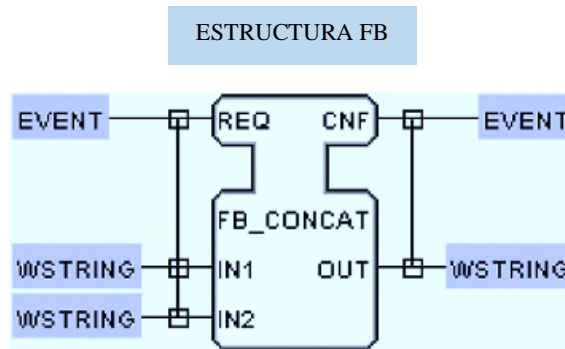


Figura 3. Modelo de bloque (FB)

En la figura 3, anterior se puede evidenciar e identificar la estructura de un bloque funcional y como se conforma en sus partes, en la parte superior se encuentran los eventos donde se definen las señales y datos de entrada, es decir a la izquierda las señales de entrada y a la derecha las señales de salida configurado dependiendo de las necesidades requeridas.

En la parte superior y fuera del bloque se nombra del mismo, que sirve para identificar cada bloque realizado dentro del esquema y en la parte central se denota el tipo de bloque que es y la función que realiza.

En la parte inferior del bloque, en la parte izquierda se encuentran los datos de entrada con el tipo de señal correspondiente a las salidas que se encuentran en el lado derecho del bloque. Y finalmente se define el conector que une varias señales con sus respectivos datos, este se representa con cuadros y una línea entrecruzada entre las señales y los datos de tal manera que se asocien entre ellos para y exista a interacción o actualización de los registros al momento de recibir o enviar una señal [12].

La norma IEC-61499, presenta y define una arquitectura que comprende los modelos de control de un sistema aportando en el área de ingeniería industrial y de automatización el entendimiento de organización del sistema de control distribuido y sus componentes de forma fácil y sencilla siendo modelos genéricos es decir que no depende del dominio y son flexibles ante plataformas de hardware comprensibles entre sí. De esta manera el elemento base son los bloques funcionales que se conforman de una estructura elemental donde la cabeza permite la conexión y flujo de eventos, es

decir acepta eventos de entrada y genera eventos de salida, y el cuerpo permita el flujo de datos recibiendo datos de entrada y generando datos de salida, de esta manera se define tres FBs [11], como se observa en la siguiente figura 4.

- **Bloque Funcional Básico.** - Es el bloque mínimo elemental de programación que se conforma por: ECC y algoritmos.
- **Bloque Funcional Compuesto.** - Es la asociación de varios FBs interconectados en forma de red de instancias.
- **Bloque Funcional Interfaz de Servicio.** - Proporciona servicios que permitan la interacción existente entre aplicaciones y recursos.

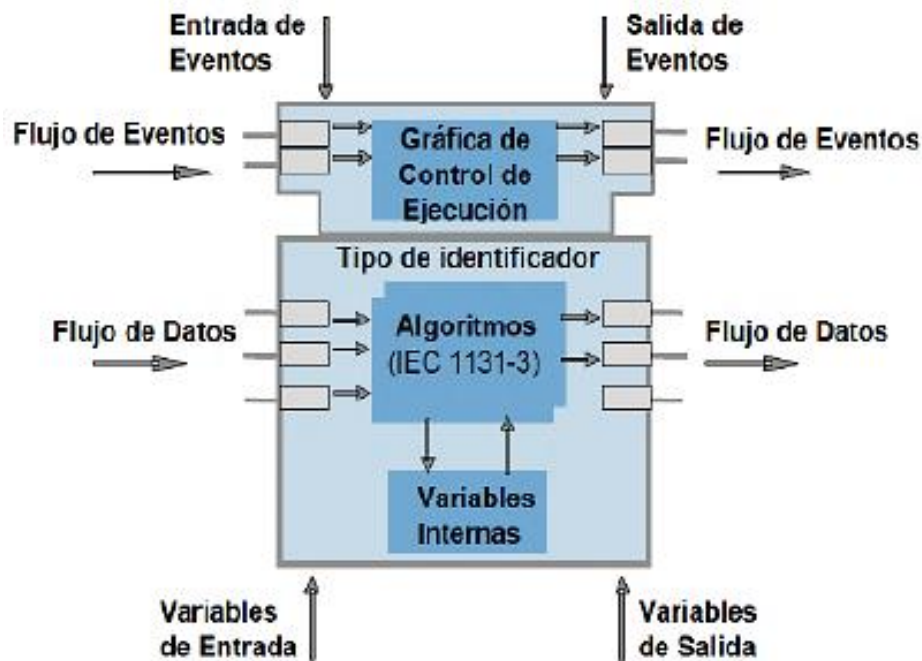


Figura 4. Estructura de bloque de función [11].

1.4.4.2 Modelo de bloque Funcional de Interfaz de Servicio

Los bloques funcionales de interfaz de servicio SIFB, son la herramienta esencial para la comunicación entre aplicaciones y herramientas de desarrollo, puesto que brindan servicios a cada bloque para tener comunicación con el mundo exterior, es decir brindan los servicios requeridos de acuerdo a las necesidades de cada bloque, aplicado al proyecto investigativo permite dar servicios de escritura y lectura de entradas y salidas y el uso adecuado entre las aplicaciones y recursos, además de efectuar la

interacción entre software y hardware a través de una interfaz, en la figura 5, se representa la estructura de un bloque SIFB [13].

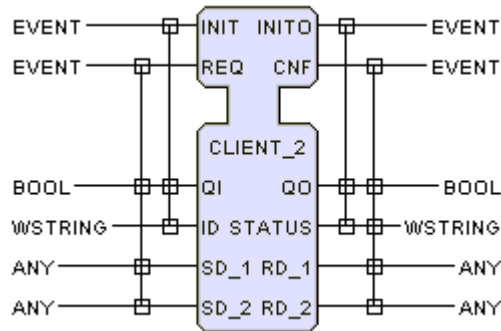


Figura 5. Estructura de un bloque funcional de interfaz de servicio SIFB [13].

1.4.4.2.1 Características del bloque funcional de interfaz SIFB

Las especificaciones del bloque funcional de interfaz SIFB es similar con los bloques funcionales FB, sin embargo, en algunas entradas y salidas la semántica tiene sus propias especificaciones y decisiones que se definen por notación grafica especializada, en las tabla 2 y tabla 3, se describen los eventos y datos de entradas y salidas del bloque funcional de interfaz de servicio bajo la norma IEC-61499 [14].

Tabla 2. Eventos y datos de entrada del bloque funcional de interfaz de servicio bajo la norma IEC-61499.

Entradas	
Eventos	Datos
<ul style="list-style-type: none"> • Init. - Es la entrada de inicialización del servicio que esta proporcionado por los bloques funcionales fb. 	<ul style="list-style-type: none"> • Qi: Bool.- Representa la asignación calificativa de un dato inicializado con su servicio, es decir, si un evento es verdadero, sobre la inicialización el proceso del servicio comienza, y si es falso, la finalización del servicio es solicitado.
<ul style="list-style-type: none"> • Req. - Se proporciona por los bloques funcionales y debe asignada a una petición para inicializar el servicio. 	<ul style="list-style-type: none"> • Any.- Existen varios parámetros que van a la par con los servicios, que pertenecen a un tipo de dato definido, el bloque funcional FB asigna el tipo de dato y el valor de inicialización.

Elaborado por: El investigador [14].

Tabla 3. Eventos y datos de salida del bloque funcional de interfaz de servicio bajo la norma IEC-61499 [14].

Salidas	
Eventos	Datos
<ul style="list-style-type: none"> • InitO. - Es la salida que indica la finalización del servicio que esta proporcionado por los bloques funcionales FB por el proceso de un servicio iniciado. 	<ul style="list-style-type: none"> • Qo: Bool.- Representa la asignación calificativa de un dato que finaliza con su servicio, es decir, si un evento es verdadero, sobre la inicialización, el proceso del servicio inicia, si es falso, la inicialización del servicio no es exitosa.
<ul style="list-style-type: none"> • Cnf. - Se proporciona por los bloques funcionales y debe asignada a una petición para la salida del servicio. 	<ul style="list-style-type: none"> • Any Status.- Debe ser apropiada para corroborar el estado del servicio cuando ocurre una verificación de salida.

Elaborado por: El investigador [14].

Se muestra en la figura 6, la inicialización de la aplicación y de recursos de eventos de entradas y salidas en los bloques funcionales.

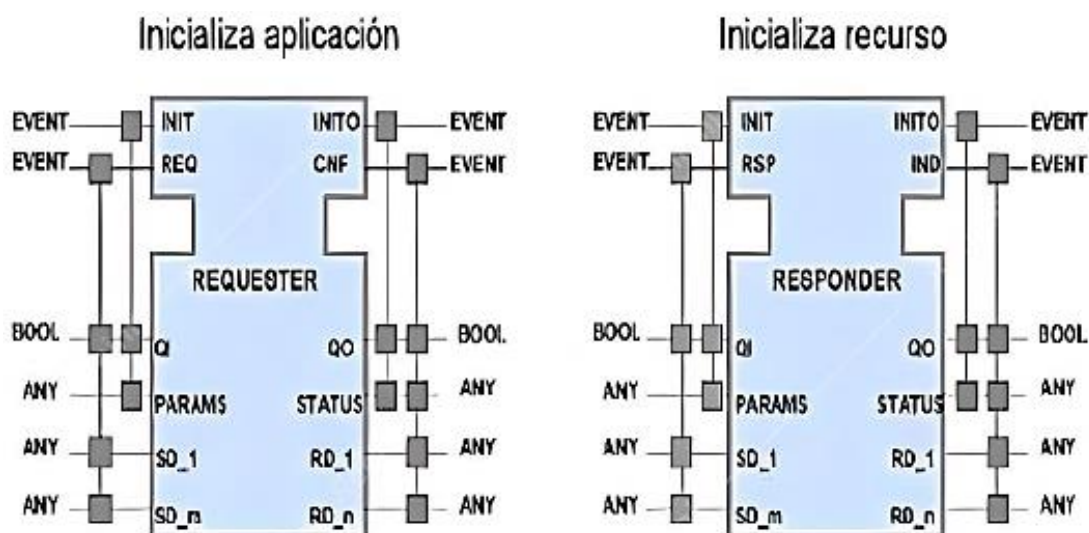


Figura 6. Bloque funcional de interfaz SIFB Solicitante y Responder [14].

1. **Solicitante (REQUESTER).** - Permite la ejecución de un evento para la interacción de inicio de una aplicación o una acción como enviar un mensaje,

que se activa cuando recibe una entrada de evento funcionando como un bloque funcional básico.

2. **Respondedor (RESPONDER).** - Permite la ejecución de un recurso para la interacción de inicio de un servicio en reposo o una acción proporcionando una salida de evento, o interrumpir la ejecución de acción de cualquier bloque funcional.

Para realizar la comunicación bajo la Norma IEC-61499 se utiliza los bloques funcionales SIFBs, los paradigmas que usan para la comunicación son publicador/subscriptor y cliente/servidor.

Correspondientemente de ello Publish-Subscribe recibe la información y la transmite en un solo sentido de dirección como se observa en la figura 7, se lleva a cabo por el uso de protocolo de Datagram del usuario UDP, que guía la información en sentido unidireccional. Se envía los datos para ser publicados de manera que cada dispositivo que escuche los datos enviados se subscriban en una misma transmisión o canal, de esta manera los dispositivos compatibles entre especificaciones de fábrica diferentes pueden hacer uso de un parámetro específico como el usuario lo requiera, sin afectar el manejo de control o rendimiento del sistema.

Una ventaja a destacar con este tipo de comunicaciones es la determinación con que realizan el proceso de comunicar pues se validan con el uso de un programa predeterminado lo cual permite asegurar la transmisión y recepción de los datos o la información de manera precisa y cuando se requiera.

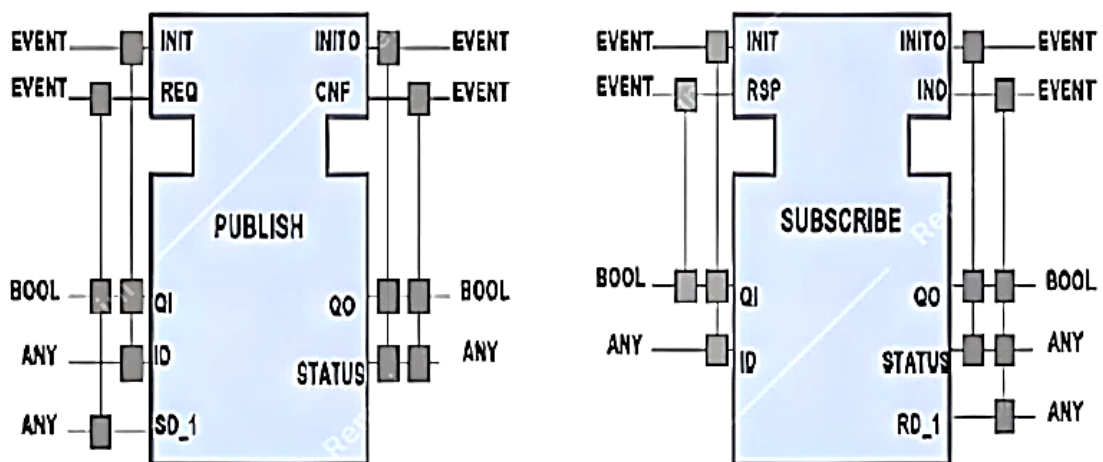


Figura 7. Paradigmas de comunicación bloques de Publish y Subscribe [14].

Mediante CLIENT-SERVER la comunicación es bidireccional es decir la información se transmite en dos direcciones como se observa en la figura 8, basado en el protocolo de control de transmisión TCP, para transferir los datos.

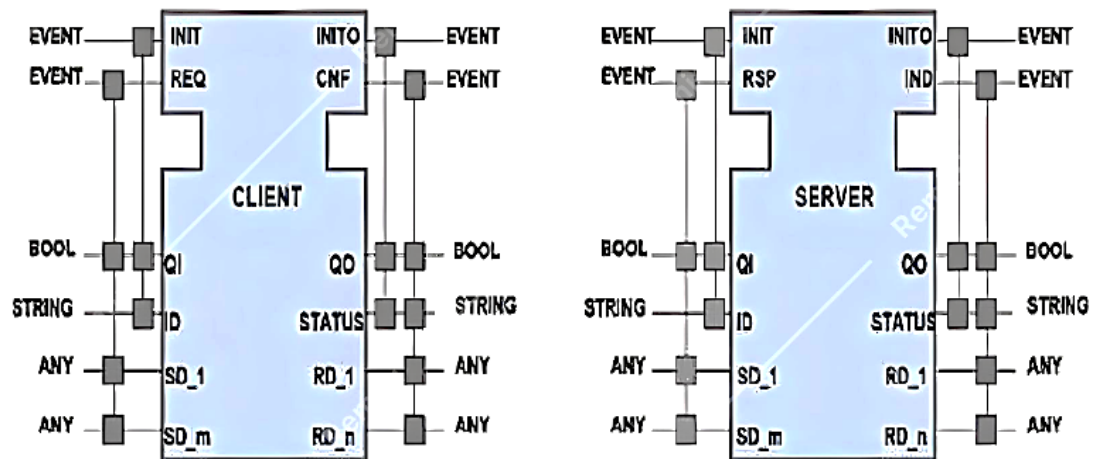


Figura 8. Interfaz de bloques de comunicación de Client-Server [14].

1.4.4.3 Modelo de bloque de recurso

Este modelo permite actuar como una interfaz de comunicación como un recurso para los sistemas en los procesos de cada dispositivo, es decir las necesidades que se requiere en los subsistemas como son entras/salidas. De esta manera se consigue el funcionamiento adecuado del control de un sistema por tanto cada bloque derivado en aplicaciones se modela independientemente para utilizarlo en donde vaya a ser ejecutado, es decir se asigna y configuran los bloques para ser usados en cada parte del dispositivo de forma independiente brindando flexibilidad al sistema de control, en la figura 9, se visualiza la estructura del modelo de recurso bajo la norma IEC-61499.

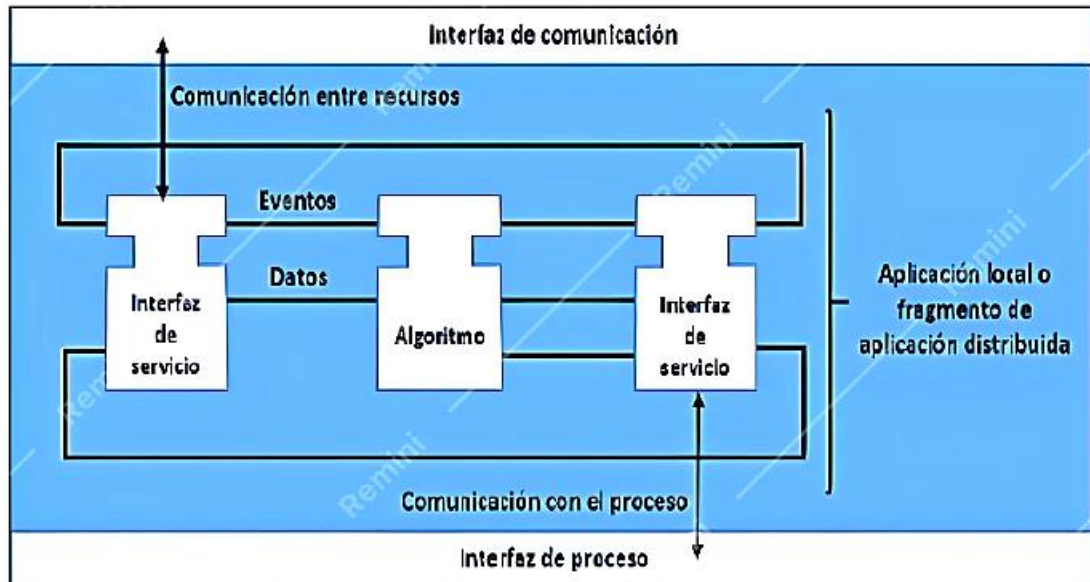


Figura 9. Interfaz del modelo de bloque de recurso [14].

1.4.4.4 Modelo de bloque del dispositivo

Este modelo se representa como la interfaz física que conforma el grupo de FBs o bloques funcionales a ser efectuados, actuando como una PC, de esta manera la comunicación puede realizarse entre dispositivos similares o entre los periféricos por medio de interfaces y a través de una interfaz de proceso poder comunicarse con FBs a través de su red, en la figura 10 se visualiza la estructura del modelo del dispositivo [15].

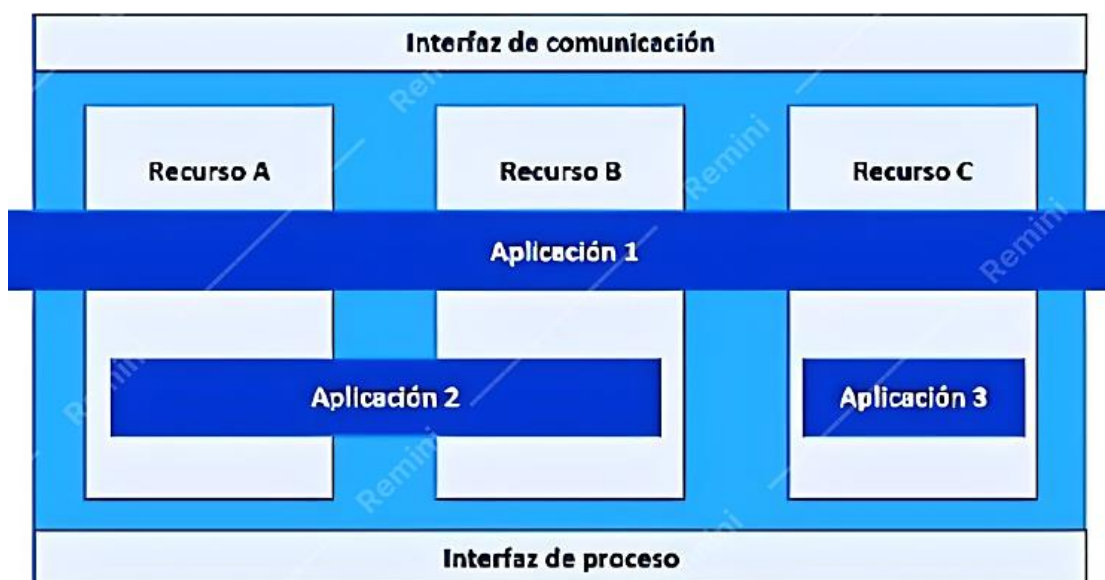


Figura 10. Estructura del modelo del dispositivo [15].

