

**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO**

**CARRERA:
INGENIERÍA ELECTRÓNICA**

**Trabajo de titulación previo a la obtención del título de:
INGENIEROS ELECTRÓNICOS**

**TEMA:
DESARROLLO DE UN CONTROLADOR GENÉTICO DIFUSO PARA EL
MÓDULO DE NIVEL DEL LABORATORIO DE TEORÍA DE CONTROL DE
LA UPS**

**AUTORES:
MARLON ANDRÉS POTOSÍ ARROYO
GRACE FABIOLA RUIZ RIVERA**

**TUTOR:
HAMILTON LEONARDO NÚÑEZ VERDEZOTO**

Quito, octubre del 2020

CESIÓN DE DERECHOS DE AUTOR

Nosotros Marlon Andrés Potosí Arroyo y Grace Fabiola Ruiz Rivera con documento de identificación N.º 1002781902 y N.º 1725510570 respectivamente, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación intitulado: “DESARROLLO DE UN CONTROLADOR GENÉTICO DIFUSO PARA EL MÓDULO DE NIVEL DEL LABORATORIO DE TEORÍA DE CONTROL DE LA UPS”, mismo que ha sido desarrollado para optar por el título de Ingenieros Electrónicos, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribimos este documento en el momento que hacemos entrega del trabajo final en digital a la Biblioteca de la Universidad Politécnica Salesiana.



Marlon Andrés Potosí Arroyo

C.I. 1002781902



Grace Fabiola Ruiz Rivera

C.I. 1725510570

Quito, octubre del 2020

DECLARATORIA DE COAUTORÍA DEL DOCENTE TUTOR

Yo declaro que bajo mi dirección y asesoría fue desarrollado el Proyecto Técnico, “DESARROLLO DE UN CONTROLADOR GENÉTICO DIFUSO PARA EL MÓDULO DE NIVEL DEL LABORATORIO DE TEORÍA DE CONTROL DE LA UPS”, realizado por Marlon Andrés Potosí Arroyo y Grace Fabiola Ruiz Rivera, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerados como trabajo final de titulación.

Quito, octubre del 2020



Hamilton Leonardo Núñez Verdezoto

C.I. 1712981487

DEDICATORIA

Quiero dedicar esta tesis a mi esposa Dianita por su gran apoyo en estos años de convivencia, a mis padres Cecilia y Patricio quienes me han inculcado buenos valores que me han ayudado en toda la vida universitaria, a mis hermanos que siempre han estado en todas mis situaciones adversas apoyándome y escuchándome.

Finalmente, a mi pequeña Alejandrina quien fue la pauta principal para culminar con mis estudios y darle un ejemplo de perseverancia, porque el camino es largo, pero no imposible de seguir y terminar.

Marlon

En primer lugar, quiero dedicar este trabajo a Dios y segundo a mis padres Juan y Marlene quienes han sido el pilar fundamental para llegar a culminar mis estudios y que me han dado bases de responsabilidad y deseos de superación, a mis hermanos John y Sebastián que me han brindado su apoyo incondicional y finalmente a mi hijo Jhoel quien ha sido el motor y la motivación en este proceso.

Grace

AGRADECIMIENTO

Agradezco a mi esposa y a mis padres, quienes nunca decayeron en su afán de apoyarme en todo este ciclo, quienes día a día me dieron ánimos con los cuales tuve la perseverancia para terminar este gran camino, agradezco a mis suegros Don Elías quien desde un inicio nos dio su apoyo incondicional para obtener este título, a mi suegra Sra. Carmita (+) quien por medio de sus palabras nos enseñó que los valores morales de una persona es la primera carta de presentación de esta.

Agradezco a todos mis amigos que hice durante este transcurso, y principalmente a la Universidad Politécnica Salesiana, quien fue nuestro segundo hogar durante estos cinco años, y en la cual culmino un escalón más de la vida.

Marlon

Agradezco a la Universidad Politécnica Salesiana y a su cuerpo docente por haberme formado profesionalmente en este tiempo, en especial a mi tutor por su enorme colaboración para la culminación de mi proyecto de titulación.

Agradezco a mis padres por sus consejos, por todo el apoyo que me han brindado, por su amor y por haberme formado como una mujer de bien, también agradezco a William mi compañero de vida por su apoyo y por darme ánimos de seguir adelante y llegar a culminar una meta más.

Grace

ÍNDICE DE CONTENIDO

CESIÓN DE DERECHOS DE AUTOR.....	ii
DECLARATORIA DE COAUTORÍA DEL DOCENTE TUTOR.....	iii
DEDICATORIA	iv
AGRADECIMIENTO.....	v
ÍNDICE DE CONTENIDO.....	vi
ÍNDICE DE FIGURAS.....	ix
ÍNDICE DE TABLAS	xi
RESUMEN.....	xii
ABSTRACT.....	xiii
INTRODUCCIÓN	xiv
CAPÍTULO 1	1
ANTECEDENTES.....	1
1.1 PLANTEAMIENTO DE PROBLEMA	1
1.2 JUSTIFICACIÓN.....	1
1.3 PROPUESTA DE SOLUCIÓN.....	2
1.4 OBJETIVOS.....	3
1.4.1 Objetivo General.....	3
1.4.2 Objetivos Específicos	3
1.5 METODOLOGÍA.....	3
CAPÍTULO 2.....	4
MARCO CONCEPTUAL	4
2.1 SISTEMA DE CONTROL REALIMENTADO (SCR).....	4
2.2 LÓGICA DIFUSA (LD).....	4
2.2.1 Teoría de conjuntos Difusos.....	4
2.2.2 Tipos de funciones de pertenencia.....	5
2.2.3 Conjuntos difusos	6
2.2.4 Reglas difusas	7
2.3 CONTROL DE LÓGICA DIFUSA (CLD).....	8
2.3.1 Fuzzificador	9