1.4.4.5 Modelo de bloque del Sistema

El modelo de sistema es la arquitectura de alto nivel, puesto que abarca todos los bloques de aplicaciones y dispositivos capaces de comunicarse es decir toda la interacción que surge a raíz de estos dos grupos, permite tener las aplicaciones distribuidas en todo el sistema a pesar de que sean unitarias, basándose en la norma IEC-61499 [16], como se muestra en la figura 11.

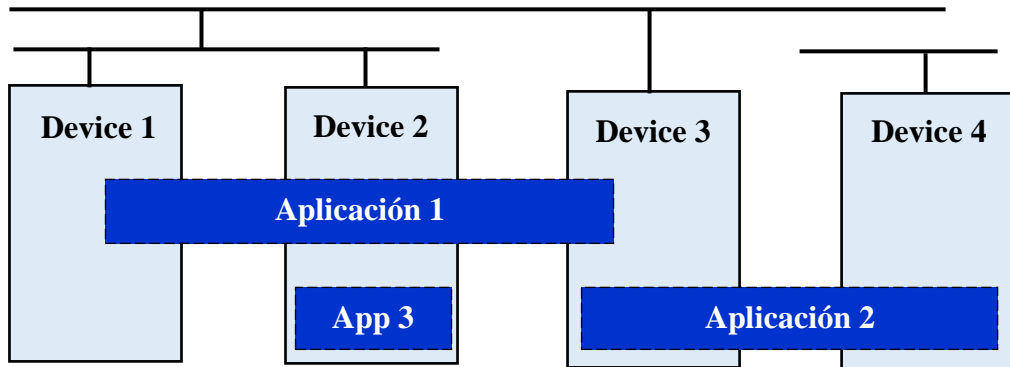


Figura 11. Interfaz de modelo del sistema bajo la norma IEC-61499 [16]

1.4.4.6 Modelo de Bloque de aplicación y sub-aplicación

El modelo de aplicación se basa en el conjunto de bloques funcionales FB, que van definidos por eventos y datos modelados, que describen el funcionamiento de la automatización sin importar donde se ejecute. Para una correcta ejecución la aplicación es distribuida entre los recursos de esta manera también se puede definir sub-aplicaciones que se comportan como bloques funcionales FBs, agrupándose y relacionándose definiendo entradas/salidas y eventos, en la siguiente figura 12, se muestra la estructura de un modelo de bloque de aplicación y sub-aplicación [16].

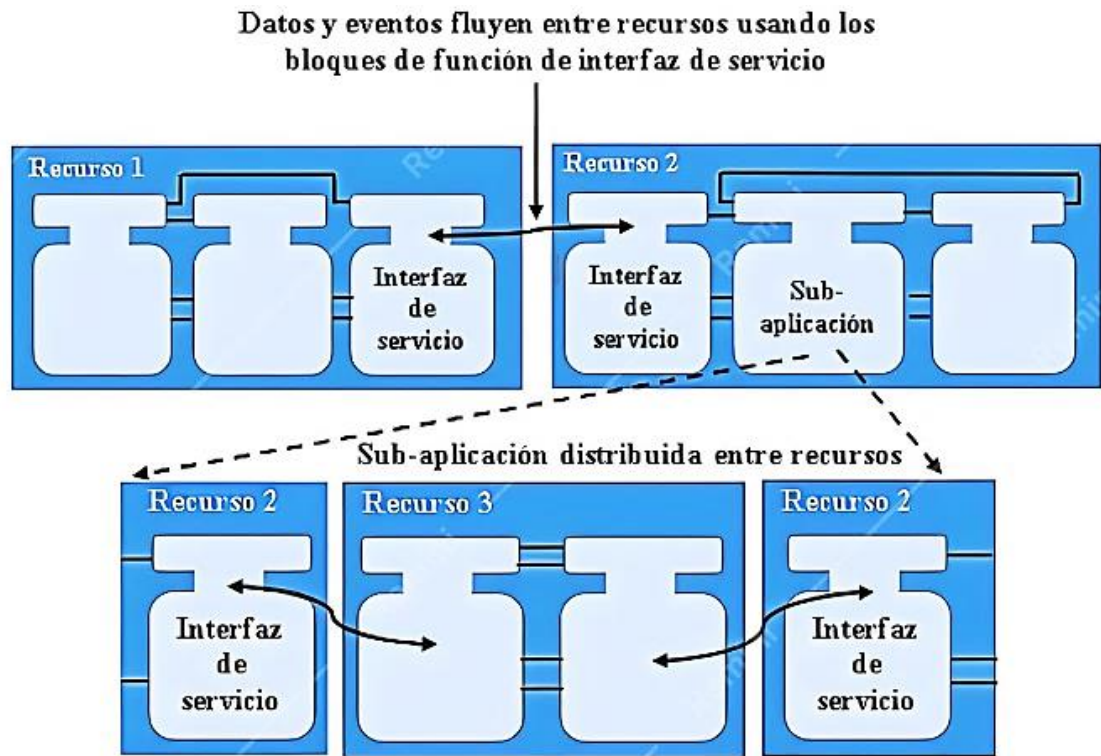


Figura 12. Estructura de un modelo de aplicación y sub-aplicación [16].

1.4.4.7 Herramientas de desarrollo

Las herramientas de desarrollo de los modelos deben cumplir con determinadas características basadas en la interoperabilidad punto importante que demanda la Norma IEC-61499, a continuación, se describen las cuatro herramientas de desarrollo, su función y el grado que presenta de interoperabilidad.

1.4.4.7.1 Herramienta 4DIAC-IDE

Es un entorno compatible con Ingeniería por su alta flexibilidad, sus requisitos cumplen con la norma IEC-61499, permitiendo el control distribuido en un sistema. Se puede modelar bloques funcionales con aplicaciones portables a dispositivos que son dirigidos y efectuados por los medios definidos del estándar, su configuración se basa por la capacidad de hardware que posee, permitiendo modelar el control de un sistema por medio de redes interconectadas, y basado en el uso del entorno Eclipse, y aprovecha los recursos para distribuir las aplicaciones de forma gráfica, también tiene actualizaciones que permiten hacer más fácil el uso de sus herramientas y funciones

básicas como son lectura de valores o de variables, en la figura 13, se observa el entorno de la herramienta 4DIC-IDE basada en la Norma IEC-61499 [17].

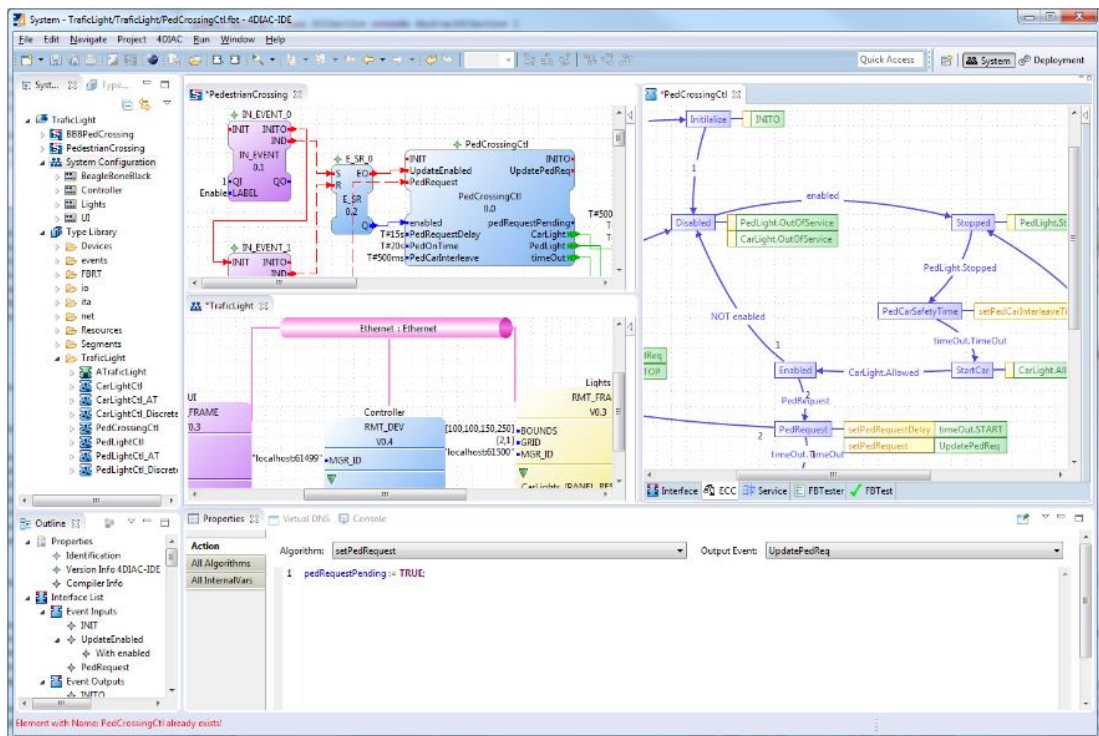


Figura 13. Entorno de 4DIC-IDE basada en la Norma IEC-61499[17].

1.4.4.8 Plataformas de cumplimiento o ejecución

Mediante el estudio de la Norma IEC-61499 previamente investigada, proponer una nueva arquitectura en el diseño de modelos que apoyen a la ingeniería automatizada e industrial en el manejo ideal de control distribuido, aclarando que no es una propuesta de implementación como tal, es decir permite el desarrollo independiente de la arquitectura de un modelo pero se requiere de una plataforma que accione dicho modelo, lo distribuya, gestione e interprete dando órdenes de accionamiento a los bloques funcionales FB, y asignando tareas de acuerdo a las necesidades requeridas.

Esta plataforma se encarga de administrar todo lo que abarca el hardware de un dispositivo que contiene el modelo diseñado, gestiona la memoria, inspecciona las entradas y salidas de cada bloque, así también evalúa el área de comunicaciones, deja a punto la ejecución de un sistema para luego dar paso al accionamiento del programa.

Para llevar a cabo la ejecución de un modelo en el sistema la plataforma debe estudiar y conocer cada uno de los bloques funcionales FBs, de los cuales se conforma en el programa, es decir compilar y ejecutar la plataforma de modo que sea compatible y adaptada a los tipos de bloques que se están usando para luego accionar la plataforma que contiene el modelo diseñado y que corra de manera exitosa.

1.4.8.8.1 Tiempo de ejecución del bloque funcional (FBRT)

Es uno de los primeros métodos de ejecución que se desarrollaron para realizar pruebas utilizando la Norma IEC-61499, se realizaron dos versiones tanto para sistemas embebidos como no embebidos de acuerdo a las necesidades que se presentaban, la primera por los dispositivos que se desarrollaban y la segunda por llegar a tener un sistema en tiempo real utilizando java, que permitió expandir su aplicación a ser ejecutado en cualquier dispositivo. Además de su flexibilidad con otras plataformas se sigue utilizando para visualizar los resultados y simular aplicaciones, para este proyecto también es compatible con la herramienta 4DIAC-IDE.

1.4.8.8.2 FORTE

La plataforma FORTE se basa en código abierto y su lenguaje de programación es C++, es versátil y fue implementado por el desarrollador de la herramienta 4DIAC, se utiliza tanto para sistemas embebidos como para sistemas en tiempo real por sus ventajas de ejecución y gestión, es una plataforma robusta y fuerte además de que llega a ser independiente y se ejecuta en cualquier plataforma en la que se haga uso de esta manera se puede implementar con hardware y sistemas operativos diferentes como son POSIX, Windows 32, eCos, para el presente desarrollo del proyecto se combina perfectamente FORTE con la herramienta de desarrollo 4DIAC, en la figura 14, se observa la interfaz de ejecución de FORTE y 4DIAC bajo la Norma IEC-61499 [18].

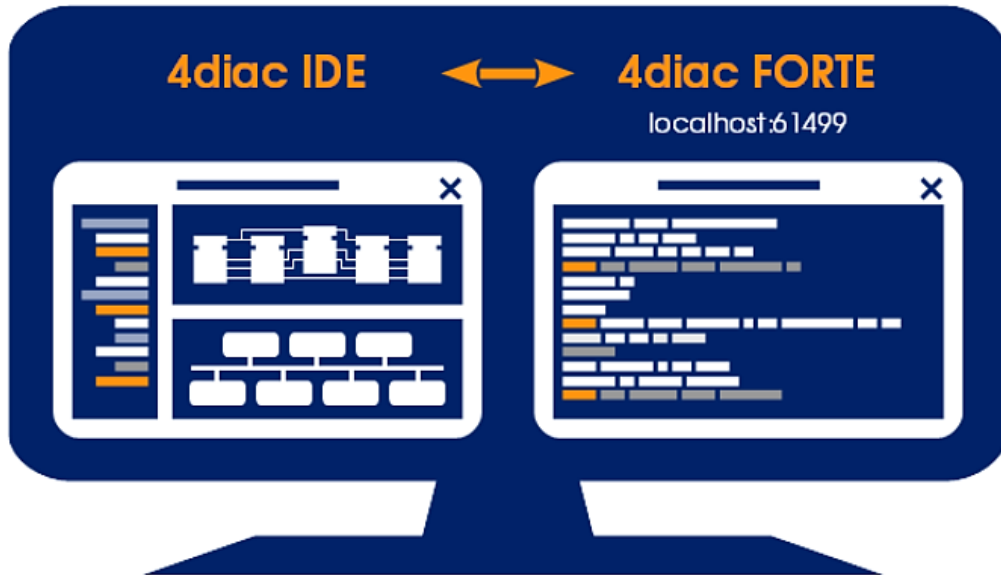


Figura 14. Plataforma FORTE y 4DIAC bajo la Norma IEC-61499 [18].

1.4.8.9 Redes Neuronales Artificiales.

Estas redes son un conjunto de neuronas que se comunican por estímulos algunas se activan otras se inhiben de manera que actúan recibiendo y transmitiendo información entre cada neurona esto incrementa a relación entre ellas y establecer niveles de memoria lo cual permite aprender y el cerebro es entrenado de acuerdo a un conjunto de datos repetitivos que producen estímulos. De acuerdo a este comportamiento se puede replicar el proceso en una red neuronal artificial interpretándola como un modelo operacional de cálculo y matemática que imite el comportamiento de una neuronal en el cerebro resaltando que la mayor ventaja de una red neuronal artificial es la capacidad que tiene para recolectar datos y aprender funciones o información de variables y una de las desventajas es la reciente implementación de esta tecnología [19].

La red neuronal artificial es considerada como un sistema computacional que interactúan con herramientas compuestas y elementos que se entrelazan para dar funciones como procesamiento de señal o interpretación de información, basados en el estado en el cual se encuentren dando respuestas a las entradas requeridas, permitiendo asemejarse a las características propias de las neuronas humanas y brindando la capacidad de aprender, memorizar y retener datos o información a través del diseño de un modelo matemático requerido de acuerdo a las necesidades presentes

en sistemas industriales, dado que al interactuar en gran número pueden llegar a resolver problemas tan sencillos o muy complejos de acuerdo a las necesidades que se requieran y la manera en cómo se entrenen. [20].

1.4.8.9.1 Funcionamiento de la Red Neuronal

La red neuronal funciona interpretando los valores de las entradas en los módulos e interconectando cada uno de ellos para llevar a la salida, dado la capacidad de memorizar eventos y reconocer problemas complejos con variables diferentes, en la figura 15, se observan el esquema básico de una red neuronal [21].

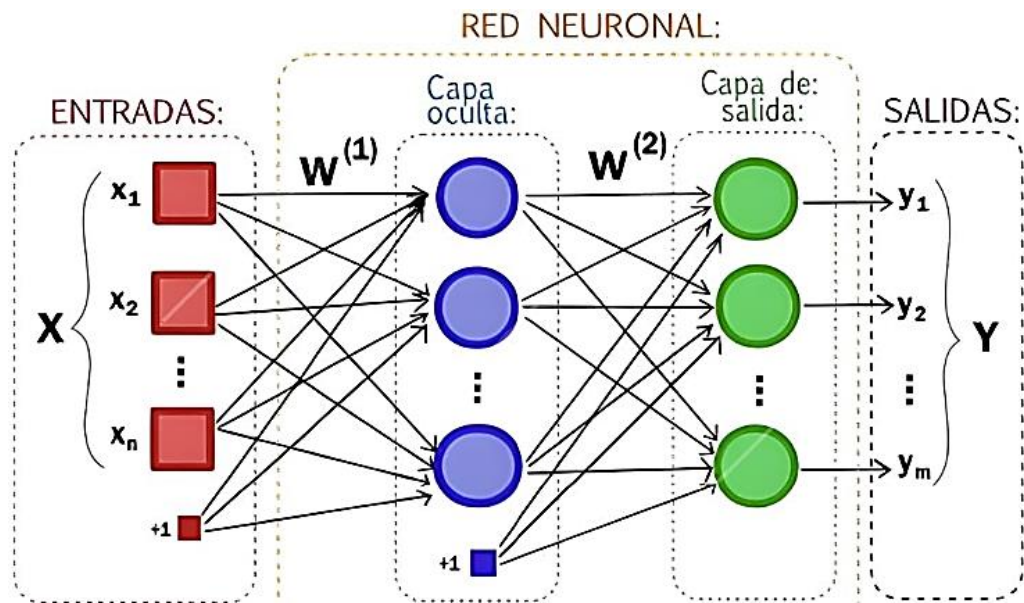


Figura 15. Esquema de las redes neuronales [21].

Se observan las entradas que van a la capa de entrada las cuales se distribuyen en la capa oculta para finalmente dirigirse a la capa de salida, la cual se obtiene aplicando las funciones de propagación, transferencia y activación.

- **Función de Propagación.** - Se considera como el sumatorio de los valores obtenidos que a su vez se multiplica por las conexiones de cada peso y se considera el valor de sesgo.
- **Función de Activación.** – Se comporta como un interruptor, en el cual cuando se requiere este activa o desactiva la salida de una neurona.
- **Función de Transferencia.** - Es la encargada de representar los resultados de salida de una función de propagación a manera de acotamiento existen diversas

funciones de transferencia que se pueden utilizar de acuerdo a las necesidades en las cuales se desee representar los resultados, en la figura 16, se observan algunas de ellas [22].

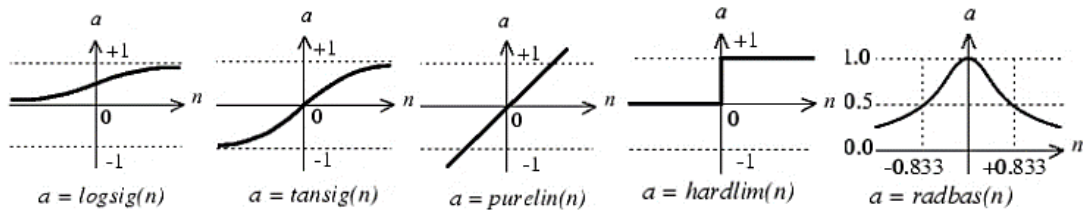


Figura 16. Tipos de funciones de transferencia [22].

1.4.8.9.2 Clasificación de las redes neuronales

En el esquema de la figura 17, se observa la clasificación de las redes neuronales tomando en cuenta los criterios más importantes a considerar.

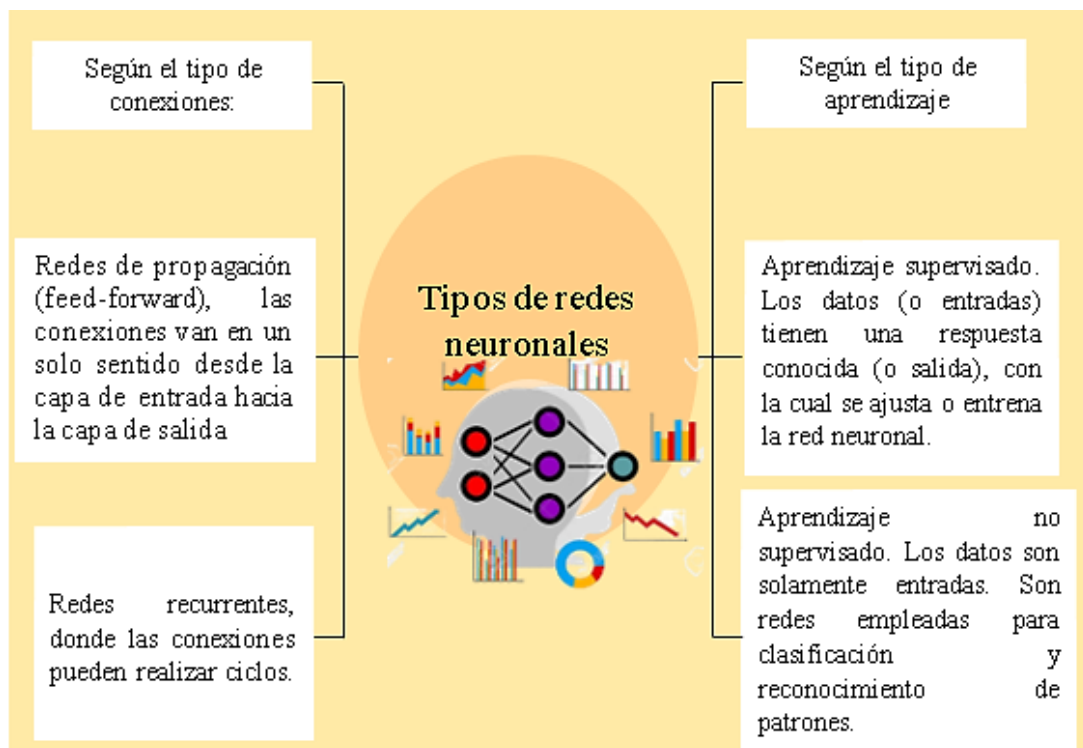


Figura 17. Clasificación de redes neuronales.

Elaborado por: El investigador a partir del artículo: Introducción a las redes neuronales artificiales. Sistemas inteligentes de ingeniería civil [23].

1.4.8.10 Controlador PID

Actualmente, el controlador PID es ampliamente utilizado en la investigación y las industrias. Consiste en un tipo de acciones de control proporcional, integral y derivativo.

Estos tipos de controladores se llaman controladores convencionales y ampliamente utilizados en los sistemas de control industriales [24]. El controlador PID se puede expresar de la siguiente manera:

$$K(s) = k_p + \frac{k_i}{s} + \frac{k_d s}{(1 + \alpha s)} \quad K(s) = k_p + \left(1 + \frac{k_d}{\tau_i s} + \frac{\tau_d s}{1 + \left(\frac{\tau_d}{N}\right) s}\right)$$

Donde k_p , k_i , k_d son; ganancia proporcional, integral y derivativa del controlador, respectivamente; $\tau_i = K_p/K_i$ es la constante de tiempo integración; $\tau_d = K_d/K_p$ es la constante de tiempo derivativo; $\alpha = \tau_d/N$ es un parámetro de diseño [25]. La función de transferencia del controlador PID es:

$$K(s) = k_p + \frac{k_d}{\alpha} + \frac{\left(k_i - \frac{k_d}{\alpha^2}\right) s + \frac{k_i}{\alpha}}{s^2 + \left(\frac{1}{\alpha}\right) s}$$

1.4.8.11 Control PID por medio de redes neuronales.

El control PID es considerado como parte fundamental en cualquier proceso industrializado, pues el objetivo principal siempre es mantener las variables de punto de control estables o en el valor deseado, intervenir en los procesos y poder llevar un mejor control en el momento deseado es lo primordial, pero dado que no siempre se pueden obtener valores exactos de un sistema se requiere del uso de diferentes herramientas de desarrollo que haga posible el control [25].

Las redes neuronales artificiales permiten satisfacer las necesidades de control, puesto que no requieren conocer de manera directa los componentes de un sistema ni sus comportamientos, lo único que requieren son los valores, datos e información en ciertas cantidades para aprender y memorizar datos predecibles y el comportamiento

deseado, este proyecto permite demostrar que las redes neuronales ayudan en mejorar el comportamiento de control PID de flujo de nivel de taques.

1.4.8.12 Entrenamiento de una red neuronal

Entrenar una red neuronal implica proporcionarle datos de entrenamiento y ajustar los pesos y los sesgos de la red para que pueda aprender a realizar una tarea específica. El proceso de entrenamiento se realiza a través de la propagación hacia adelante y hacia atrás de la información a través de la red, que se conoce como el algoritmo de retropropagación del error, los pasos generales para entrenar una red neuronal son:

1. **Selecciona una arquitectura de red adecuada:** La elección de la arquitectura de la red neuronal depende de la tarea que se desee resolver. Una red neuronal típica tiene una o varias capas ocultas que ayudan a procesar la información.
2. **Prepara los datos de entrenamiento:** Para entrenar una red neuronal, necesitas un conjunto de datos de entrenamiento que contenga entradas y salidas correspondientes. Las entradas son los datos que se introducen en la red, y las salidas son los resultados que se espera que la red produzca para cada entrada.
3. **Inicializa los pesos y sesgos de la red:** Los pesos y sesgos de la red se inicializan aleatoriamente en un principio. La red se ejecuta en los datos de entrada y se evalúa su rendimiento.
4. **Propagación hacia adelante:** Durante la propagación hacia adelante, los datos de entrada se procesan a través de la red y se calculan las salidas correspondientes.
5. **Cálculo del error:** Después de la propagación hacia adelante, se calcula el error de la salida de la red en comparación con la salida esperada.
6. **Retropropagación del error:** La retropropagación del error se utiliza para ajustar los pesos y sesgos de la red. El objetivo es minimizar el error en la salida de la red. Este proceso se realiza mediante la aplicación de la regla de la cadena para calcular las derivadas parciales del error con respecto a los pesos y sesgos de cada capa en la red.
7. **Actualización de los pesos y sesgos:** Una vez que se han calculado las derivadas parciales del error, se actualizan los pesos y sesgos de la red para minimizar el error.

8. Repetición de los pasos 4-7: Se repiten los pasos 4 a 7 para todo el conjunto de datos de entrenamiento.
9. Validación y pruebas: Una vez que la red ha sido entrenada, se prueba su rendimiento con un conjunto de datos de validación y pruebas diferentes a los utilizados para el entrenamiento.
10. Ajuste de hiperparámetros: En función del rendimiento de la red en los datos de validación y pruebas, se pueden ajustar los hiperparámetros de la red, como la tasa de aprendizaje, el número de capas ocultas, el número de neuronas por capa, etc.
11. Despliegue: Finalmente, después de la validación y pruebas, la red puede ser desplegada para su uso en producción.

Cabe destacar que el entrenamiento de una red neuronal es un proceso iterativo y que puede requerir una gran cantidad de datos de entrenamiento y potencia de cómputo para obtener buenos resultados

1.4.8.13 Propuesta de solución

En el presente proyecto se realizará la implementación de un control Neuro-PID para el sistema de tanques en serie bajo norma IEC61499, el cual permitirá un control adaptativo para obtener los parámetros k_p , k_i , k_d en tiempo real por medio de una red neuronal previamente entrenado la cual será aporte para la investigación: “Plataforma de reconfiguración dinámica de control automático de un proceso industrial basado en IEC-61499 y multiagentes” aprobado por el Consejo de Investigación e Innovación CONIN, mediante resolución CONIN-P-227-2020 de fecha 24 de septiembre de 2021.

1.5 Objetivos

1.5.8 Objetivo general

Implementar un control Neuro-PID para un sistema de tanques en serie bajo la norma IEC61499.

1.5.9 Objetivos Específicos

- Determinar las condiciones y aspectos para aplicar norma IEC-61499 en un sistema de tanques en serie.

- Desarrollar un controlador NEURO-PID en base a modelos matemáticos adaptables para el control de nivel de un sistema de tanques en serie.
- Validar el control NEURO-PID para el sistema de tanques en serie bajo norma IEC-61499 determinando su correcto funcionamiento.

CAPÍTULO II

METODOLOGÍA.

Materiales

Para el presente proyecto de investigación se basó en la utilización de diferentes herramientas que proporcionaron información encontrados en libros, artículos científicos, proyectos similares al tema planteando, asesoramiento del campo a investigar por parte de profesionales en el área, de igual forma se usaron herramientas de software y hardware para simulación y diseño del sistema de control PID, se emplearon los laboratorios de automatización industrial de la Facultad de Ingeniería en Sistemas Electrónica e Industrial de la Universidad Técnica de Ambato, con el manejo del módulo FESTO® MPS-PA.

Métodos

2.2.1 Modalidad de Investigación

El presente proyecto se fundamentó en una investigación aplicada, utilizando las siguientes modalidades:

Investigación Espacial

El proyecto se desarrolló en la Facultad de Ingeniería en Sistemas Electrónica e Industrial de la Universidad Técnica de Ambato, ubicado en Av. Los Chasquis y Río Cutuchi del cantón Ambato de la provincia de Tungurahua.

Investigación Bibliográfica

El este proyecto investigativo se utilizó una investigación bibliográfica, debido a la obtención de información científica en base al tema de investigación se llevó a cabo principalmente con el uso de revistas científicas, artículos científicos, publicaciones y proyectos de titulación de repositorios públicos y privados desarrollados en los últimos años, cada uno relacionados y vinculados a control PID, Redes Neuronales y Norma IEC-61499.

Investigación de Campo

Debido a que el desarrollo del proyecto se realizó en un entorno real, por lo cual el control Neuro-PID es condicionado de manera funcional bajo cualquier circunstancia respondiendo eficazmente a los cambios.

Investigación Experimental

El proyecto fue efectuando con pruebas del control Neuro-PID para el sistema de tanques en serie bajo norma IEC61499.

2.2.2 Recolección de Información

Para la recolección de información se analizaron varios artículos científicos, tesis, libros, así como guías prácticas y manuales por lo que se tomó en cuenta bases de datos confiables acerca de control Neuro-PID para el sistema de tanques en serie bajo norma IEC61499.

2.2.3 Procesamiento y análisis de datos

Para el procesamiento y análisis de datos se realizó las siguientes actividades:

- Revisión de la información recopilada.
- Interpretación correcta la información recopilada
- Estudio de las propuestas de solución planteada.
- Planteamiento de la propuesta de solución.
- Control y verificación de los datos obtenidos.

2.2.4 Propuesta de solución

En el presente proyecto se realizó la implementación de un control Neuro-PID para el sistema de tanques en serie bajo norma IEC61499, el cual permitió un control adaptativo para obtener los parámetros k_p , k_i , k_d en tiempo real por medio de una red neuronal previamente entrenado la cual es un aporte para la investigación: “Plataforma de reconfiguración dinámica de control automático de un proceso industrial basado en IEC-61499 y multiagentes” aprobado por el Consejo de Investigación e Innovación CONIN, mediante resolución CONIN-P-227-2020 de fecha 24 de septiembre de 2021.

2.2.5 Desarrollo del proyecto

Se desarrolló el proyecto de investigación con los siguientes objetivos planteados:

- 1.** Investigar sobre el estándar IEC-61499 en la gestión de sistemas automatizados.
- 2.** Seleccionar el sistema de tanques en serie a utilizar aplicable a la norma IEC-61499.
- 3.** Esquematizar las variables del sistema de tanques, sus sensores y actuadores.
- 4.** Seleccionar el dispositivo controlador y software para la ejecución del control Neuro PID.
- 5.** Diseñar el control NEURO-PID acorde a modelos matemáticos para el control de nivel del sistema de tanques en serie.
- 6.** Construir y entrenar a la red neuronal de acuerdo a las necesidades del NEURO-PID
- 7.** Implementar el control NEURO-PID para el control de nivel para el sistema de tanques en serie.
- 8.** Elaborar pruebas para la verificación del desempeño de controlador NEURO-PID bajo la norma IEC-61499.
- 9.** Comprobar de los resultados de las pruebas de desempeño del controlador Neuro-PID bajo la norma IEC-61499.
- 10.** Elaborar del Informe Final

CAPÍTULO III

RESULTADOS Y DISCUSIÓN.

3.1 Análisis y discusión de los resultados

El diseño de un control Neuro-PID para un sistema de tanques en serie bajo la norma IEC-61499, permitió diseñar bloques funcionales FBs, modelados que contribuyó a manipular los parámetros presentes en el control automático PID de un proceso industrial como es el de nivel de líquido en los tanques presentes en el módulo de FESTO® MPS-PA, por medio de herramientas de desarrollo como fueron 4DIAC-FORTE y programación en C++, permitiendo ejecutar correctamente el algoritmo de Control Proporcional, Integral y Derivativo (PID) para el nivel de líquido alcanzando un control ideal y óptimo en las pruebas de validación de los valores continuos en el tiempo del sistema industrial, además, se demostró la interacción de los entornos de desarrollo mediante redes neuronales siendo un método confiable a ser aplicado en plantas de procesos industriales, considerándolo como una gran alternativa de estudio en la implementación del estándar IEC-61499, pues los resultados obtenidos demuestran los beneficios de control y predicción que poseen las redes neuronales entrenadas, brindando buenos resultados en diseño de sistemas PID y evaluando la robustez que fue comprobada poniéndolo en funcionamiento con un sistema de nivel de tanque dinámico industrial.

3.2 Desarrollo de la propuesta

Actualmente, las aplicaciones de software de automatización diseñadas para un sistema específico no son compatibles con otros sistemas debido a las patentes que limitan la innovación. Esto impide que los usuarios puedan actualizar sus sistemas de producción de manera sencilla y combinar las mejores soluciones de diferentes proveedores a un costo razonable. La industria experimenta un cambio abismal en todo el mundo, empresas industriales están recurriendo a tecnologías como inteligencia artificial, aprendizaje automático y las arquitecturas de Cloud y/o EDGE. El objetivo de este proyecto es sustituir el controlador PID convencional por un controlador basado en redes neuronales que sea capaz de manejar un sistema de tanques en serie, bajo la norma IEC-61499 empleando computadoras industriales IPC. Este proyecto de

titulación es un aporte para la investigación: “Plataforma de reconfiguración dinámica de control automático de un proceso industrial basado en IEC-61499 y multiagentes, en la figura 18, se muestra el diseño esquemático de la propuesta.

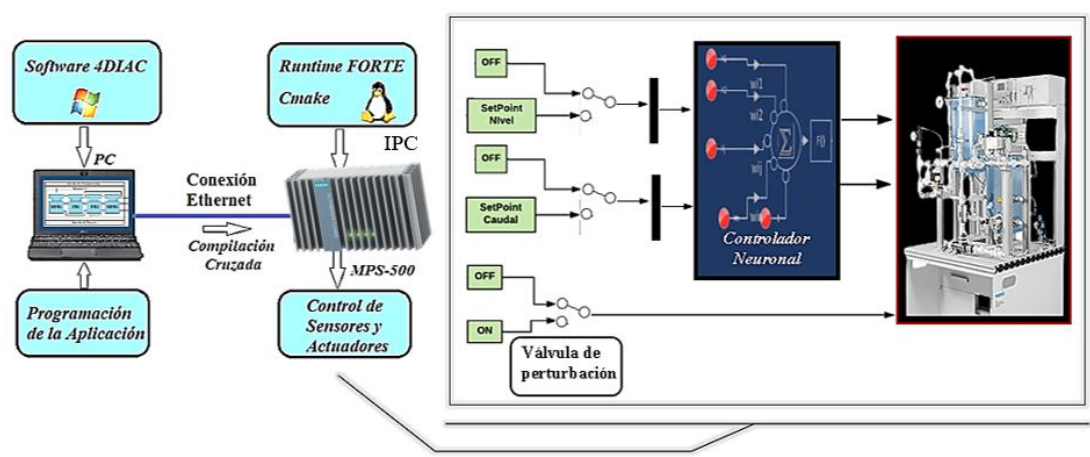


Figura 18. Diagrama del sistema propuesto

Fuente: Elaborado por el investigador

La figura 18 representa el diagrama del sistema que se utilizó en la investigación y que se implementó prácticamente. En primer lugar, se utilizó la plataforma 4DIAC-IDE para desarrollar los bloques funcionales (FBs) que interactúan con el sistema embebido, permitiendo el control y la recepción de acciones interpretadas a través de la plataforma FORTE. De esta forma, se manipulan los parámetros necesarios para controlar el módulo de operación FESTO utilizando el control PID. Todos estos datos se utilizan para el diseño y la creación de una red neuronal que permite recopilar y distribuir información, y así lograr el control Neuro-PID del fluido de nivel del módulo que se desea controlar. Este sistema resulta de gran utilidad para mejorar la eficiencia en procesos industriales, ya que permite ajustar automáticamente los parámetros en tiempo real. Además, esta investigación contribuye a la plataforma de reconfiguración dinámica de control automático de un proceso industrial basado en la norma IEC-61499 y multiagentes.

3.2.1 Descripción del módulo de nivel de tanques FESTO® MPS-PA, para la implementación del sistema de control.

Para el desarrollo del proyecto es fundamental estudiar y reconocer el módulo de nivel de tanques que se va a emplear para implementar el sistema de control, las partes que lo conforman y las características de operación y funcionamiento, se seleccionó el módulo de tanques FESTO® MPS-PA, que se observa en la figura 19, del laboratorio de Automatización Industrial de la Facultad de Ingeniería en Sistemas Electrónica e Industrial de la Universidad Técnica de Ambato, este módulo contiene procesos que permiten estudiar y conocer controles de nivel, temperatura, estados de presión o caudal, siendo independientes entre sí, al ser evaluado y con el conocimiento adquirido de redes neuronales se tiene mayor proyección en el área que se puede aplicar, para el modulo seleccionado se controlará el nivel de líquido del tanque mediante la manipulación de controlador, sensor , válvula y bomba de agua.

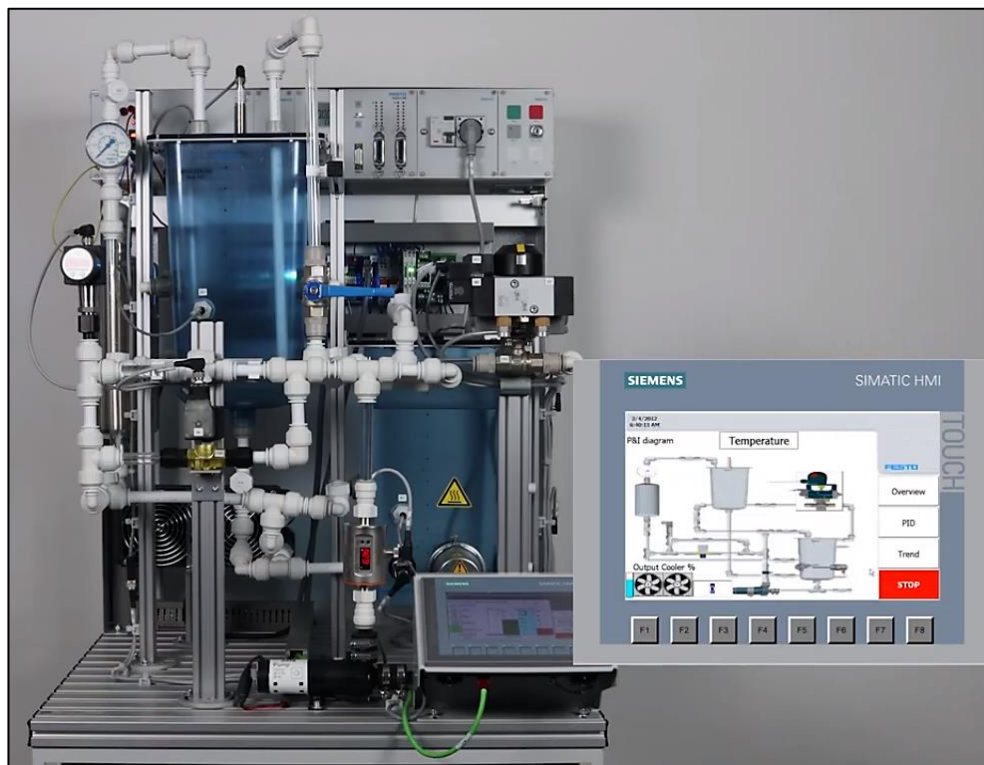


Figura 19. Módulo de tanques FESTO® MPS-PA.