2.3.2 Inferencia y Base de reglas	9
2.3.3 Defuzzificador	9
2.4 ALGORITMOS GENÉTICOS (AG)	10
2.4.1 Estructura de los algoritmos genéticos	10
2.5 ÍNDICES DE DESEMPEÑO	13
2.5.1 Integral del error absoluto (IAE)	14
2.6 TEST ESTADÍSTICO DE WILCOXON.....	14
CAPÍTULO 3	15
DISEÑO E IMPLEMENTACIÓN.....	15
3.1 DESCRIPCIÓN DEL HARDWARE	15
3.1.1 Bomba Centrífuga.....	15
3.1.2 Pantalla Táctil	16
3.1.3 Servo Válvula	17
3.1.4 Sensor de presión diferencial (SPD).....	18
3.1.5 Tanque graduado.....	19
3.1.6 Tanque de reserva	20
3.1.7 Variador de frecuencia.....	21
3.2 DISEÑO DE LA TARJETA DE CONTROL	22
3.2.1 Controlador	23
3.2.2 Conversor de Voltaje DC-DC.....	24
3.2.3 Conexiones y acondicionamiento de señales en el microcontrolador	24
3.2.4 Desarrollo del layout de la tarjeta	26
3.3 DISEÑO DEL ALGORITMO DIFUSO Y OPTIMIZACIÓN	27
3.3.1 Identificación del sistema	28
3.3.2 Controlador Difuso	29
3.3.3 Software de optimización genético difuso.....	32
3.3.4 Controlador Genético Difuso.....	34
3.4 DESCRIPCIÓN DEL SOFTWARE DE AUTOMATIZACIÓN.....	36
3.4.1 IDE Controlador Arduino Mega.....	36
3.4.2 Edición de la pantalla.....	37
3.4.3 Interfaz Hombre—Máquina (HMI)	37
CAPÍTULO 4	39
PRUEBAS Y RESULTADOS	39
4.1 PRUEBA DEL CONTROLADOR FUZZY CONVENCIONAL.....	39

4.2	PRUEBAS DEL CONTROLADOR GENÉTICO DIFUSO (CGD).....	41
4.3	ANÁLISIS COMPARATIVO DE CONTROLADORES	44
4.4	TEST DE WILCOXON: CF VS. CGD	47
	CONCLUSIONES	50
	RECOMENDACIONES	51
	REFERENCIAS	52
	ANEXOS	55

ÍNDICE DE FIGURAS

Figura 2.1 Sistema de control en lazo cerrado	4
Figura 2.2 Función de pertenencia triangular	6
Figura 2.3 Estructura de un conjunto difuso	6
Figura 2.4 Evaluación de reglas por FAM	7
Figura 2.5 Sistema de control difuso realimentado	8
Figura 2.6 Sistema de control lógica difusa	9
Figura 2.7 Etapa de Cruzamiento	12
Figura 2.8 Etapa de Mutación	12
Figura 2.9 Diagrama de flujo del AG	13
Figura 3.1 Bomba centrífuga	15
Figura 3.2 Pantalla Táctil Nextion	16
Figura 3.3 Servo Válvula	17
Figura 3.4 Sensor de presión diferencial	18
Figura 3.5 Sensor de presión instalado	19
Figura 3.6 Tanque graduado	20
Figura 3.7 Tanque de reserva	21
Figura 3.8 Variador de frecuencia	22
Figura 3.9 Controlador	23
Figura 3.10 Diseño circuito placa de control	26
Figura 3.11 Diseño placa del controlador	27
Figura 3.12 Gráfica de prueba de relación lineal	27
Figura 3.13 Gráfica de la señal pseudoaleatoria de tipo escalón y respuesta del sistema	28
Figura 3.14 Diagrama de flujo del control difuso	30
Figura 3.15 Variable de entrada Nivel	31
Figura 3.16 Variable de entrada Válvula	31
Figura 3.17 Variable de salida Motor	31
Figura 3.18 Selección de la Población AG – Matlab	32
Figura 3.19 Función Fitness	32
Figura 3.20 Parámetros de optimización – Matlab	33
Figura 3.21 Optimización Población Nivel – Matlab	34