Las partes que conformar el módulo se describe en la siguiente tabla 4.

Tabla 4. Diseño de conformación del Módulo de tanques FESTO® MPS-PA.

Pertenencia	Cantidad	Elemento
Componentes mecánicos	2	Depósito de líquido, tanques
	1	Depósito de presión
	1	Sistema de tubos
	1	Marco de montaje
	1	Placa de perfil
Sensores	2	Sensores capacitivos
	1	Sensor ultrasonido
	2	Interruptores de flotador
	1	Sensor de presión
	1	Sensor de flujo
	1	Sensor de temperatura
Actuadores	1	Bomba
	1	Válvula de control proporcional direccional
	1	Válvula de bola con actuador de proceso neumático
	1	Calefactor
Componentes eléctricos	1	Controlador de motor
	1	Placa de conexión de entrada y salida
	1	Transductor de medición
	1	Syslink
	1	81/80
	1	Terminal analógico
	15	Pines de conexión
Media	1	Manual técnico y documentación con libro de trabajo

Elaborado por: El investigador.

3.2.2 Diagrama de bloques del sistema de control

El sistema de control permite la interacción entre los elementos que componen el funcionamiento del sistema, con el objetivo de manipular las variables de control para llevar a valores deseados o prefijados de este sistema, en la figura 20, se presenta el diagrama del sistema de control.

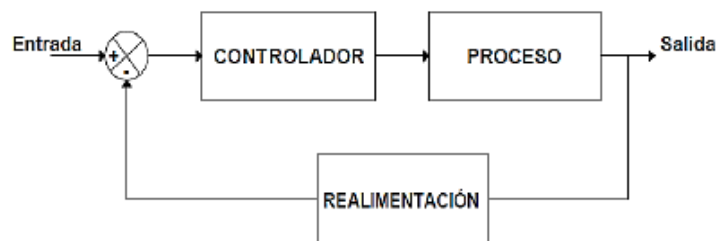


Figura 20. Diagrama de bloques del sistema de control.

3.2.3 Diagrama PI&D de flujo de nivel

La figura 21, presenta el diagrama PI&D con los componentes que conforman el sistema de flujo de nivel al cual se controlará, en primer lugar, se diseña la red neuronal a partir de las variables k_p , k_i , k_d , donde se representa como un modelo matemático que procesa las entradas, pondera los pesos, genera la interacción entre neuronas conectadas, y se aplica la función de transferencia para obtener una salida. Procediendo con el sistema se representan los tanques de almacenamiento de fluido siendo B102 y B101 que se conectan entre sí por medio de tuberías, las válvulas representadas en el esquema como V101 y V110 que permanecen abiertas para el paso del líquido y son operadas de forma manual, el control de fluido de nivel que es monitoreado por los sensores analógicos LIC B101 que arroja datos en tiempo real. La bomba de agua que está representada con P101, que se acciona para suministrar y llevar el líquido de un tanque a otro. El objetivo, mantener constante el fluido de líquido controlado a un cierto nivel por medio del Setpoint incluso si existen factores que perturben el entorno, trabajando en conjunto con la norma IEC 61499 y plataformas de desarrollo compatibles.

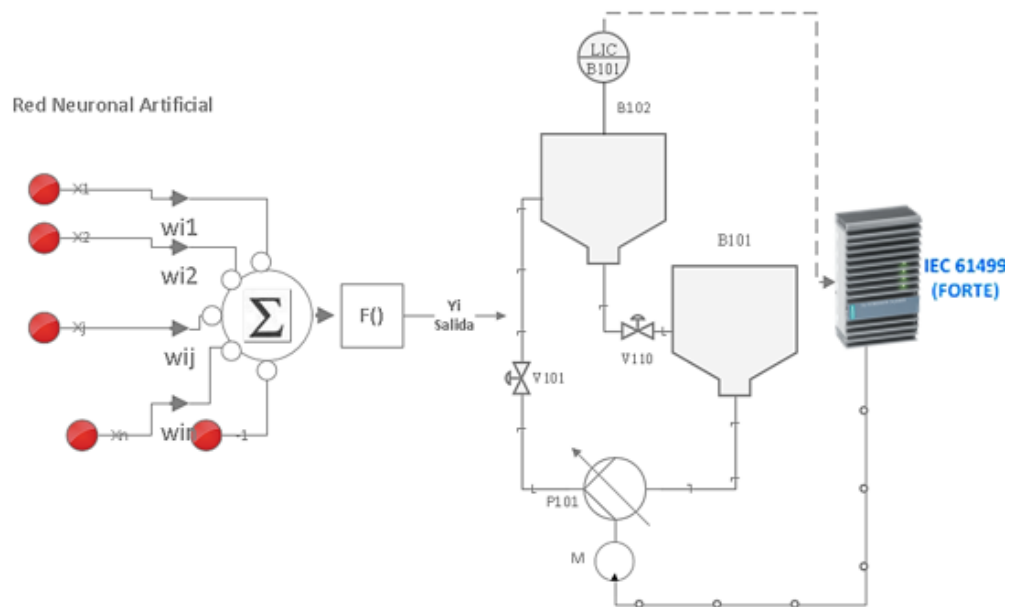


Figura 21. Diagrama PI&D de flujo de nivel.

Elaborado por: El investigador.

3.2.4 Selección de herramientas de software

3.2.4.1 Selección de la plataforma de desarrollo

La plataforma de desarrollo seleccionado, que solventa las necesidades del proyecto en cuestión se denomina 4DIAC-IDE, es un entorno de desarrollo gratuito y de código abierto, siendo utilizado para procesos de automatización e investigación, áreas en los cuales se basa el proyecto en cuestión, además que es extensible con la norma establecida IEC 61499, para ser implementado en procesos de control distribuido, el cual permitió modelar el bloque de control y las interconexiones del proceso mediante el uso de redes, también cuenta con diferentes funciones como configuración de sistemas y aplicaciones, edición y modelado de redes, parametrización de hardware, creación de bloques básicos y compuestos, descarga de aplicaciones, funciones de monitoreo y depuración y funcionalidades de pruebas. Además por la versatilidad y compatibilidad que tiene con las herramientas existentes en la actualidad, la figura 22, demuestra la herramienta de software 4DIAC-IDE y la interacción entre dispositivos de procesos como control distribuido y funcionalidades de prueba [26].

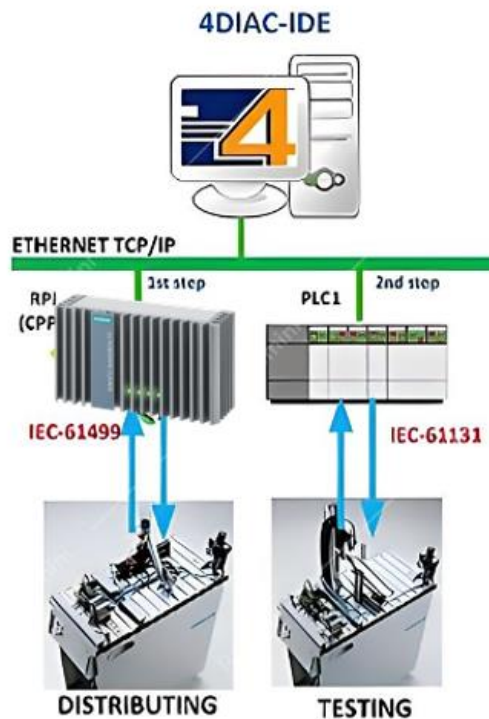


Figura 22. Plataforma de desarrollo 4DIAC-IDE [26].

3.2.4.2 Selección de la plataforma de composición (runtime)

La selección del entorno de tiempo de ejecución que trabaja bajo la Norma IEC-61499, es portable e independiente, es decir, permite la interacción con diferentes dispositivos de control, se basa en programación C++, trabaja conjuntamente con los bloques de función FB proporcionados por la Norma en tiempo real, su compatibilidad con estructuras y datos de otras normas permite que FORTE se adapte a las necesidades requeridas para el desarrollo del proyecto, se desarrollan las aplicaciones con cualquier elemento que conforme la norma, utilizando bloques de funciones básicas o compuestas, de interfaz de servicio, adaptadores, etc. Para conectar los bloques de funciones Forte hace uso de conversiones y proporciona una infraestructura flexible para la comunicación, funcionando en varios sistemas operativos, con es el caso del proyecto en Windows [27].

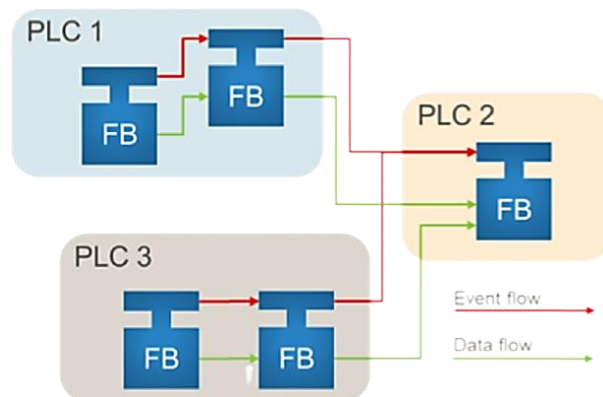


Figura 23. Plataforma de composición 4DIAC-FORTE [27].

3.2.4.3 Selección de plataformas de creación para redes neuronales

Las redes neuronales son un componente fundamental de la inteligencia artificial, ya que permiten a las computadoras aprender, retener, comunicar y procesar datos de forma autónoma, tal como lo hace la mente humana a través de estímulos. Para crear estas tecnologías se requiere un software capaz de manejarlas, y Python resulta una herramienta ideal para este propósito debido a su facilidad de uso y rápida evolución.

Con la plataforma Python, es sencillo establecer y manipular redes neuronales. Se pueden crear nodos de entrada que permiten recibir los datos que la red debe procesar, y nodos ocultos que se encargan de procesar los datos recibidos y transmitirlos a través

de la red. Además, se pueden utilizar nodos de salida para obtener información procesada y transmitirla al exterior de la red. Python es un lenguaje de código abierto, haciéndolo ideal para desarrolladores debido a su sintaxis fácil de leer y entender, lo que a su vez reduce el tiempo de trabajo y ofrece amplios recursos para el desarrollo de redes neuronales. [28].

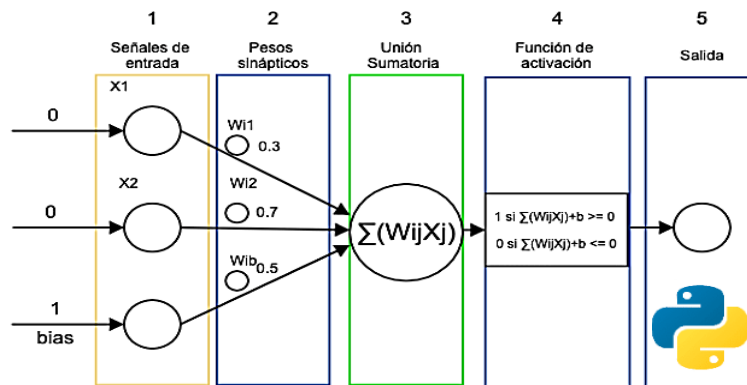


Figura 24. Plataforma de creación de redes neuronales Python [28].

3.2.4.4 Selección de elementos de hardware

Una vez seleccionadas las plataformas de software, se deben seleccionar los elementos que conforman las herramientas de hardware compatibles a ser utilizados para la implementación del sistema de control, como es un dispositivo embebido o un controlador lógico programable. Para ello se debe conocer el concepto de cada grupo seleccionado

Dispositivo Embebido. – Es un sistema electrónico que está diseñado para realizar tareas específicas y se encuentra integrado en otros productos o sistemas más grandes. Los dispositivos embebidos suelen incluir un procesador, memoria, periféricos y sistemas de entrada/salida para interactuar con su entorno. A menudo, utilizan sistemas operativos y lenguajes de programación específicos, y pueden requerir habilidades técnicas especializadas para su desarrollo y programación.

Controlador lógico programable. - Es un dispositivo que permite la programación de una lógica compleja para controlar la operación de máquinas y procesos en la industria, se componen de varios módulos que incluyen una unidad central de

procesamiento, una fuente de alimentación, módulos de entrada y salida, una memoria y un sistema de comunicación.

En la tabla 5 se exponen los tipos de elementos compatibles con la plataforma de desarrollo FORTE considerado en cada grupo.

Tabla 5. Dispositivos controladores que trabajan con 4DIAC-IDE y FORTE

Dispositivos Embebidos	Controladores lógicos programables
Digi Connect ME Device Server Modules	PLC Controlador WAGO PFC200
BeagleBone Black Wireless	SPS-Programmierung mit dem Raspberry Pi
Lego Mindstorms Education	Desarrollador de hardware CONMELEON C1
KIPR Link CBCv2 robot controllers	M-PLC Bachmann electronic GmbH
Raspberry PI Model B	
IPC Siemens 1500	OPC UA C++ for dll

Elaborado por: El investigador.

3.2.4.5 Desarrollo del circuito de entradas y salidas del sistema de control.

Utilizando el arduino nano como un procesador con características suficientes para las necesidades del proyecto, se desarrolló la placa de entradas y salidas del sistema de control PWM de la bomba de agua presente en el módulo de FESTO, en el cual se manipulan las entradas analógicas, siendo los 10 V que arroja el sensor de nivel, y los 24 V de la fuente principal del módulo de tanques, así también las entradas conectadas de la bomba de agua, y la normalización realizada de voltaje que proviene del dispositivo IPC de 10 a 5 V.

Además, se consideran algunos parámetros específicos para el funcionamiento adecuado del procesador como es un divisor de voltaje de 5 voltios a la entrada. Para la implementación de control se conecta un sensor analógico que tiene un rango de 0-5V. a continuación se presenta el esquema del circuito.

3.2.4.6 Esquema del Circuito de acondicionamiento de salida PWM

En la figura 25, se presenta el esquema del circuito realizado en Proteus en el cual se visualiza el montaje del procesador arduino nano conjuntamente con un transistor MOSFET esencial para reducir la tensión en los componentes que regulan las entradas y salidas para controlar el PWM de la bomba de agua, para ser programado de manera que se pueda tener un control de estabilidad en el PID de nivel de los tanques.

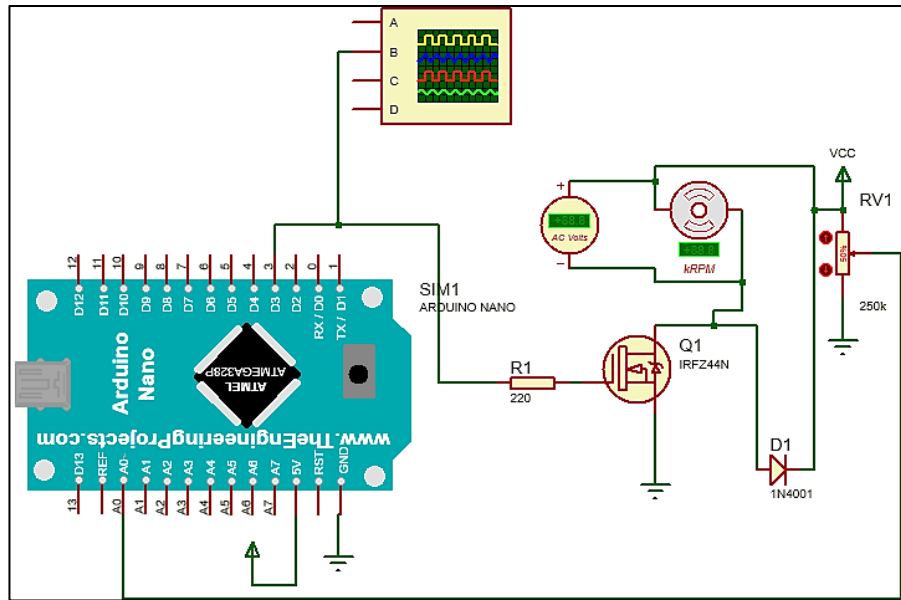


Figura 25. Circuito de regulación PWM

3.2.4.7 Diagrama del circuito de adaptación

Se realizó una adaptación al circuito para la entrada analógica con una regulación de tensión de voltaje entre 0-10 Voltios a un rango de 0-5 Voltios, para ello se implementó y diseño un divisor de voltaje que se visualiza en la figura 26.

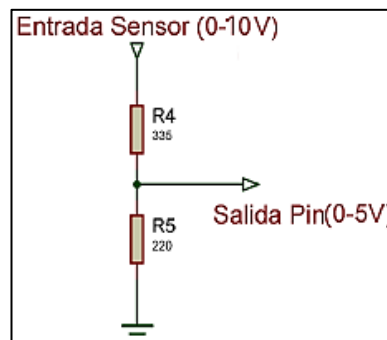


Figura 26. Circuito de adaptación de tensión de 0-5V

- Cálculos realizados para la adaptación del circuito

Para el cálculo se consideró una entrada de 10V y una salida de 5V, con una resistencia del valor de 1kΩ.

$$V_o = \left(\frac{R_2}{R_1 + R_2} \right) * V_{in} \quad (1)$$

$$5V = \left(\frac{1k\Omega}{R_1 + 1k\Omega} \right) * 10V$$

$$5R_1 + 5k\Omega = 10k\Omega$$

$$5R_1 = 5k\Omega$$

$$R_1 = 1k\Omega$$

3.2.4.8 Circuito de acondicionamiento para alimentación del procesador

Como el microcontrolador arduino nano requiere de una fuente de alimentación que debe estar conectada constantemente para funcionar y programarse, se realizó un circuito de acondicionamiento para convertir los 24Voltios que brinda el modulo a 5V por medio de un regulador de tensión L7805CV y capacitores, para ser alimentado en el mismo circuito sin la necesidad de estar conectado mediante cables, a continuación, se presenta el circuito en la figura 27.

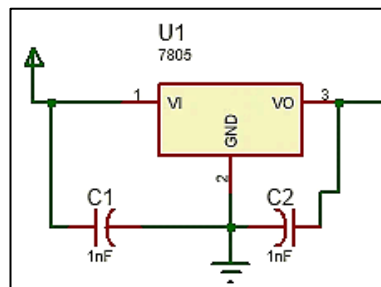


Figura 27. Circuito de acondicionamiento para alimentación del procesador.

3.2.4.9 Diagrama de bloques del sistema para la adquisición de datos

En la figura 28, se muestra el diagrama de bloques del sistema implementado, que representa las etapas que realiza para la adquisición de datos e información del sistema, su procesamiento, así como la salida de las mismas.

Para ello se recolecta los datos reales del comportamiento del módulo, etapa indispensable a evaluar pues de ello depende el desarrollo óptimo de las etapas contribuyentes al proyecto, y el objetivo principal del mismo como es el control óptimo de nivel de tanques de un sistema. Para la adquisición de datos se utilizó Matlab como plataforma de entorno en el cálculo de la función de transferencia, además para la interacción y comunicación de la tarjeta de desarrollo Arduino nano y la PC, y para los cálculos realizados necesarios para convertir señales de voltaje y de flujo para el control del sistema hidráulico.