Figura 3.22 Control Genético Difuso Entrada Nivel – Matlab	34
Figura 3.23 Control Genético Difuso Entrada Válvula – Matlab	35
Figura 3.24 Diagrama de flujo del Control Genético Difuso	35
Figura 3.25 IDE controlador	36
Figura 3.26 Nextion Editor	37
Figura 3.27 Pantalla 1 HMI	37
Figura 3.28 Pantalla 2 HMI	38
Figura 4.1 Comportamiento de la planta con el CF	39
Figura 4.2 Tool box de Matlab de Fuzzy ingreso de datos en las variables	40
Figura 4.3 CF en Simulink para obtención del IAE	40
Figura 4.4 Grupo de condiciones generado mediante la optimización por AG	41
Figura 4.5 Comportamiento de la planta con el CGD	41
Figura 4.6 Tool box de Fuzzy Matlab Prueba 1 CGD	42
Figura 4.7 Carga de archivo Prueba 1 en simulink	42
Figura 4.8 Prueba 1 en Simulink para obtención del IAE	43
Figura 4.9 Tool box de Fuzzy Matlab Prueba 2 CGD	43
Figura 4.10 Prueba 2 en Simulink para obtención del IAE	44
Figura 4.11 Comportamiento del los controladores estudiados.....	44
Figura 4.12 Generación de 30 pruebas mediante optimización de AG	45

ÍNDICE DE TABLAS

Tabla 2.1 Evaluación de reglas usando “IF-THEN”	8
Tabla 3.1 Especificaciones técnicas Bomba Centrífuga	16
Tabla 3.2 Características Pantalla táctil Nextion	17
Tabla 3.3 Características Servo Válvula	18
Tabla 3.4 Características del sensor de presión diferencial	19
Tabla 3.5 Características tanque graduado	20
Tabla 3.6 Características Tanque de reserva	21
Tabla 3.7 Características variador de frecuencia	22
Tabla 3.8 Características Controlador	23
Tabla 3.9 Conversor DC-DC	24
Tabla 3.10 Conexiones Microcontrolador	24
Tabla 3.11 Variables Lingüísticas	29
Tabla 3.12 Reglas Difusas	30
Tabla 4.1 Datos IAE del CF y datos IAE del CGD	46
Tabla 4.2 Test de Wilcoxon	48
Tabla 4.3 Decisión de validez	49

RESUMEN

El módulo de control de nivel de líquidos pertenece al laboratorio de teoría de control de la Universidad Politécnica Salesiana el mismo que es utilizado para capacitar a los alumnos en la explicación y orientación de técnicas de control. En el presente proyecto se optimizó el módulo tanto en software como en hardware, realizando la implementación de un nuevo algoritmo de control denominado Genético Difuso e instalando un nuevo sensor de nivel y una nueva pantalla HMI para el control de mismo respectivamente. Se empezó realizando una caracterización del módulo de nivel a utilizarse para conocer su dinámica y verificar su funcionamiento. Se planteó y planificó la mejor arquitectura de control tomando en cuenta los elementos presentes en el lazo existente. Se realizó el diseño del nuevo controlador difuso en base a un sistema de múltiples entradas y de una única salida (MISO de sus siglas en inglés Multiple Inputs Single Output). Se planteó la estrategia de control óptima que interactuará entre los algoritmos difuso y genético en forma paralela. Se establece a la vez la función de costo (o función fitness) que evaluará los mejores conjuntos de funciones de membresía difusas que empleará el control. Finalmente, se implementó el nuevo controlador Genético Difuso en el control de nivel de la planta realizando pruebas de funcionamiento y validándolas estadísticamente empleando índices de desempeño.

ABSTRACT

The liquid level control module belongs to the control theory laboratory of the Salesian Polytechnic University, which is used to train students in the explanation and orientation of control techniques. In the present project, the module was optimized in both software and hardware, testing the implementation of a new control algorithm called Diffuse Genetics and installing a new level sensor and a new HMI screen to control it, respectively. He began to carry out a characterization of the level module to detect its dynamics and verify its operation. The best control architecture was proposed and planned taking into account the elements present in the existing loop. The design of the new diffuse controller was made based on a multiple input and single output system (MISO from its initials in English Multiple inputs Single output). The optimal control strategy that will interact between the diffuse and genetic algorithms in parallel was proposed. At the same time, the cost function (or fitness function) is established that evaluates the best fuzzy membership function sets that the control will use. Finally, the new Diffuse Genetic Controller will be implemented in the control of the plant level, perform performance tests and statistically validate using performance indices.

INTRODUCCIÓN

Este proyecto de titulación tiene como objetivo desarrollar un nuevo algoritmo de control dentro del módulo de nivel del laboratorio de control de la UPS ubicada en el sur de Quito.

El primer capítulo detalla los antecedentes del proyecto, la problemática y todas sus variables que se va a solucionar, así como la propuesta de solución con sus 5 etapas, los objetivos a ser alcanzados dentro del proyecto y finalmente cual es el camino o metodología que vamos a usar para culminar con el proyecto.

En el segundo capítulo se analiza la fundamentación teórica acerca de los dos algoritmos de control que van a ser validados, tanto el control difuso como el genético difuso, así como el algoritmo para optimizar y convertir de control difuso a genético difuso.

Dentro del tercer capítulo se detalla de forma breve las características de cada uno de los elementos a formar parte de esta implementación. Se presenta la metodología y el desarrollo de los algoritmos de control implementados en módulo de control.

En el cuarto capítulo se muestran las pruebas y los análisis realizados en el módulo al haber implementado el algoritmo de control, así como su validación y comparación con el control difuso clásico.

Finalmente, en el quinto capítulo se establecen las conclusiones y recomendaciones del presente proyecto de acuerdo con todos y cada uno de los objetivos planteados.

CAPÍTULO 1

ANTECEDENTES

1.1 PLANTEAMIENTO DE PROBLEMA

El módulo de control de nivel del laboratorio de teoría de control de la UPS fue desarrollado con el objeto de ensayar en él estrategias de control clásicas y modernas como control PID y difuso respectivamente, sin embargo se observó que su comportamiento no fue satisfactorio con el controlador difuso, teniendo como conclusiones que su algoritmo no fue sintonizado apropiadamente, además de que existe presencia de fallas en su proceso de manufactura (Espinoza, 2019).

Debido a esto, el módulo necesita ser repotenciado en forma física para la implementación de controladores inteligentes para validar nuevos algoritmos de control.

1.2 JUSTIFICACIÓN

Varias marcas de autómatas programables como Siemens (SIMATIC , 2019), Mitsubishi (MELSEC, 2019) y Omron (SYSMAC, 2019) ya ofrecen al mercado industrial controladores difusos, demostrando su eficacia alrededor del mundo, por lo tanto, es necesario comenzar a desarrollar controladores de este tipo y variaciones del mismo para analizar las ventajas y desventajas de la utilización de técnicas de control moderno para el diseño e implementación de sistemas de control. (Lopez, 2014)

Al momento de emplear algoritmos genéticos para la sintonización dentro de un algoritmo difuso, mejora el control, ya que el algoritmo cuenta con un conjunto inicial de soluciones (genes), los cuales van evolucionando en torno a la respuesta más óptima, tomando como base la teoría de Darwin sobre la selección natural (Maldonado, 2013).

1.3 PROPUESTA DE SOLUCIÓN

Para el desarrollo de los objetivos propuestos en esta investigación se plantean cuatro etapas las cuales permiten comprender las acciones que se efectuarán para el progreso de los mismos.

Etapa I: Caracterización del módulo de nivel del laboratorio de teoría de control.

En esta fase se efectuarán las siguientes actividades:

- 1) Recolección de la información del hardware del módulo de nivel a utilizarse.
- 2) Identificación del sistema para conocer la dinámica.

Etapa II: Repotenciación del sistema para la evaluación de nuevos controladores.

En esta fase se efectuarán las siguientes actividades:

- 1) Sustitución o adición de algunos elementos de la planta.
- 2) Verificación de funcionamiento de la planta con los nuevos elementos.

Etapa III: Identificación de la estructura del algoritmo genético difuso idóneo.

En esta fase se efectuarán las siguientes actividades:

- 1) Investigación de diferentes estructuras para el algoritmo genético difuso.
- 2) Modificación de la estructura del algoritmo para la implementación.

Etapa IV: Implementación del controlador genético difuso para el funcionamiento del módulo de control de nivel.

En esta fase se efectuará la siguiente actividad:

- 1) Pruebas de funcionamiento del algoritmo en la planta establecida.

Etapa V: Comparación del algoritmo genético difuso con el difuso convencional para la validación del algoritmo desarrollado.

En esta fase se efectuarán las siguientes actividades:

- 1) Empleo de índices de desempeño de controladores.
- 2) Empleo de test estadísticos.

1.4 OBJETIVOS

1.4.1 Objetivo General

Desarrollar un controlador genético difuso empleando herramientas computacionales para el ensayo de nuevas estrategias de control aplicables a procesos de nivel.

1.4.2 Objetivos Específicos

- Caracterizar el módulo didáctico de nivel del laboratorio de electrónica mediante manuales y diagramas técnicos para la comprensión de la dinámica de la planta.
- Repotenciar el sistema sustituyendo algunos elementos de la planta para la obtención de un mejor rendimiento.
- Identificar la estructura del algoritmo genético difuso idóneo utilizando referencias técnicas y textos especializados para usarlo en el desarrollo del controlador.
- Implementar el controlador genético difuso empleando un sistema embebido para el funcionamiento del módulo de control de nivel.
- Comparar el algoritmo genético difuso con el difuso convencional mediante índices de desempeño para la validación del algoritmo desarrollado.

1.5 METODOLOGÍA

Para la caracterización y repotenciación del módulo de nivel se empleará el método inductivo mediante el cual se valorará el estado funcional del sistema.

En la identificación de la estructura del algoritmo genético difuso se utilizará el método analítico mediante el cual se estudiará en forma intensiva cada uno de los elementos y las relaciones entre sí para poder desarrollar el algoritmo de control de la planta.

La implementación del controlador se realizará mediante el método práctico y para el análisis de pruebas y resultados se empleará el método hipotético inductivo el cual permitirá el análisis del respectivo funcionamiento.

CAPÍTULO 2

MARCO CONCEPTUAL

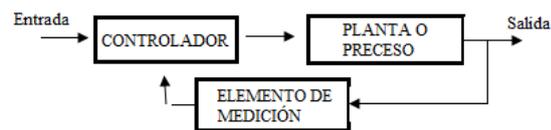
2.1 SISTEMA DE CONTROL REALIMENTADO (SCR)

Los SCR se denominan también sistemas de control en lazo cerrado (Ogata, 2003), es aquel que mantiene una relación entre las señales de entrada y de salida (o señal de referencia) (Vásquez Cortés, Cardona Guio, & Leal Gómez, 2015).

En los SCR, las señales de salida y de referencia son comparadas y la diferencia entre estas permite obtener una señal la cual se denomina señal de error del sistema. Una vez generada la señal de error, el controlador o regulador recibe dicha señal para que esta se ejecute en la planta con el fin de reducir el error, y así llevar la salida del sistema al valor deseado (Valdivia Miranda, 2012).