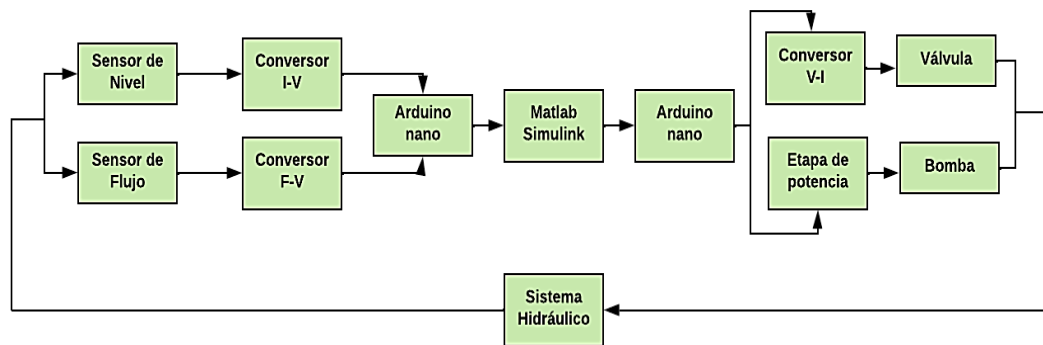


Figura 28. Diagrama de bloques del sistema para la adquisición de datos.

Elaborado por: El investigador.

3.2.4.10 Diseño de bloques funcionales en 4DIAC bajo la norma IEC-61499

Los bloques funcionales son la sección donde se lleva a cabo la programación de un proyecto, implementando un controlador basado en tiempo clásico que toma datos de manera constante y en función de los eventos desarrollados.

Tomando en cuentas estos aspectos el software que permite solventar las necesidades de los bloques comportamentales del controlador es la plataforma 4DIAC-IDE, que se basa en entornos y herramientas de desarrollo permitiendo la interacción de hardware de distintos fabricantes industriales.

Con 4DIAC se desarrolló el modelo funcional del sistema creando bloques para la programación del controlador, así como las aplicaciones que llegan a interpretarse por el entorno FORTE, estas acciones se van ejecutando en el sistema de control, la figura 29, muestra la estructura realizada de control, que se conforma de los bloques BFB (Bloques básicos de función), conformándose por el grafico de control de ejecución,

ECC, que contiene los estados INIT, RST, y REQ, ejecutándose los algoritmos que permiten la transición de los eventos pasando de un estado a otro, mediante los eventos también se envían datos de salida, como CNF, mientras que el algoritmo REQ permite realizar el cálculo del controlador PID.

Aplicando el ECC se mantiene las partes de inicialización y reset del sistema controlado, para la etapa de ejecución este recibe el evento REQ para entrar al estado de check point, factor importante en el control del tiempo a evaluar entre la última ejecución y el momento actual, las transiciones que arroja el proceso son comprobadas como condiciones evaluadas en los parámetros del controlador de manera óptima a tener una mayor precisión en el estado SENDNOMOD solo envía el evento pues no tiene que ejecutarse ninguna acción, y el evento NO_MOD, realiza a acción contraria ejecutando el algoritmo REQ para realizar acciones nuevas de control. Este proceso se realiza para el desarrollo de los elementos y bloques que incluye el control del sistema modelando una ejecución óptima que llame a cada bloque programado FBs, y los eventos necesarios para el control de tiempo de procesamiento, para los cálculos requeridos dentro del controlador, de igual manera para la representación de la salida PWM.

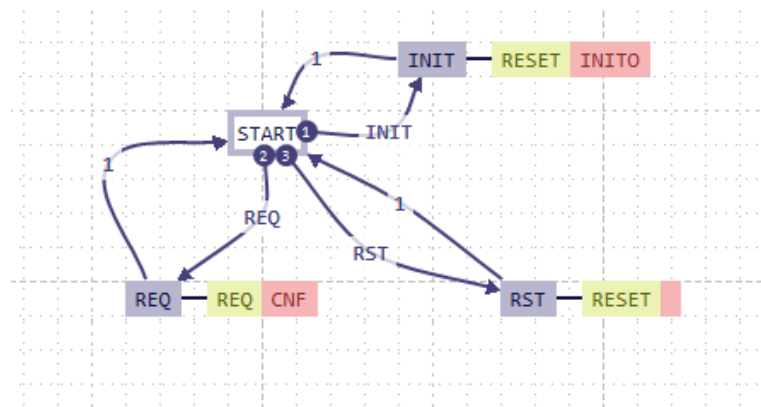


Figura 29. Gráfico del control de ejecución del controlador PID del sistema.

La aplicación se ejecuta en la fase de inicialización de un evento en la entrada INIT, que contiene el bloque FB, una vez ejecutado se inicia la cadena de eventos relacionados entre si hasta llegar a la salida controlada PWM, que permite efectuar la fase de ejecución del bloque que contiene el evento START, dando inicio a los eventos y las ejecuciones de los bloques de manera periódica calculando de esta forma los datos necesarios para el control del sistema.

3.2.4.11 Bloque de función del controlador modelado de ejecución

El objetivo principal es obtener el mejor método de evaluación de un controlador en un proceso industrial bajo la Norma IEC-61499, para ello se lleva a cabo el desarrollo de un controlador periódico que muestre la respuesta del sistema ante el control en lo relativo a la respuesta de nivel de líquido de tanques presente en el módulo FESTO.

Se considera la suma de los tiempos de reacción de ejecución necesarios para que cada bloque realice la acción correspondiente a los eventos, así como la cantidad de ejecuciones del controlador. Esto se expresa mediante la siguiente ecuación.

$$t_{PIDtb} = Q(t_{AIN} + t_{PID} + t_{PWM}) \quad (1)$$

Donde:

t_{PIDtb} = La suma del tiempo que consume cada FBs

t_{AIN} = tiempo de ejecución del bloque que recibe la lectura a la entrada analógica

t_{PID} = tiempo de ejecución del bloque de control PID

t_{PWM} = tiempo de ejecución del bloque de escritura de la salida analógica.

En la figura 30, se modela el bloque general de la modelización de la aplicación de control PID basado en la Norma IEC-61499 desarrollado en el entorno 4DIAC, en el cual se generan los algoritmos de control para ejecutarse completamente, estos tiempos de ejecución se representan en la ecuación (2), observando que el tiempo t_{PIDe} representa el consumo de tiempo del bloque PID cada vez que se ejecuta una nueva acción de control, y el tiempo t_{PIDne} el tiempo normal de ejecución sin cálculos previos de control, y t_{CLK} el tiempo de cada bloque CLOCK que se obtiene en la ejecución del mismo procesador, representando la nueva ecuación de la siguiente manera.

$$t_{PIDeb} = Q(n(t_{AIN} + t_{PIDe} + t_{PWM} + t_{CLK}) + (1 - n)(t_{AIN} + t_{PIDne} + t_{CLK})) \quad (2)$$

3.2.4.12 Bloques de funciones

En la figura 30, se observa los bloques necesarios en 4diac para poder realizar la configuración del PID, teniendo las entradas y salidas analógicas que se establecen con el IPC, aquí se puede leer los valores del Nivel de tanque y la salida analógica necesaria para poder realizar el control PID mediante las PWM, a su vez cuenta con una entrada digital que se encarga de activar o desactivar el sistema.

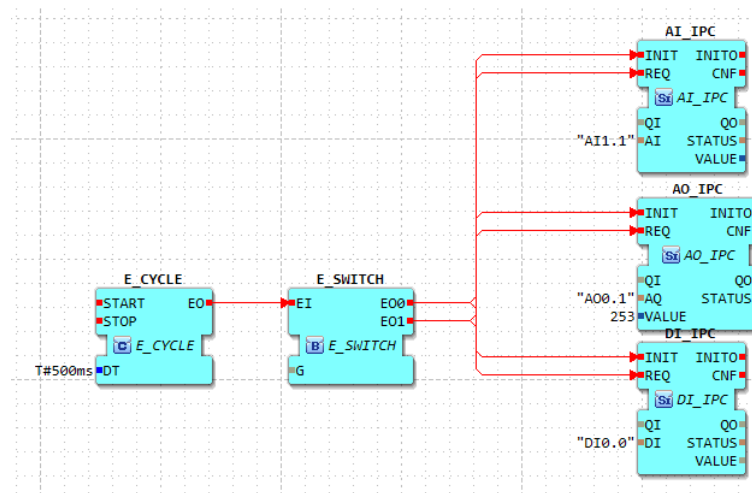


Figura 30. Bloques de funciones desarrollados para lecturas y escrituras analógicas

3.2.4.13 Aplicación de la norma IEC-61499 para crear el módulo PID

La norma IEC 61499 es una norma internacional para la programación de sistemas distribuidos basados en controladores lógicos programables (PLC) y sistemas embebidos. Esta norma se enfoca en proporcionar una arquitectura de programación modular y flexible para el desarrollo de sistemas de control distribuido.

En cuanto a su aplicación en el control PID (Proporcional-Integral-Derivativo), la norma IEC 61499 se utilizó para crear un módulo de controlador PID, que puede ser reutilizado en diferentes aplicaciones. Para ello, se calculó una función de transferencia controlador PID que se adapte a diferentes entradas y salidas de los sistemas a controlar, la norma IEC 61499 también se aplicó en el control de 4DIAC para diseñar y programar módulos que permitan el control de los DIAC de manera distribuida, logrando la creación del bloque PID que se pueden reutilizar en diferentes sistemas de control, lo que reduce el tiempo y el costo de desarrollo.

En la figura 31, se observa el bloque creado para el PID que cuenta con las variables K_p , K_i , K_d que son las constantes que dará la inteligencia artificial para poder determinar el valor de la salida PWM, también se establece el ingreso del setpoint deseado.

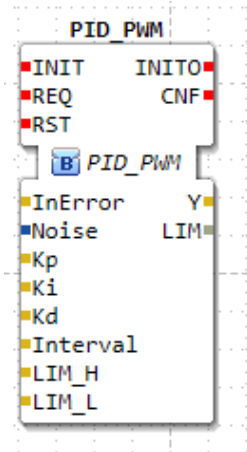


Figura 31. Bloque PID

En la figura 32, se muestra el bloque de la inteligencia artificial, el cual se encarga de establecer los valores necesarios del PID para que funcione el sistema. Para esto se utiliza el parámetro de entrada Setpoint para poder determinar que valores de K_p , K_i , K_d se necesita, con eso se realiza la programación dentro del bloque.

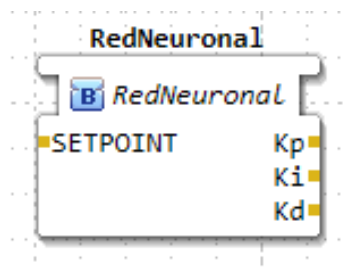


Figura 32. Bloque de la red neuronal

3.2.4.14 Gráfica de respuesta del proceso ante el impulso escalón

En la figura 33, se muestra la gráfica del sistema de tanques con los valores obtenidos de Arduino, los mismos que posteriormente se utilizarán para poder sacar la función de transferencia y las constantes PID.

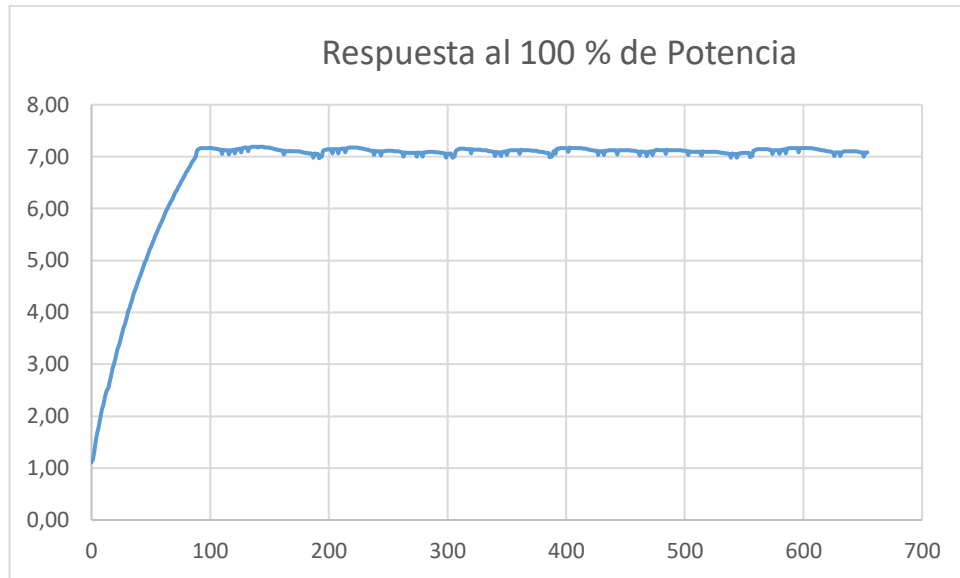


Figura 33. Grafica con los datos de Arduino

3.2.4.15 Implementación del algoritmo de control PID

Sintonización del controlador PID

Para realizar la sintonización del controlador PID, se desarrolla la programación en Arduino la cual consiste en que después de 15 segundos se aplica una potencia al 100% equivalente al 100% de PWM y se puede observar cómo actúa el sistema, teniendo una respuesta tal como se muestra la figura 34. Se establece en un sistema de primer orden y una vez realizado la programación se procede a tomar los datos tanto del nivel como el valor del impulso escalón para proceder a ponerlos en Matlab como se visualiza en la figura 32.

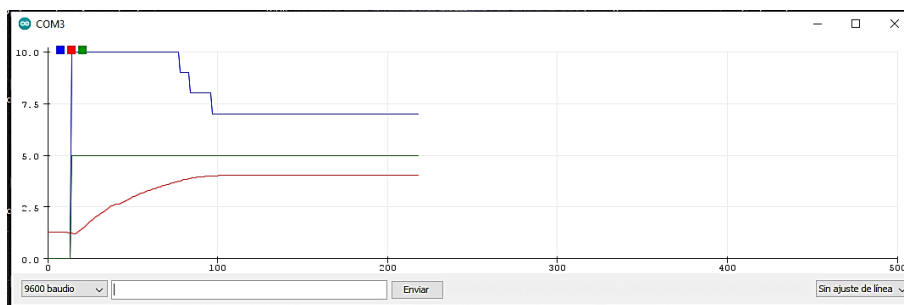


Figura 34. Respuesta al impulso escalón en Arduino

Después de recolectar los datos con Arduino, se transfieren al workspace de Matlab. Posteriormente, se utiliza la función `ident` para determinar la respuesta al enviar una

señal de escalón utilizando los datos obtenidos previamente con Arduino. En la figura 35, se muestran los valores correspondientes al nivel de agua y la potencia aplicada, los cuales han sido registrados en el workspace de Matlab.

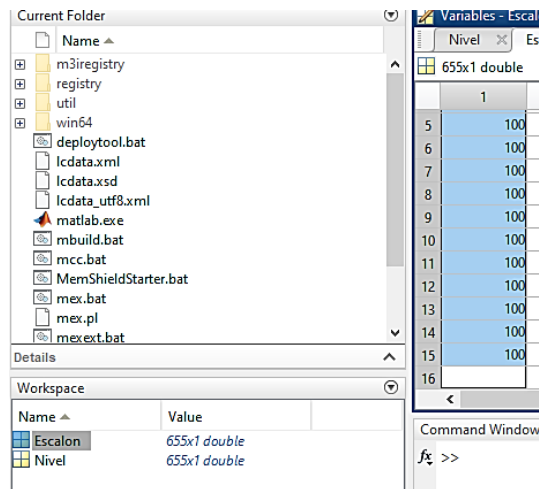


Figura 35. Valores procesados mediante Matlab

Al utilizar la función `ident` mostrará una ventana para poder importar los valores del Escalón y del Nivel y así visualizar su gráfica, para posteriormente encontrar su función de transferencia, esta ventana cuenta con diferentes opciones como se muestra en la figura 36.

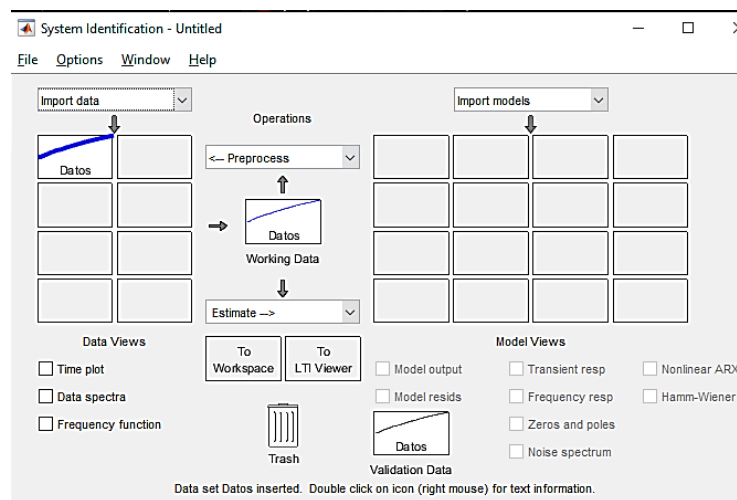


Figura 36. Valores procesados mediante Matlab

Se realiza la función de transferencia utilizando el software de Matlab, al seleccionar la opción "process models", se genera una gráfica que debe coincidir con un valor

superior al 75%, como se muestra en la figura 37. El software de Matlab proporciona una función de transferencia con un valor del 87.38%, lo que es adecuado para el trabajo a realizar. La gráfica obtenida a partir de la programación en Arduino muestra una respuesta de la función de transferencia estable que coincide con la gráfica generada por el software de Matlab.

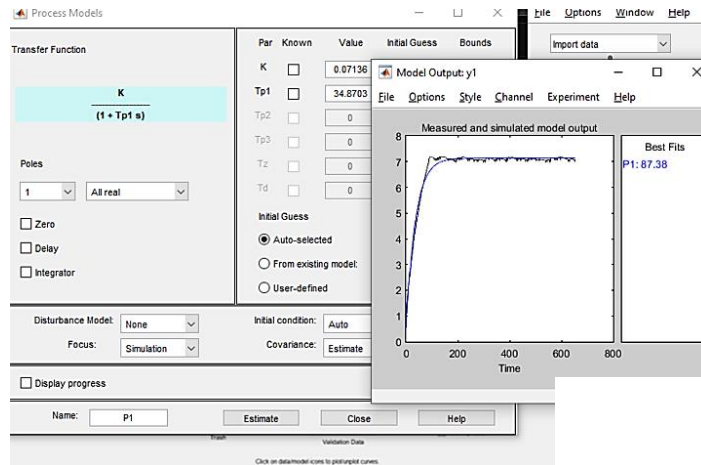


Figura 37. Gráfica de la función de transferencia

En la figura 38, se muestra la gráfica que tiene una relación entre el tiempo y la amplitud para el “step response” observando su curva que equivale a nuestro sistema de primer orden, para poder realizar su control PID.

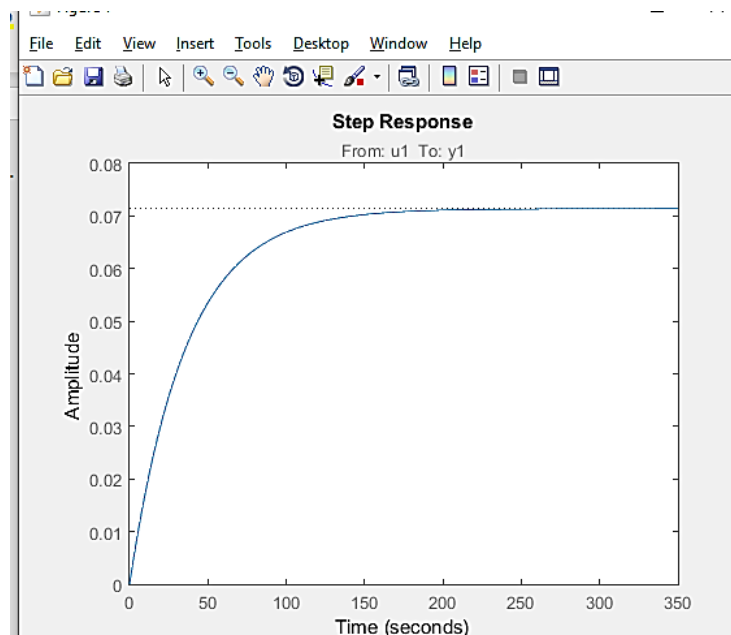


Figura 38. Gráfica de la amplitud en función del tiempo

La función de Transferencia que se obtiene para el presente sistema del proyecto mediante el software de Matlab y los valores obtenidos mediante Arduino es:

$$G = \frac{0.07136}{34.87 S + 1}$$

Implementación del sistema de control

Por último, mediante PID Tuner se procede a sacar las constantes del PID regulando la gráfica que nos presenta en la figura 39, para obtener la que mejor sea acorde para el sistema.