En la Figura 2.1 se muestra un esquema de un sistema de control en lazo cerrado o realimentado mostrando así los elementos principales del mismo.

Figura 2.1 Sistema de control en lazo cerrado



Sistema de control en lazo cerrado, Fuente: (Vásquez Cortés, Cardona Guio, & Leal Gómez, 2015)

2.2 LÓGICA DIFUSA (LD)

La LD es una técnica de inteligencia artificial determinista que permite tomar decisiones. Permite, también, obtener un comportamiento coherente y reproducible en función de las reglas que se le proveen (Mathivet, 2018).

2.2.1 Teoría de conjuntos Difusos

Los conjuntos clásicos suelen darse de forma natural y su función de pertenencia se limita a tomar valores exactos entre 0 ó 1; al contrario, en la teoría de conjuntos

difusos, la restricción anterior es más flexible por lo que la función de pertenencia puede adquirir valores del intervalo $[0,1]$ (Córdova, Cano, Dorado, & Herrera, 2016).

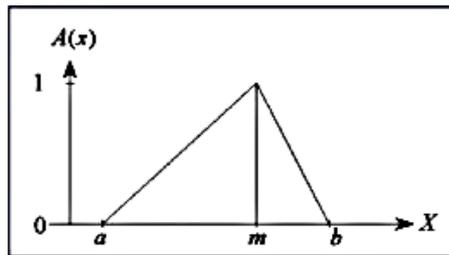
Por lo tanto, se considera importante dar a conocer las siguientes definiciones para la comprensión de la teoría de conjuntos difusos:

- **Variable lingüística:** Está hecha por palabras, sentencias o lenguaje natural. Esto constituye un modo para aproximar la caracterización de fenómenos complejos o no completamente definidos, por ejemplo “EDAD” (Armendáriz, y otros, 2014).
- **Universo de discurso:** Es el dominio (o rango) de valores que adquieren los elementos que fueron expresados en las variables lingüísticas. (Córdova, Cano, Dorado, & Herrera, 2016).
- **Valor lingüístico:** Es la o las etiquetas o términos que se les asigna a las variables lingüísticas (e.g., frío, caliente, tibio) (Córdova, Cano, Dorado, & Herrera, 2016).
- **Función de pertenencia:** La función de pertenencia es una curva que determina cual es el grado de dominio de un elemento. Esta curva define de qué manera cada punto del espacio de entrada se mapea para dar lugar a un grado de pertenencia bien definido (Gómez Ramirez, Cortés Ríos, Ortiz de la Vega, Melin Olmeda, & Castillo López, 2017).

2.2.2 Tipos de funciones de pertenencia

La construcción de las funciones de pertenencia depende del contexto al que se refiera, del concepto a definir y de la aplicación en general (Gómez Ramirez, Cortés Ríos, Ortiz de la Vega, Melin Olmeda, & Castillo López, 2017). Existen, no obstante, diferentes funciones clásicas (Mathivet, 2018). Las comúnmente usadas son: función triangular, función trapezoidal, función de pertenencia Gaussiana y función sigmoide. En este proyecto se utiliza la función triangular la cual viene delimitada por dos términos a y b , inferior y superior respectivamente, y un valor m de tal modo que $a < m < b$, así como se observa en la Figura 2.2.

Figura 2.2 Función de pertenencia triangular



Función de pertenencia de tipo triangular, Fuente: (Gómez Ramirez, Cortés Ríos, Ortiz de la Vega, Melin Olmeda, & Castillo López, 2017)

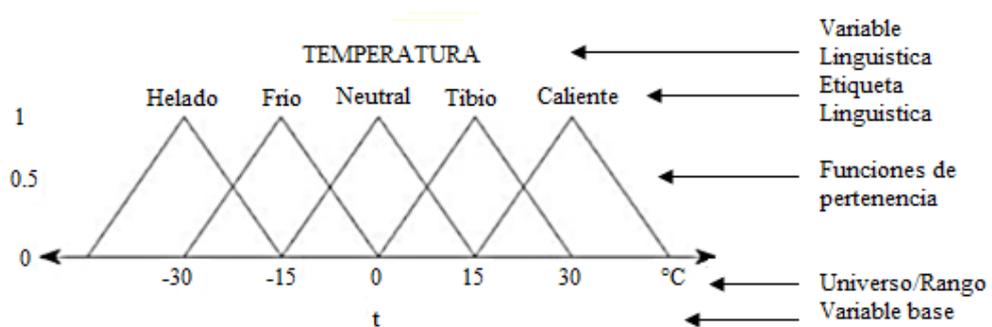
2.2.3 Conjuntos difusos

Los conjuntos difusos están compuestos por el valor lingüístico, la función de pertenencia y el universo de discurso (Córdova, Cano, Dorado, & Herrera, 2016).

Los conjuntos difusos mantienen ciertas características esenciales, cada una de estas características convierten su función de pertenencia en propia, es decir, que la función de pertenencia define por completo al conjunto (Gómez Ramirez, Cortés Ríos, Ortiz de la Vega, Melin Olmeda, & Castillo López, 2017).

En la Figura 2.3 se detalla los elementos de un conjunto difuso en el caso del ejemplo mostrando una variable lingüística de temperatura.

Figura 2.3 Estructura de un conjunto difuso



Estructura de la variable lingüística temperatura, Fuente: (Gómez Ramirez, Cortés Ríos, Ortiz de la Vega, Melin Olmeda, & Castillo López, 2017)

2.2.4 Reglas difusas

Kosko interpreta una regla difusa como una asociación (Nedjah & Mourelle, 2005) en su forma más simple como “if-then” / “si-entonces”, para lo cual se muestra la siguiente analogía.

Si x es A entonces y es B

Donde:

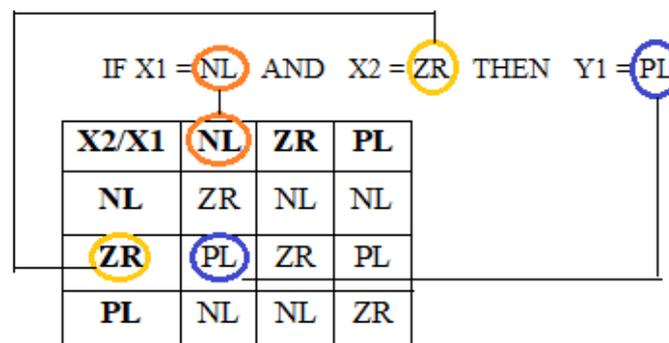
A y B son valores lingüísticos

x es la variable IN

y es la variable OUT

La evaluación de reglas difusas se puede realizar utilizando la tabla de matriz asociativa difusa (FAM) la cual es muy útil para representar el editor de reglas en forma de matriz que muestra todas las salidas posibles para todas las entradas posibles como se detalla en la Figura 2.4, también se puede representar las reglas empleando “IF-THEN” como se indica en la Tabla 2.1 (Baldovino & Dadios, 2015).

Figura 2.4 Evaluación de reglas por FAM



Evaluación de reglas por la matriz FAM, Fuente: (Baldovino & Dadios, 2015).

Tabla 2.1 Evaluación de reglas usando “IF-THEN”

1	SI X1 = ZR AND X2 = NL ENTONCES Y1 = NL
2	SI X1 = ZR AND X2 = ZR ENTONCES Y1 = ZR
3	SI X1 = PL AND X2 = NL ENTONCES Y1 = PL
4	SI X1 = PL AND X2 = ZR ENTONCES Y1 = PL

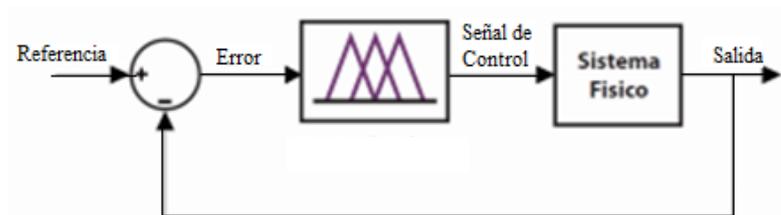
Evaluación de reglas usando “IF-THEN”, Fuente: (Baldovino & Dadios, 2015)

2.3 CONTROL DE LÓGICA DIFUSA (CLD)

La particularidad esencial de la LD es que admite la utilización del lenguaje común como lenguaje de descripción en un ordenador, de tal modo que permite integrar el conocimiento previo de un experto en una tarea determinada (Santos & Miranda, 2012).

En la Figura 2.5 se aprecia un sistema de CLD realimentado con sus respectivos elementos.

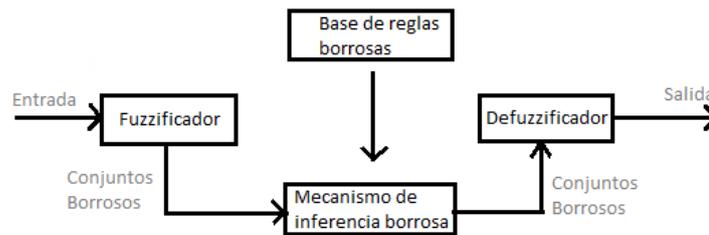
Figura 2.5 Sistema de control difuso realimentado



Sistema de control difuso realimentado, Fuente: (Valenzuela Hernández, Montoya Giraldo, & Giraldo Buitrago, 2013)

El CLD es principalmente compuesto de tres componentes: fuzzificación, inferencia o proceso de motor y defuzzificación (Baldovino & Dadios, 2015), en la Figura 2.6 se observan los elementos y el proceso que sigue el CLD.

Figura 2.6 Sistema de control lógica difusa



Sistema de lógica difusa, Fuente: (Santos & Miranda, 2012).

2.3.1 Fuzzificador

El Fuzzificador se encarga de tomar los valores numéricos de la entrada y transformarlos en valores difusos, adecuados para su manipulación por el mecanismo de inferencia. (Valenzuela Hernández, Montoya Giraldo, & Giraldo Buitrago, 2013).

2.3.2 Inferencia y Base de reglas

El mecanismo de inferencia es el cual procesa los datos proporcionados por el Fuzzificador, es este punto se aplican las operaciones lógicas como la implicación, la conjunción, la disyunción, la negación, etc. Debido a esto se toma los niveles de pertenencia y se asigna a la variable de salida (Escaño, García, & Nuevo, 2019). Para este proyecto se presenta una inferencia con la operación lógica AND.

La base de reglas difusas es la manera que tiene el sistema difuso para que se almacene el conocimiento lingüístico lo cual permitirá resolver los problemas que sean establecidos (Santos & Miranda, 2012).