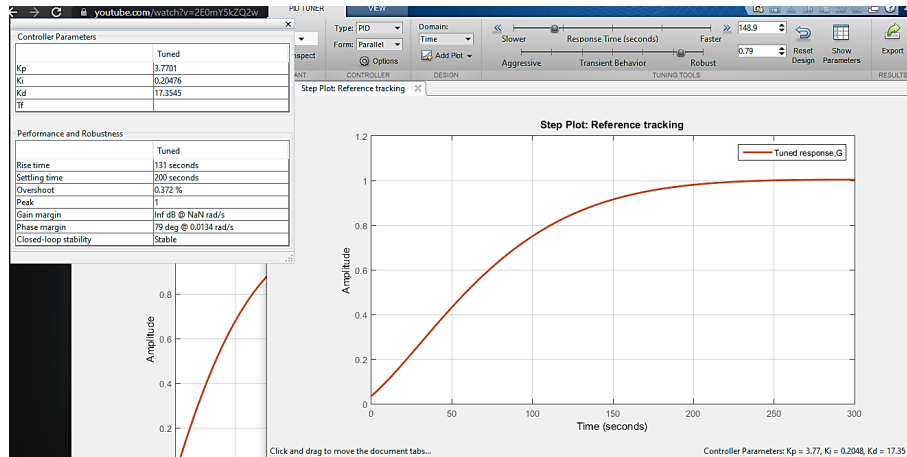


Figura 39. Gráfica de respuesta mediante las constantes PID

3.2.4.16 Programación en Tía Portal

En Tia Portal se realiza la programación para la lectura del valor de nivel del tanque mediante los bloques que se muestran en la figura 40. Primero se realiza la normalización y después se desarrolla el escalamiento para una correcta visualización de los datos.

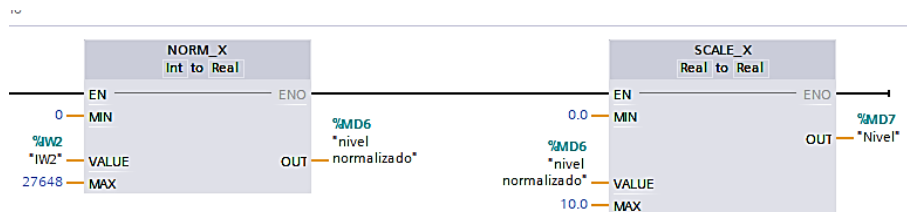


Figura 40. Lectura del nivel del Tanque

Además, se ha trabajado en la implementación de la salida PWM, la cual proporciona una señal analógica que se dirige hacia la placa desarrollada anteriormente. La programación correspondiente se muestra en la figura 41.

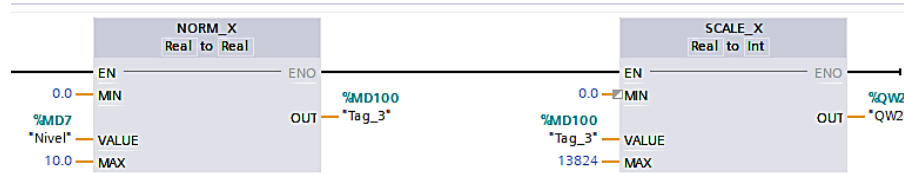


Figura 41. Salida analógica para los valores de PWM.

3.2.4.17 Diseño de la red neuronal

El controlador PID es ampliamente utilizado para procesos industriales debido al estudio previo de su funcionamiento en este proyecto se sabe que su arquitectura funciona como un lazo cerrado y compara el valor de la salida del sistema con el valor deseado, manipulando y calculando los valores que comprende un control como son las ganancias proporcionales, integrales y derivativas k_p , k_i y k_d , respectivamente, integrando la red neuronal en el estudio de estos parámetros se puede basar en suposiciones sobre el comportamiento del sistema, de acuerdo a una investigación un sistema no lineal se expresa tanto en sus entradas como en la salida [29], en términos matemáticos de la siguiente manera:

$$\hat{y}(k) = NN \left[\begin{matrix} \hat{y}(k-1), \dots, \hat{y}(k-n), u(k-1), \dots, u(k-n), \\ r(k-1), \dots, r(k-n), e(k), \dots, e(k-n) \end{matrix} \right] \quad (3)$$

Donde:

NN =red neuronal

\hat{y} =salida de la red neuronal

u =señal de control

r =punto de referencia

e =señal de error

n =retardo en el tiempo

En la figura 42, se representa el esquema de la RNA, diseñada para el controlador de sistema.

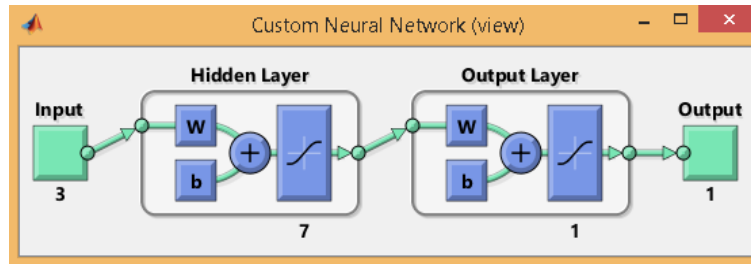


Figura 42. Esquema de la red neuronal aplicada al controlador del sistema.

Elaborado por: El Investigador

El sistema no lineal se puede controlar con la red neuronal la cual evalúa y predice los valores de control PID, en el esquema de la figura 43, se representa el controlador basado en la velocidad de la regulación del flujo emitido por la bomba de agua controlando los PWM, para ello la ecuación se expresa de la siguiente manera.

$$u(k) = u(k - 1) + k_p(e(k) - e(k - 1)) + k_1 e(k) + k_D e(k) - 2e(k - 1) + e(k - 2)) \quad (4)$$

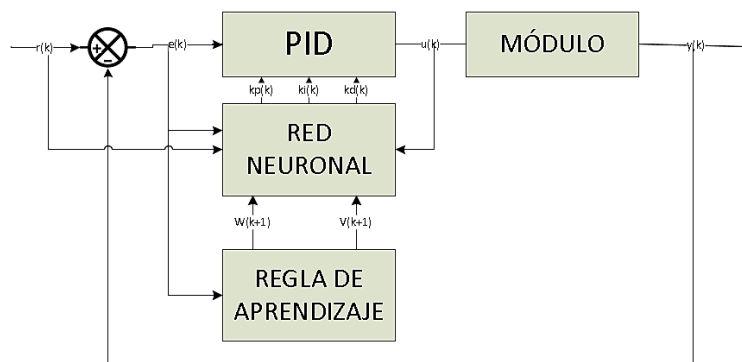


Figura 43. Esquema del control PID del módulo con la red neuronal

3.2.4.18 Datos de Aprendizaje para la red Neuronal

Se crea un conjunto de datos de entrenamiento y de prueba. En la figura 44, el conjunto de datos incluye 6 datos de entrada (k_p , k_i , k_d , setpoint, entrada de líquido y nivel del tanque) y 3 datos de salida (los parámetros futuros del control PID).

```

Serial.print("AntKp: ");
Serial.print(Kp);
Serial.print(" AntKi: ");
Serial.print(Ki);
Serial.print(" AntKd: ");
Serial.print(Kd);
nivelT = 5.0 * 2 * analogRead(Sensor_nivel) / 1024.0;
error[0] = setpoint - nivelT;
d_error[0] = error[0] - prev_error;
integral += error[0];
output[0] = abs(Kp * error[0] + integral - Kd * d_error[0]);
Kp -= learning_rate * error[0] * output[0];
Ki -= learning_rate * error[0] * integral;
Kd -= learning_rate * error[0] * d_error[0];
prev_error = error[0];
outputP = output[0];
Potencia = abs(map(outputP, 0, 100, 0, 255));
if (Potencia > 255) {
  Potencia = 255;
}
analogWrite(bomba, Potencia);
Serial.print(" L: ");
Serial.print(nivelT);
Serial.print(" P: ");
Serial.print(Potencia);
Serial.print(" Kp: ");
Serial.print(Kp);
Serial.print(" Ki: ");
Serial.print(Ki);
Serial.print(" Kd: ");
Serial.println(Kd);

```

Figura 44. Recolección de datos para aprendizaje para la predicción de Kp Ki Kd

3.2.4.19 Implementación red neuronal

Se genera un conjunto de datos de entrenamiento aleatorios y entrena una red neuronal con tres capas ocultas utilizando el algoritmo de optimización Adam y la pérdida de error cuadrático medio. En la figura 45, la red neuronal tiene cinco entradas, que incluyen los valores actuales de kp, ki y kd, así como el nivel del tanque y la entrada de líquido. Una vez entrenado, el modelo se puede utilizar para hacer predicciones de los valores futuros de kp, ki y kd a partir de los valores actuales, el nivel y la entrada de líquido en tiempo real.

```

1 # Importar las librerías necesarias
2 import numpy as np
3 from keras.models import Sequential
4 from keras.layers import Dense
5
6 # Defina las constantes
7 N_INPUTS = 5 # Número de entradas (kp, ki, kd, nivel del tanque, entrada de líquido)
8 N_OUTPUTS = 3 # Número de salidas (kp, ki, kd)
9 N_SAMPLES = 1000 # Número de muestras de entrenamiento
10
11 # Generar datos de entrenamiento
12 X_train = np.random.random((N_SAMPLES, N_INPUTS))
13 y_train = np.random.random((N_SAMPLES, N_OUTPUTS))
14
15 # Crear el modelo de red neuronal
16 model = Sequential()
17 model.add(Dense(64, input_dim=N_INPUTS, activation='relu'))
18 model.add(Dense(32, activation='relu'))
19 model.add(Dense(N_OUTPUTS, activation='linear'))
20
21 # Compilar el modelo
22 model.compile(loss='mean_squared_error', optimizer='adam')
23
24 # Entrenar el modelo
25 model.fit(X_train, y_train, epochs=10, batch_size=32)
26
27 # Hacer predicciones
28 X_test = np.random.random((1, N_INPUTS))
29 y_pred = model.predict(X_test)
30 print(y_pred)

```

Figura 45. Código de la NN basado en Keras para la predicción de Kp Ki Kd

Diseño de la red neuronal puesta con Neural Network Toolbox y de Simulink

Para llevar a cabo el diseño neuronal se aplica no solo en el software Python sino también en Matlab que posee herramientas para facilitar el modelo de control de la red, para ello en primer lugar se recopilan los datos obtenidos por los cálculos hechos en las mediciones de entradas y salidas del sistema de control, a continuación, se muestra la tabla de datos en la siguiente figura 46.

	A	B	C	D	E	F	G	H	I
1	Kp	Ki	Kd	SET POINT		SET POINT	Kp	Ki	Kd
2	37,701	0,20476	173,545	2		37-38	0,1-0,6	173-175	
3	37,92066	0,38942	173,7053	2		38-39	0,6-1	175-176	
4	39,45439	1,23487	177,3937	4		39-40	1-1,6	177-178	
5	42,8525	2,53247	183,4667	7		40-41	1,5-2	179-180	
6	40,12683	1,57324	179,8277	5		41-42	2-2,5	181-182	
7	38,31346	0,5880424	173,867	2		42-43	2,5-3	183-184	
8	38,244	0,62934	175,284	3					
9	41,9872	2,325128	181,2521	6					
10	38,3575	0,2233621	173,9621	2					
11	42,95165	2,8950647	184,5552	7					
12	38,58026	0,9269107	176,219	3					
13	39,79355	1,3232768	177,6019	4					
14	38,18157	0,5469701	174,2805	2					
15	41,70836	1,9508395	181,8687	6					
16	43,18313	2,6086913	184,0206	7					
17	41,91025	2,4909549	181,5044	6					
18	41,09814	1,8404717	179,8029	5					
19	38,41995	0,3136807	174,2833	2					
20	42,29346	2,8333214	181,6882	6					
21	40,00559	1,8465353	178,2542	4					
22	38,96039	1,1077442	176,4689	3					
23	42,06263	2,1465407	182,4474	6					
24	43,14274	3,4504862	183,8777	7					
25	38,43974	0,5698702	173,825	2					
26	40,91684	1,6142217	180,157	5					
27	41,51462	1,9444259	181,9723	6					
28	43,12524	3,3642047	184,0799	7					
29	37,95746	0,0060422	174,1327	2					
30	42,44317	2,8962585	182,2813	6					
31	39,35303	1,4403743	177,6259	4					
32	38,91636	0,4541398	175,8863	3					
33	41,91605	2,357746	181,9076	6					
34	43,41889	3,4553126	184,1834	7					

Figura 46. Datos de medición de entrada y salida del sistema

Luego de agrupar los datos, se selecciona una cantidad determinada de ellos para utilizar en el entrenamiento de la red neuronal, mientras que una cantidad más reducida se utiliza para llevar a cabo pruebas. En este caso, se han tomado 300 datos para el entrenamiento y 100 datos para las pruebas. Estos datos se importan a la interfaz, tal como se muestra en la figura 47.

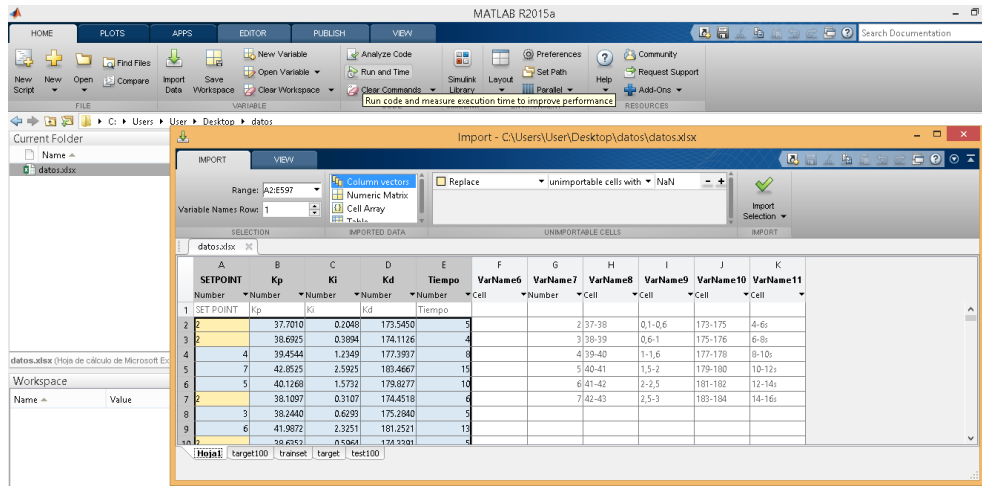


Figura 47. Importación de datos para crear la red neuronal

Para crear la red neuronal se llama al comando (nntool) el cual despliega una ventana intuitiva, vista en la figura 48.

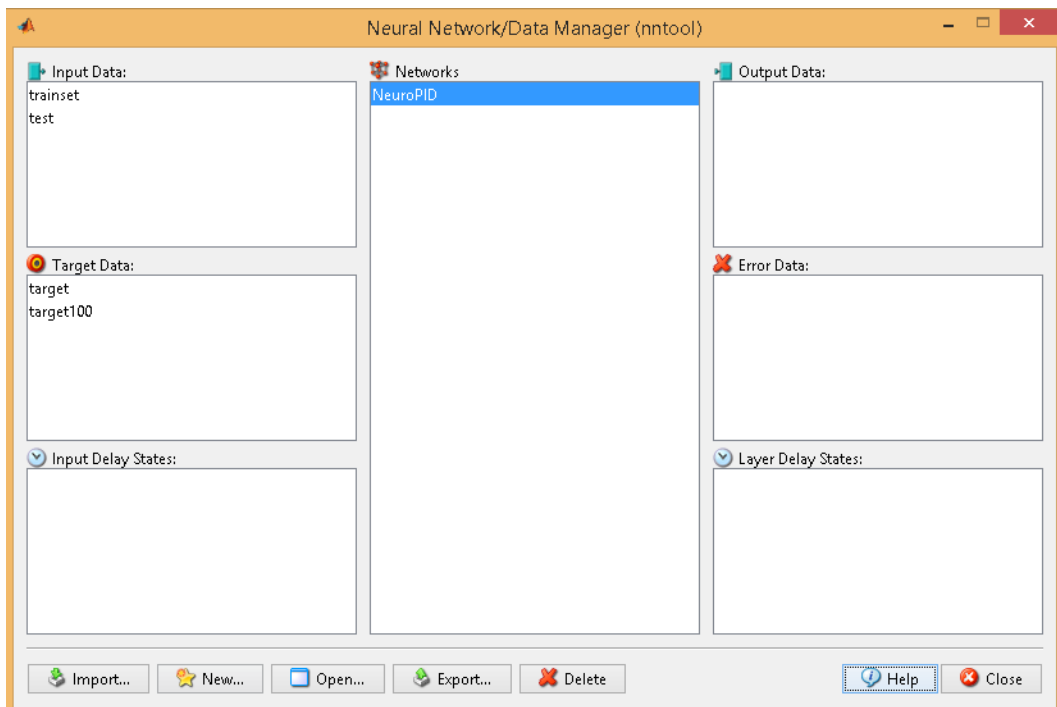


Figura 48. Ventana de interfaz para crear una red neuronal

Se procede a seleccionar las variables de entradas y salidas de los datos, y modificar parámetros acordes al comportamiento que se quiera dar a la neurona en este caso se crea 7 neuronas las cuales se distribuyen en la siguiente topología de la figura 49.

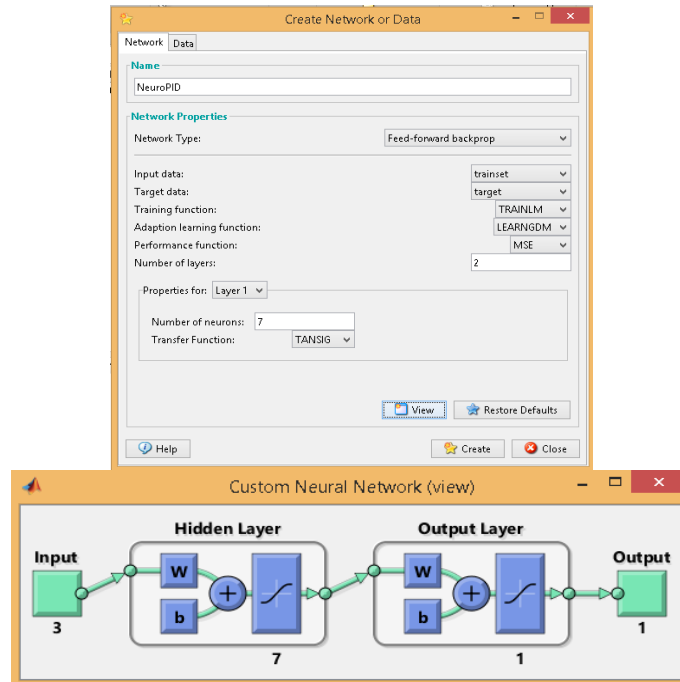


Figura 49. Topología de la red neuronal

Se crea la neurona y se procede a entrenarla, para ello dentro de esta se especifican las entradas y las salidas evaluadas, así como también el error, en la figura 50, se aprecia el proceso.