2.3.3 Defuzzificador

El defuzzificador encuentra la salida concreta que mejor represente la información contenida en las reglas activadas por las variables de entrada (Escaño, García, & Nuevo, 2019), por lo tanto realiza un escalamiento para que los rango de salida sean

los adecuados en base a lo que se ha definido en el controlador (Valenzuela Hernández, Montoya Giraldo, & Giraldo Buitrago, 2013).

2.4 ALGORITMOS GENÉTICOS (AG)

Los AG son una técnica de modelamiento que proviene de la inteligencia artificial (De Solminihaç T, Echaveguren N, & Chamorro G, 2018), además, son una familia de algoritmos de búsqueda inspirados por los principios de la evolución en la naturaleza (Wirsansky, 2020), por lo que se basa en lo expresado por Charles Darwin de la teoría de la evolución (la supervivencia del más apto), inventados por John Holland en la década de 1960 (González Palomino, 2016).

Los AG son métodos de búsqueda probabilística que procesan soluciones a los problemas de optimización (Boggi, Razzitte, & Fano, 2014), por lo tanto este tipo de algoritmo de búsqueda es una forma para hallar la solución más conveniente (o la más apta) en el espacio del problema (o campo de búsqueda) (González Palomino, 2016).

Dada su analogía con la evolución natural permite que los algoritmos genéticos superen algunos de los obstáculos que encuentran los algoritmos tradicionales de búsqueda y optimización, especialmente para problemas con una gran cantidad de parámetros y representaciones matemáticas complejas. (Wirsansky, 2020)

Por lo tanto, la técnica de AG sobresale por ser un método con alto rendimiento, confiable, robusto y por su capacidad de búsqueda global, es apto para hallar relación entre varios datos, y puede hacer predicciones más rápidas y más posibles que algún otro método clásico. (Morales, Cappelletti, Casas, Hasperué, & Peltezer y Blancá, 2016)

2.4.1 Estructura de los algoritmos genéticos

Para que los AG realicen la optimización necesitan de una o varias funciones fitness (o función objetivo) (González Palomino, 2016). La función fitness (o función objetivo) es aquella que valora la capacidad de un individuo midiendo el resultado que representa de manera referente al óptimo que pueda obtenerse, usualmente, estas

funciones se establecen para acrecentar o despreciar algún criterio (Morales, Cappelletti, Casas, Hasperué, & Peltezer y Blancá, 2016).

En cada iteración del algoritmo, los individuos son evaluados usando una función de aptitud (también llamada función fitness). Esta es la función que buscamos optimizar o el problema que intentamos resolver. (Wirsansky, 2020)

Las etapas principales para llevar a cabo un AG son los siguientes:

2.4.1.1 Generar población inicial

Como primera etapa en un AG se debe generar una población original (ó inicial) de cromosomas de forma aleatoria. Dicha población se muestra mediante una matriz en la cual cada serie es un cromosoma que encierra las variables a optimizar (Boggi, Razzitte, & Fano, 2014).

2.4.1.2 Selección natural

Como siguiente etapa se encuentra la selección natural donde se decide qué cromosomas de la población inicial son más capaces para reproducirse y sobrevivir en la siguiente descendencia. La etapa de selección natural debe suceder en cada una de las iteraciones (Boggi, Razzitte, & Fano, 2014).

2.4.1.3 Selección de parejas

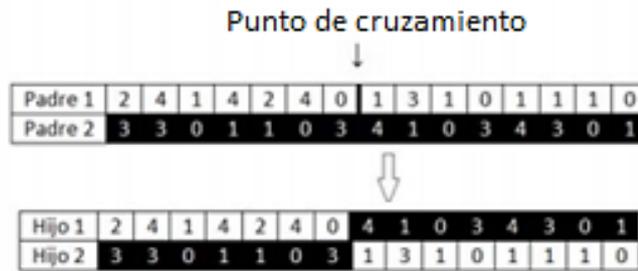
En la selección de parejas se selecciona dos cromosomas para cruzar, respetando que mientras más aptitud tenga mayor probabilidad de selección (Pinto, 2016).

2.4.1.4 Recombinación o Cruzamiento

En la etapa de cruzamiento los individuos seleccionados deben realizar un cruce. Por lo tanto, el resultado de intercambiar sus cromosomas en este proceso es dar otros dos individuos en la nueva generación. Los nuevos individuos obtenidos pueden ser mejores o iguales en aptitud que aquellos que los originaron (Morales, Cappelletti, Casas, Hasperué, & Peltezer y Blancá, 2016).

En la Figura 2.7 se puede observar la etapa de cruzamiento donde se elige un punto al azar en los cromosomas padres donde se recombinan las secciones para generar los dos nuevos hijos (Boggi, Razzitte, & Fano, 2014).

Figura 2.7 Etapa de Cruzamiento



Etapa de cruzamiento, Fuente: (Jiménez, Guardado, Cabrera, Rodriguez, & Figueroa, 2017)

2.4.1.5 Mutación

La mutación contribuye a la diversidad genética de la población (Jiménez, Guardado, Cabrera, Rodriguez, & Figueroa, 2017). Los AG pueden llegar a converger de forma precipitada en una sección de un mínimo local de la función fitness, por lo cual este es un problema que debe evitarse. Entonces por medio de mutaciones o cambios en alguna de las variables de los cromosomas se puede solucionar dicho problema. (Boggi, Razzitte, & Fano, 2014) cómo se ilustra en la Figura 2.8.