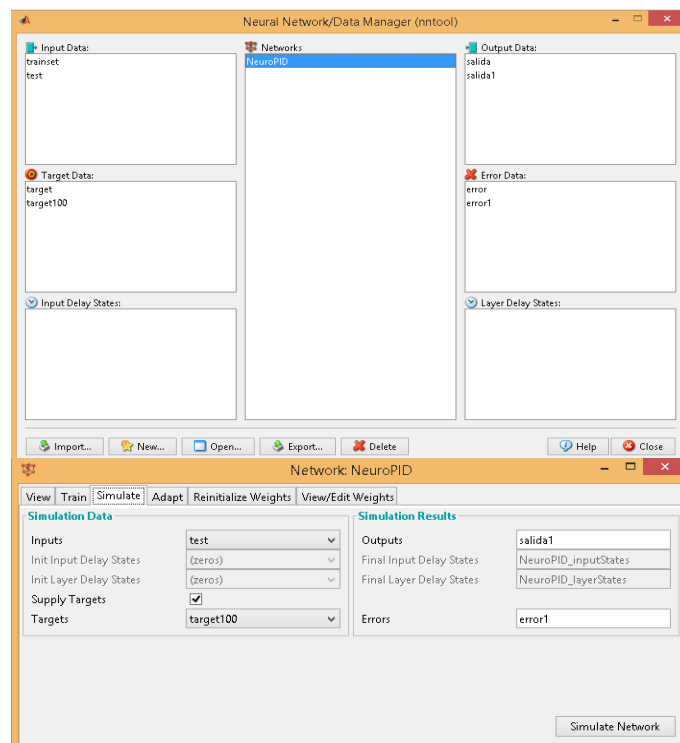


Figura 50. Configuración de la red neuronal

Una vez que se ha modificado cada parámetro, se procede a iniciar el proceso de entrenamiento de la neurona mediante la función "TrainNetwork". Este proceso se lleva a cabo tantas veces como sea necesario para lograr el mejor rendimiento posible. En la figura 51 se puede observar el resultado final del proceso de entrenamiento, que requirió de 1000 iteraciones y validaciones para alcanzar el nivel óptimo de rendimiento

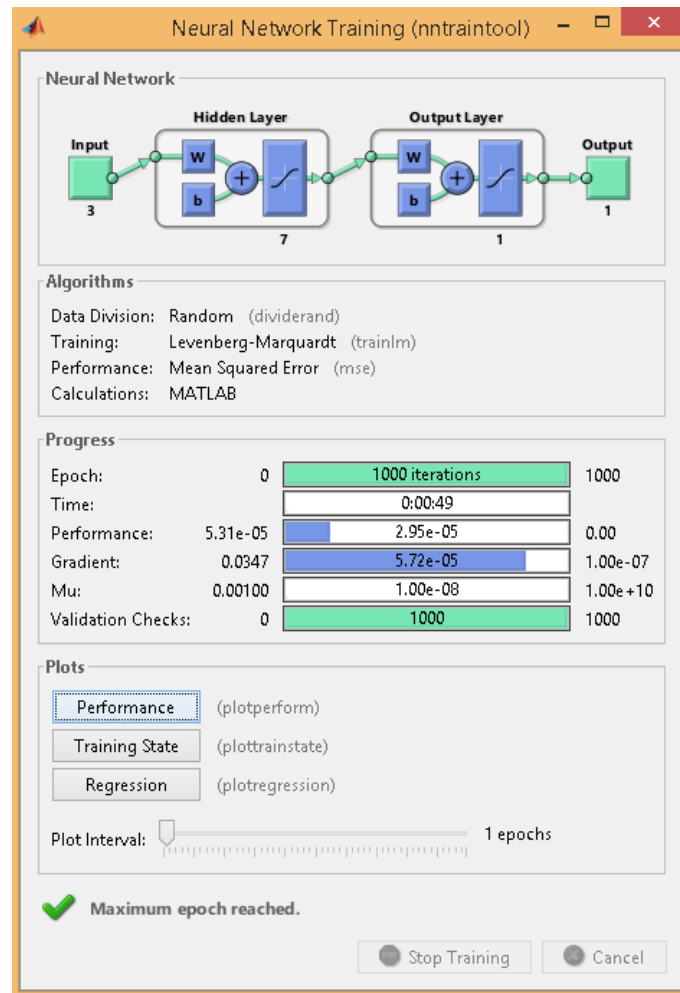


Figura 51. Entrenamiento de la neurona con 1000 interacciones

En la matriz de la figura 52, se pueden observar los parámetros de entrenamiento (Training), validación (Validation), prueba (Test), todo(All), entre más alto sea el valor de R, mejor será la eficiencia o rendimiento de la neurona, en este caso es 99%.

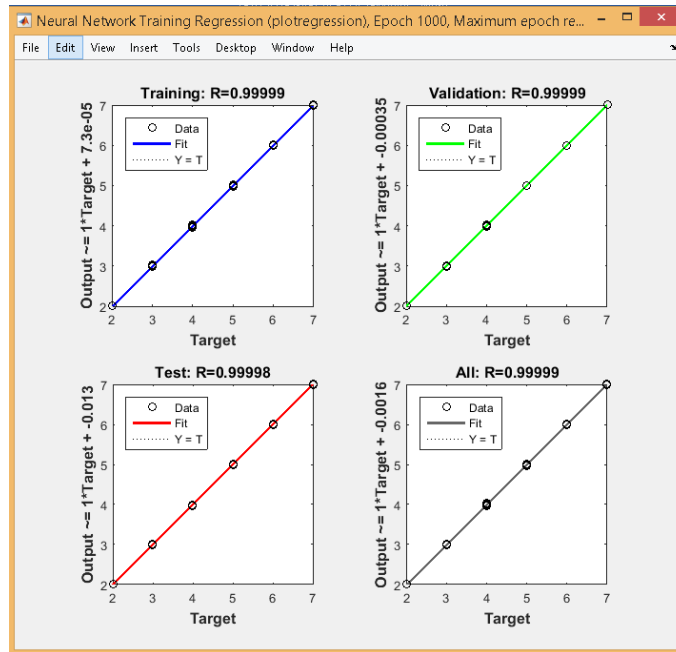


Figura 52. Eficiencia y rendimiento de la neurona.

Diseño de la placa para envío de PWM

El método de experimentación para evaluar el controlador comportamental basado en la norma IEC-61499, se conformó por el IPC, por sus características tanto de entradas como salidas útiles a ser aplicados al proyecto, de manera que buscando la eficiencia de tener un control optimo, simple y versátil se aplicó un circuito electrónico para acoplar al sistema a controlar de manera que permita la interacción de los componentes con las partes del sistema, el circuito se muestra en la figura 53 y figura 54.

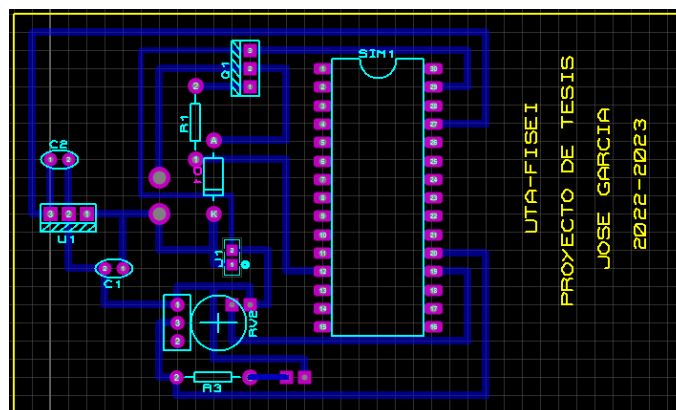


Figura 53. Esquema PCB del circuito aplicado al sistema de control

Elaborado por: El Investigador

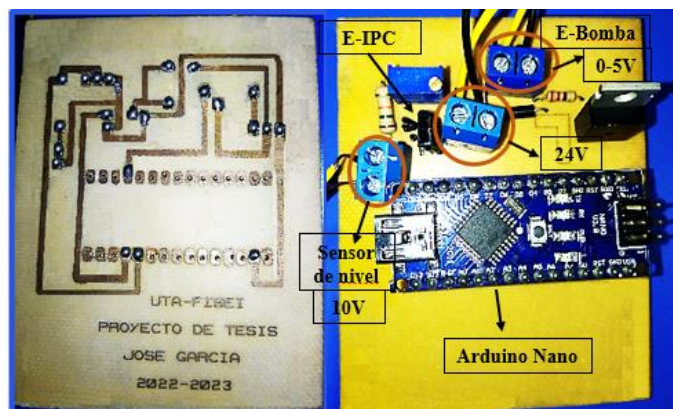


Figura 54. Circuito ensamblado aplicado al sistema de control
Elaborado por: El Investigador

Para llevar a cabo las pruebas experimentales, se programa la tarjeta que se conecta a un ordenador y se accede al entorno de directorios mediante comandos para activar los eventos utilizando FORTE runtime. Una vez compilado el código, se carga en la tarjeta que se conecta a la placa de control, lo que permite ejecutar automáticamente el programa de los bloques en 4DIAC.

Programación del arduino nano

En el arduino nano se realiza la programación referente al envío de PWM a la bomba, ya que el IPC no cuenta con un sistema para generar PWM en sus salidas digitales, en la figura 55, se muestra la programación utilizada en donde se declara una variable para la lectura de los valores analógicos del IPC y se realiza la equivalencia con la PWM que se enviarán mediante el pin 9.

```

int sensorPin = A0;
int bomba= 9;
int sensorValue = 0;
void setup() {
  // declare the ledPin as an OUTPUT:
  pinMode(ledPin, OUTPUT);
}

void loop() {
  // read the value from the sensor:
  sensorValue = analogRead(sensorPin);
  Potencia = map(sensorValue, 0, 1024, 0, 255);
  analogWrite(bomba, 128);
}

```

Figura 55. Programación referente al envío de PWM

3.2.5 Pruebas de funcionamiento

En Tabla 6, se establece el rango de valores tomados por la generación de datos de entradas y salidas del sistema, para ser considerados como referencia en el momento de probar el rendimiento de funcionamiento de la red neuronal creada.

Tabla 6. Valores de entradas y salidas del sistema.

SET POINT	Kp	Ki	Kd
2	37-38	0,1-0,6	173-175
3	38-39	0,6-1	175-176
4	39-40	1-1,6	177-178
5	40-41	1,5-2	179-180
6	41-42	2-2,5	181-182
7	42-43	2,5-3	183-184

Elaborado por: El Investigador

Para probar el rendimiento de la red una vez creado se exporta y con comandos se llama a la red neuronal para ponerlo en marcha, dando valores de nivel del tanque de manera que arroje a sus salidas los valores de kp, ki y kd óptimos acorde al nivel al cual se desea controlar. En la siguiente figura 56, se observa el resultado de la prueba realizada.

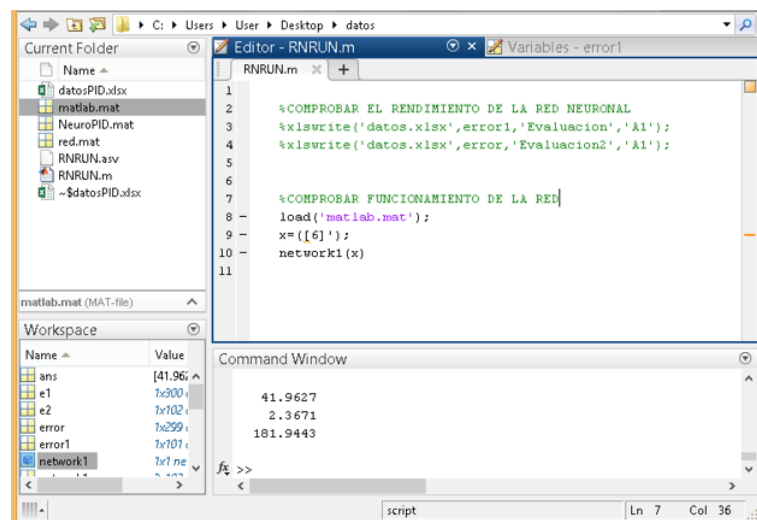


Figura 56. Prueba de funcionamiento de red neuronal

Como se puede apreciar se ingresó un valor de Setpoint de (6,4,5,7), la red neuronal creada automáticamente predijo y proporcionó los datos de control kp, ki y kd para ese nivel, basándose en la tabla 6, se compara los rangos de control de nivel con los valores dados por la red neuronal, demostrando que la eficiencia predicha es correcta en un 99,9%, como se observa en la tabla 7.

Tabla 7. Valores establecidos y valores arrojados por la red neuronal

Métodos	Set Point	Kp	Ki	Kd
Rangos establecidos	6	41-42	2-2,5	181-182
	4	39-40	1-1,6	177-178
	5	40-41	1,5-2	179-180
	7	42-43	2,5-3	183-184
Red Neuronal	6	41,9627	2,3671	181,9443
	4	39,8687	1,1301	177,9280
	5	40,4848	1,8806	180,3996
	7	42,8354	3,3597	184,4060

Elaborado por: El Investigador

De la misma manera, se comprobó ingresando el valor de kp, ki y kd, para saber la respuesta de la red neuronal, el cual identifica sin problema el nivel al cual se está realizando el control, como se observa en la figura 57, tomando valores aleatorios para su comprobación.

The screenshot shows a MATLAB script in the Editor window titled 'RNRUN.m'. The code includes a function call to 'NeuroPID(x)' with a vector of control parameters: `x = ([43.1687692 2.573248114 184.246165]');`. The Command Window displays the output 'ans = 7.0000'. To the right of the code editor, a table displays the results of the control process, with columns for time (row index), Kp, Ki, Kd, and the predicted level (7). The predicted level is consistently 7, matching the setpoint.

293	42,8384325	3,230743441	184,420356	7
294	38,0004942	0,213566083	174,144144	2
295	41,7341987	1,999380922	181,853027	6
296	42,883337	2,554396348	184,595683	7
297	38,165778	0,564048745	174,099967	2
298	43,1687692	2,573248114	184,246165	7
299	40,5304882	2,497368815	180,302222	5
300	37,701	0,20476	173,545	2
301	38,6792314	0,38942	174,305303	2
302	39,45439	1,23487	177,3937	4
303	42,8525	2,59247	183,46672	7
304	40,12683	1,57324	179,82765	5

Figura 57. Comprobación de rango de nivel de control

Para evaluar la eficiencia de la neurona en base a los errores y las salidas que genera se realiza el cálculo de errores mostrando la variabilidad del mismo, en este caso se

utilizará el método del error medio cuadrático para lo cual se hace uso de la siguiente ecuación matemática.

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

Donde:

y_i =resultado real esperado

\hat{y}_i =es la predicción del modelo

N =numero de datos

Para esto se pasan todos los datos generados por la red neuronal a una hoja de cálculo por medio de una línea de comando como se observa en la figura 58.

```

1
2 %COMPROBAR EL RENDIMIENTO DE LA RED NEURONAL
3 xlswrite('datosPID.xlsx', network1_errors, 'Evaluacion1', 'A1');
4 xlswrite('datos.xlsx', error, 'Evaluacion2', 'A1');
5

```

Figura 58. Código de comando para pasar los datos generados

Una vez obtenidos los valores se procede aplicar la fórmula elevando los datos a la potencia 2, sumando sus valores y dividiendo para el número de datos, finalmente para saber el porcentaje de error medio cuadrático de la red neuronal se multiplica por 100% dando como resultado lo siguiente:

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

$$MSE = 0,01228 * 100$$

$$MSE = 1,2280\%$$

Donde:

MSE =error medio cuadratico

N =número de muestras tomadas

y_i = es el resultado real esperado

\hat{y}_i =es la predicción del modelo

Dando un porcentaje de error cuadrático medio de 1,22% de la red neuronal pudiendo elegir una mejor configuración de la misma, concluyendo que es un porcentaje mínimo a errores de predicción para tener mejores resultados de respuesta más asertivos.

CAPÍTULO IV

CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

En esta investigación se ha presentado el diseño de un controlador NeuroPID en un módulo experimental de tanques de nivel, evaluado en eventos bajo la norma IEC-61499, para el control automatizado, realizando un estudio en la implementación de bloques funcionales de control y el uso de redes neuronales, así como el comportamiento de algoritmos frente a la respuesta del sistema de control, se estableció un valor inicial de las constantes de $k_p=3,7701$, $k_i=0,20476$ y $k_d=17,3545$, mediante la función de transferencia del sistema con un porcentaje de 87.38 % de estabilidad.

Se ha diseñado un controlador mediante la creación de un modelo de red neuronal multicapa, utilizando la función de transferencia sigmoideal como activación. El entrenamiento de la red neuronal se ha llevado a cabo mediante el algoritmo de retropropagación del error. Se ha evaluado el desempeño del controlador bajo las condiciones de perturbación constante y variante en el tiempo, utilizando una señal de referencia escalón unitario determinando que el control basado en redes neuronales es superior al control realizado mediante lógica difusa, según los criterios de error en estado estacionario y tiempo de establecimiento. Se han utilizado 1000 datos de entrenamiento, 300 datos de validación y 100 datos de prueba. El controlador basado en redes neuronales ha obtenido una precisión de predicción del 99%, con un error máximo absoluto del 0.01 y un tiempo de establecimiento de 0:00:49 segundos en condiciones de perturbación constante.

Se ha llevado a cabo una investigación experimental para evaluar la viabilidad de crear algoritmos de control basados en la norma IEC-61499, utilizando una plataforma que permite la interacción de pruebas adaptadas a las características de respuesta del sistema evaluado en bloques funcionales. El algoritmo desarrollado ha demostrado ser robusto ante cambios en la respuesta del sistema, al considerar la curva total de control y minimizar la integral del error, lo que lo hace menos sensible a los errores presentes en las perturbaciones. Los resultados obtenidos en las pruebas experimentales

muestran un valor de cálculo del 1,22% de error, lo que indica una alta precisión en la respuesta del algoritmo. Durante las pruebas se ha aplicado el cálculo de una función de transferencia de primer orden para evaluar la respuesta del sistema y asegurar la estabilidad del control.

4.2 Recomendaciones

En este proyecto de investigación se propuso implementar un controlador NeuroPID para un sistema de tanques, utilizando una red neuronal artificial en combinación con un controlador PID clásico para mejorar el rendimiento del sistema en términos de tiempo de ejecución. Se debe tomar en cuenta el desempeño del controlador mediante el análisis de las perturbaciones presentes en el sistema, estableciendo criterios para determinar cuáles perturbaciones son aceptables y cuáles deben ser rechazadas.

Se recomienda utilizar un algoritmo de entrenamiento basado en el método de retro propagación del error para ajustar los parámetros de la red neuronal evaluando el desempeño del controlador en términos de tiempo de respuesta, error de seguimiento y capacidad de rechazo, utilizando diferentes tipos de perturbaciones tales como cambios abruptos en la posición del setpoint, cambios en la carga del sistema, y perturbaciones aleatorias.

Es importante destacar que cada controlador debe ser diseñado específicamente para cumplir las necesidades requeridas y enfoques particulares del sistema en cuestión. Aunque el NeuroPID presentado en este proyecto es altamente eficiente en mejorar el desempeño del sistema, no se puede afirmar que sea el controlador más adecuado en todas las situaciones, ya que su eficacia depende de la naturaleza del sistema y las necesidades específicas de cada caso. Por lo tanto, se recomienda siempre realizar un estudio previo exhaustivo y modelar un controlador eficiente que cumpla con los objetivos específicos del sistema, con el fin de maximizar su eficiencia y rendimiento.

Bibliografía

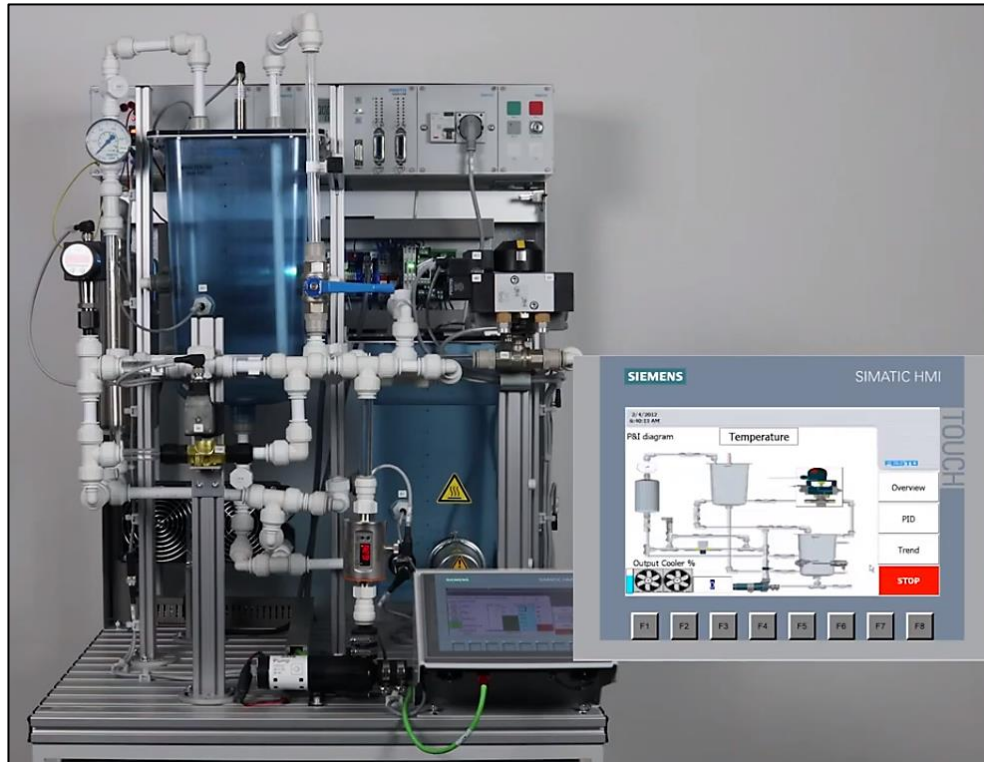
- [1] «Online Tuning of PID Controller Using a Multilayer Fuzzy Neural Network Design for Quadcopter Attitude Tracking Control - PMC». <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7847850/> (accedido 8 de diciembre de 2022).
- [2] J. Günther, E. Reichensdörfer, P. M. Pilarski, y K. Diepold, «Interpretable PID parameter tuning for control engineering using general dynamic neural networks: An extensive comparison», *PLOS ONE*, vol. 15, n.º 12, p. e0243320, dic. 2020, doi: 10.1371/journal.pone.0243320.
- [3] «(PDF) PID Tuning with Neural Networks». https://www.researchgate.net/publication/332152709_PID_Tuning_with_Neural_Networks (accedido 8 de diciembre de 2022).
- [4] «A New PID Neural Network Controller Design for Nonlinear Processes | Journal of Circuits, Systems and Computers». <https://www.worldscientific.com/doi/abs/10.1142/S0218126618500652> (accedido 8 de diciembre de 2022).
- [5] H. Malik, S. Srivastava, Y. R. Sood, y A. Ahmad, *Applications of Artificial Intelligence Techniques in Engineering: SIGMA 2018, Volume 1*. Springer, 2018.
- [6] «Innovation Summit World Tour - Schneider Electric Blog». <https://blog.se.com/tag/innovation-summit-world-tour/> (accedido 8 de diciembre de 2022).
- [7] «IEC 61499: standard for portability and Industry 4.0 | Industrial Ethernet Book». <https://iebmedia.com/technology/iiot/industrial-automation-standard-for-portability-and-industry-4-0/> (accedido 8 de diciembre de 2022).
- [8] «5 ways IEC 61499 is liberating Industrial Automation». <https://blog.se.com/industry/machine-and-process-management/2020/09/28/5-ways-iec-61499-is-liberating-industrial-automation/> (accedido 8 de diciembre de 2022).
- [9] E. Querol, J. A. Romero, A. M. Estruch, y F. Romero, «NORMA IEC-61499 PARA EL CONTROL DISTRIBUIDO. APLICACIÓN AL CNC.», p. 8, 2014.
- [10] «Norma Iec-61499 para El Control Distribuido | PDF | Algoritmos | Control numerico». <https://es.scribd.com/document/282660735/Norma-Iec-61499-Para-El-Control-Distribuido> (accedido 8 de diciembre de 2022).
- [11] M. V. García, F. Pérez, I. Calvo, y F. López, «Desarrollo de CPPS sobre IEC-61499 Basado en Dispositivos de Bajo Coste», 2015.
- [12] «[PDF] IEC 61499 vs. 61131: A Comparison Based on Misperceptions | Semantic Scholar». <https://www.semanticscholar.org/paper/IEC-61499-vs.->

- 61131%3A-A-Comparison-Based-on-Thramboulidis/afeb9999fd3848988b5bf54ea9a3c8bdfcfdc2b5 (accedido 12 de diciembre de 2022).
- [13] J. H. Christensen, T. Strasser, y A. Valentini, «The IEC 61499 Function Block Standard: Overview of the Second Edition», 2012.
- [14] G. J. Caiza Guanochanga, «Implementación de sistemas distribuidos de bajo costo bajo norma IEC-61499, en la estación de clasificación del MPS 500 de la Universidad Politécnica Salesiana.», jul. 2017, Accedido: 13 de diciembre de 2022. [En línea]. Disponible en: <http://dspace.espace.edu.ec/handle/123456789/6926>
- [15] «Desarrollo de un sistema de control en red mediante el estándar IEC-61499». <http://repositori.uji.es/xmlui/handle/10234/148085> (accedido 14 de diciembre de 2022).
- [16] «Modelling control systems using IEC 61499 applying function blocks to distributed systems / Robert Lewis. - University of Queensland». https://search.library.uq.edu.au/primo-explore/fulldisplay/61UQ_ALMA51271889560003131/61UQ (accedido 14 de diciembre de 2022).
- [17] «IDE 4diac». https://www.eclipse.org/4diac/en_ide.php (accedido 15 de diciembre de 2022).
- [18] «Implementation of Low Cost Distributed System Under IEC-61499 Standard, in Classification and Handling Station of MPS-500 | Ingenius». <https://ingenius.ups.edu.ec/index.php/ingenius/article/view/18.2017.05> (accedido 16 de diciembre de 2022).
- [19] C. A. M. Espitia y J. L. M. Bermudez, «IDENTIFICACIÓN DE UN SISTEMA DE CONTROL DE NIVEL Y FLUJO UTILIZANDO REDES NEURONALES ARTIFICIALES».
- [20] S. P. Banks, *Signal Processing, Image Processing and Pattern Recognition*, 1st Edition. New York: Prentice Hall, 1991.
- [21] J. J. M. Moreno, «Redes Neuronales Artificiales aplicadas al Análisis de Datos».
- [22] Y. Mejías César, R. Carrasco Velar, I. Ochoa Izquierdo, y E. Moreno Lemus, «Funciones de transferencia en el perceptrón multicapa: efecto de su combinación en entrenamiento local y distribuido», *Rev. Cuba. Informática Médica*, vol. 5, n.º 2, pp. 186-199, dic. 2013.
- [23] M. F. Moreno, «Introducción a las Redes Neuronales Artificiales».
- [24] V. Goyal, M. Gupta, A. Trivedi, y M. L. Kolhe, Eds., *Proceedings of International Conference on Communication and Artificial Intelligence: ICCAI 2020*, vol. 192. Singapore: Springer Singapore, 2021. doi: 10.1007/978-981-33-6546-9.

- [25] A. Marino y F. Neri, «PID Tuning with Neural Networks», 2019, pp. 476-487. doi: 10.1007/978-3-030-14799-0_41.
- [26] C. A. García, E. X. Castellanos, y M. V. García, «UML-Based Cyber-Physical Production Systems on Low-Cost Devices under IEC-61499», *Machines*, vol. 6, n.º 2, Art. n.º 2, jun. 2018, doi: 10.3390/machines6020022.
- [27] «4diac FORTE - El entorno de ejecución de 4diac». https://www.eclipse.org/4diac/en_rte.php (accedido 27 de diciembre de 2022).
- [28] Enzyme, «Redes neuronales con Python: ¿por qué es el mejor lenguaje para IA?» <https://enzyme.biz/blog/redes-neuronales-python> (accedido 27 de diciembre de 2022).
- [29] K. S. Narendra y K. Parthasarathy, «Gradient methods for the optimization of dynamical systems containing neural networks», *IEEE Trans. Neural Netw.*, vol. 2, n.º 2, pp. 252-262, mar. 1991, doi: 10.1109/72.80336.

ANEXOS

Anexo A. Módulo de nivel de tanques FESTO® MPS-PA, para la implementación del sistema de control.



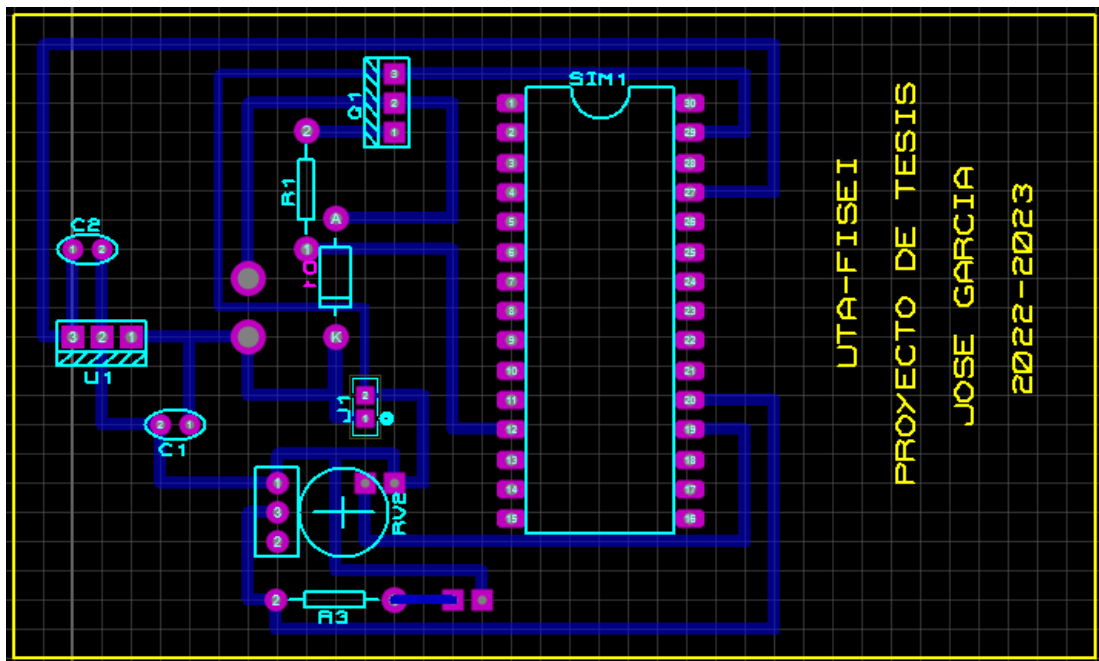
Anexo B. Modulo operable en el proceso de control de nivel



Anexo C. Funcionamiento de la placa en el modulo

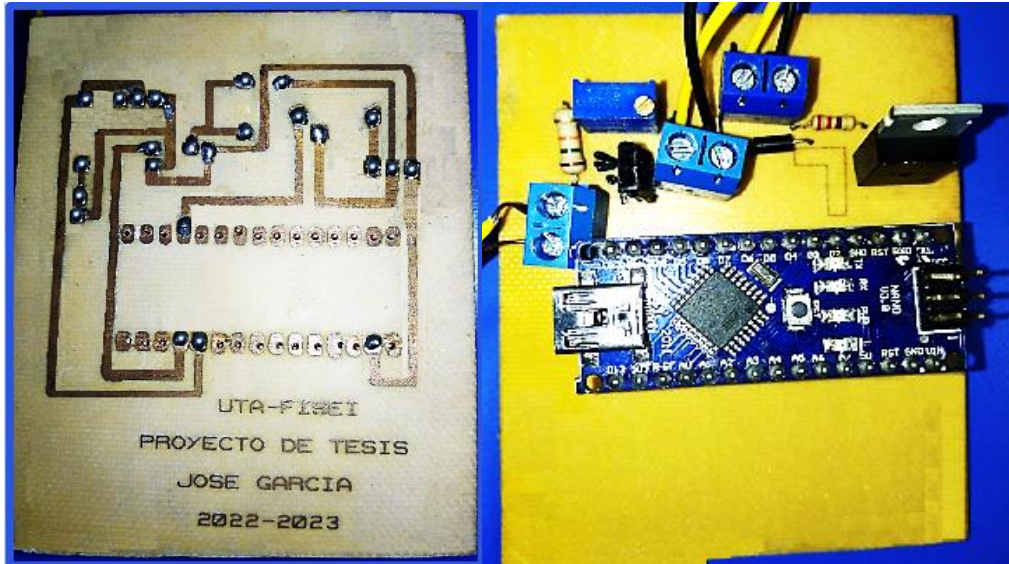


Anexo C. Esquema PCB del circuito aplicado al sistema de control



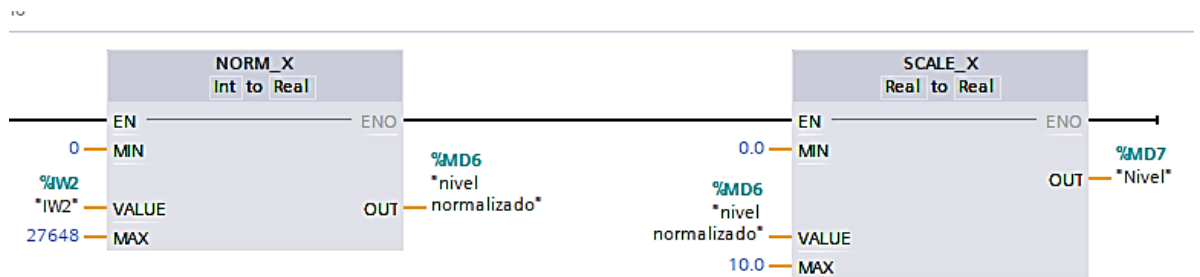
Elaborado por: El Investigador

Anexo D. Circuito de acoplamiento ensamblado en la placa



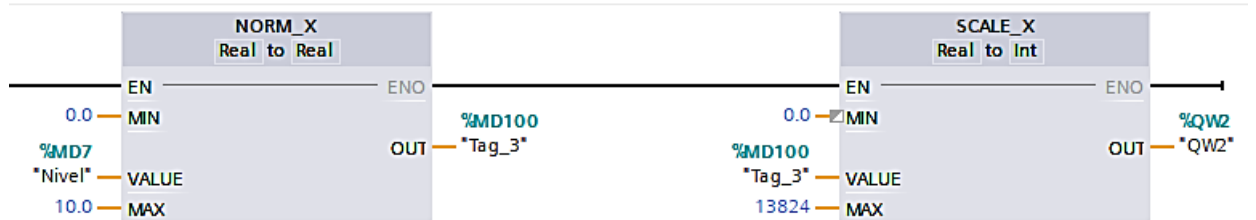
Elaborado por: El Investigador

Anexo E. Circuito Lectura del nivel del Tanque



Elaborado por: El Investigador

Anexo F. Salida analógica para los valores de PWM.



Elaborado por: El Investigador

Anexo G. Código de la NN basado en Keras para la predicción de Kp Ki Kd

```
1 # Importar las librerías necesarias
2 import numpy as np
3 from keras.models import Sequential
4 from keras.layers import Dense
5
6 # Defina las constantes
7 N_INPUTS = 5 # Número de entradas (kp, ki, kd, nivel del tanque, entrada de líquido)
8 N_OUTPUTS = 3 # Número de salidas (kp, ki, kd)
9 N_SAMPLES = 1000 # Número de muestras de entrenamiento
10
11 # Generar datos de entrenamiento
12 X_train = np.random.random((N_SAMPLES, N_INPUTS))
13 y_train = np.random.random((N_SAMPLES, N_OUTPUTS))
14
15 # Crear el modelo de red neuronal
16 model = Sequential()
17 model.add(Dense(64, input_dim=N_INPUTS, activation='relu'))
18 model.add(Dense(32, activation='relu'))
19 model.add(Dense(N_OUTPUTS, activation='linear'))
20
21 # Compilar el modelo
22 model.compile(loss='mean_squared_error', optimizer='adam')
23
24 # Entrenar el modelo
25 model.fit(X_train, y_train, epochs=10, batch_size=32)
26
27 # Hacer predicciones
28 X_test = np.random.random((1, N_INPUTS))
29 y_pred = model.predict(X_test)
30 print(y_pred)
```

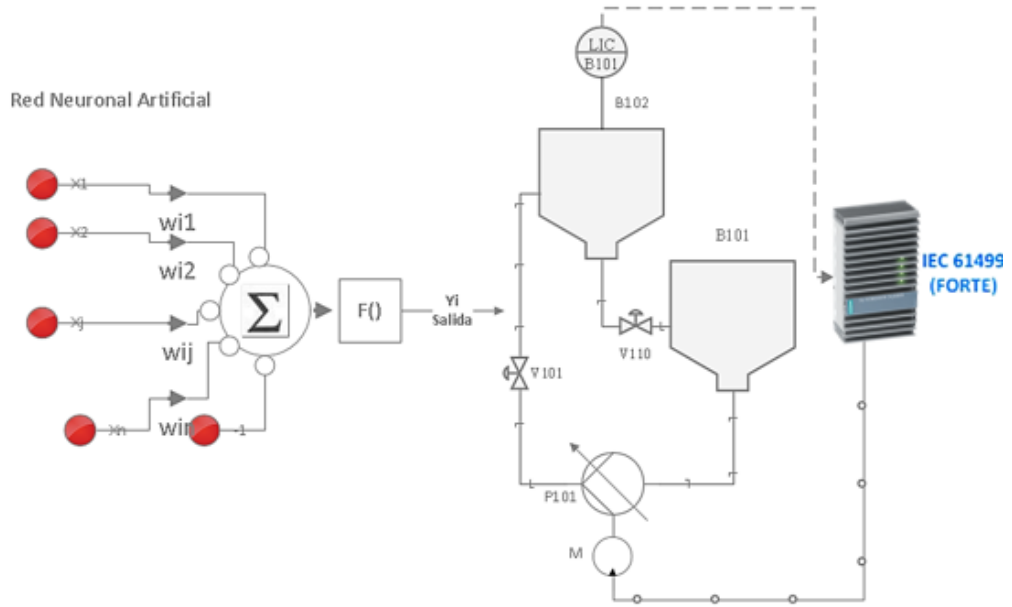
Elaborado por: El Investigador

Anexo I. Respuesta al Impulso escalón por la plataforma Arduino Ide.



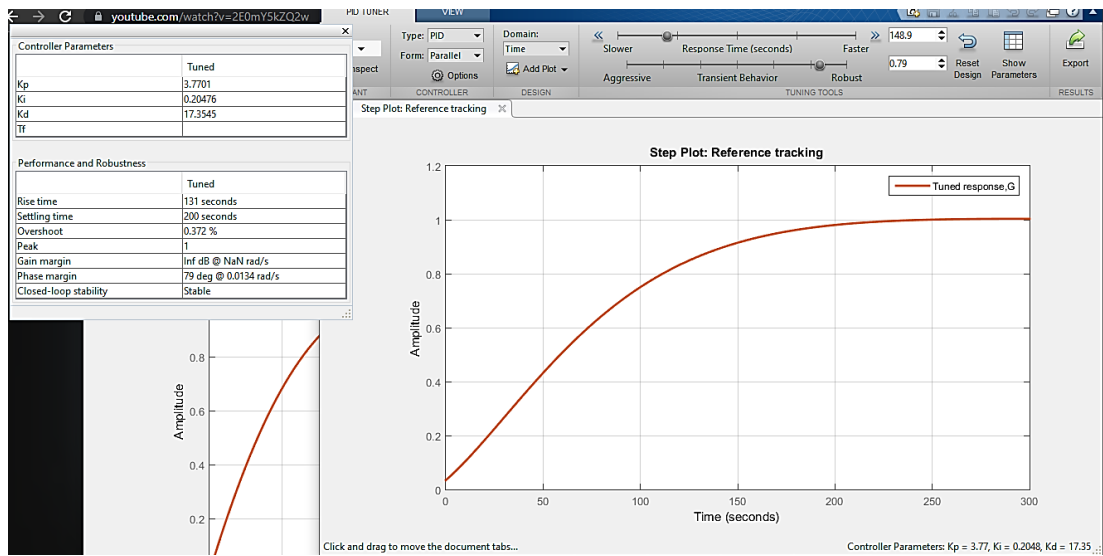
Elaborado por: El Investigador

Anexo J. Diagrama PI&D de flujo de nivel



Elaborado por: El Investigador

Anexo K. Grafica de respuesta mediante las constantes PID



Elaborado por: El Investigador