Figura 2.8 Etapa de Mutación



Etapa de mutación, Fuente: (Jiménez, Guardado, Cabrera, Rodriguez, & Figueroa, 2017)

2.4.1.6 Nueva generación

Otra etapa importante en los AG es la nueva generación de individuos (que ha sido seleccionada, cruzada y/o mutada) que pueden volver a iterar todo el proceso mencionado hasta que se llegue a un punto de vista donde se de un criterio de terminación (Boggi, Razzitte, & Fano, 2014). Todo este proceso puede tener un límite

de iteraciones o puede terminarse cuando el resultado (o solución) hallado sea el más apta para el problema planteado (Morales, Cappelletti, Casas, Hasperué, & Peltezer y Blancá, 2016).

Una vez conocidas las etapas para desarrollar un AG en la Figura 2.9 se detalla dicho proceso mediante un diagrama de flujo.

Figura 2.9 Diagrama de flujo del AG

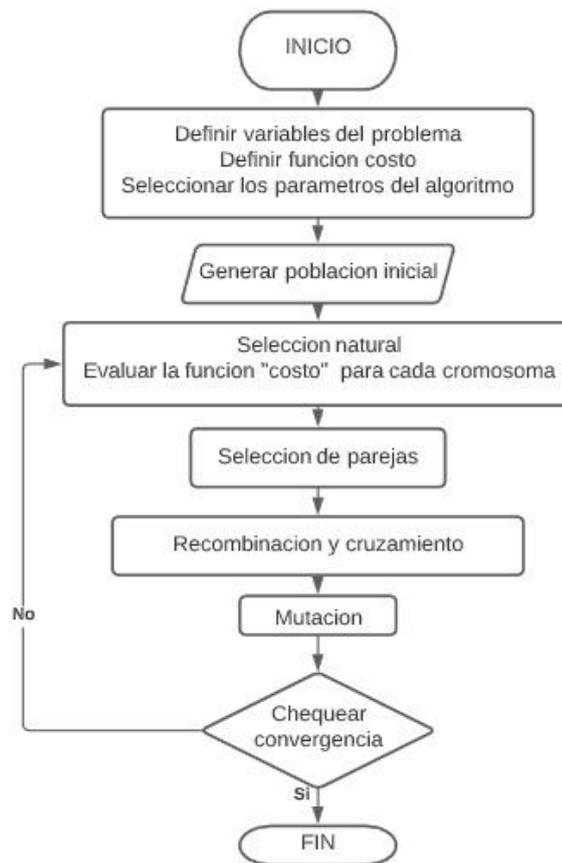


Diagrama de flujo del algoritmo genético, Fuente: Marlon Potosí y Grace Ruiz.

2.5 ÍNDICES DE DESEMPEÑO

Los criterios de desempeño que se proponen para ser seleccionados son:

- Máximo sobre impulso (M_p): Es el valor en el punto máximo de la curva de respuesta del sistema el cual es medido desde la señal de referencia.
- Tiempo de establecimiento (t_s): Es el tiempo que requiere la señal de respuesta para llegar y perdurar en un rango cerca del valor final.

- Tiempo de respuesta (t_r): Es el tiempo que demora la respuesta en llegar a un cierto porcentaje de su valor final, como puede ser del 10 al 90 por ciento.
- Error en estado estable (E_{ss}): Es el valor de la resta entre la señal de salida del comparador y el valor constante de la señal de entrada.
- Integral del error absoluto (IAE): Es una medida la cual determina la energía que el controlador emplea es decir el esfuerzo de control. (Andrade Sánchez & Tipán Jiménez, 2016).

2.5.1 Integral del error absoluto (IAE)

La Integral del Error Absoluto (por sus siglas en inglés Integral Absolute Error) es la integral de los valores absolutos del error de control durante un período de tiempo especificado como se muestra en la Ecuación 2.1 (Domański, 2019).

$$IAE = \int_{t_1}^{t_2} |\varepsilon(t)| dt \quad \text{Ec. (2.1)}$$

En términos de energía se puede decir que es la energía que emplea el control en un proceso.

2.6 TEST ESTADÍSTICO DE WILCOXON

La prueba de rango con signo utilizada para datos emparejados (Glen, 2016). Para emplear este test las muestras deben ser igual o mayor a 25 (Andrade Sánchez & Tipán Jiménez, 2016).

CAPÍTULO 3

DISEÑO E IMPLEMENTACIÓN

3.1 DESCRIPCIÓN DEL HARDWARE

En el presente trabajo se implementa un control genético difuso el mismo que va a ser aplicado en el módulo de nivel del laboratorio de Teoría de Control, en el cual los elementos principales del lazo de control son una bomba de agua, el sensor diferencial de nivel y la válvula.

3.1.1 Bomba Centrífuga

El primer actuador en ser caracterizado es la bomba centrífuga, la cual va a impulsar el agua con una rapidez dependiente del controlador implementado, permite transportar el agua del tanque de reserva hacia el tanque graduado e iniciar el proceso de control de la planta.

En la Figura 3.1 se observa la bomba centrífuga montada dentro del módulo de nivel, posteriormente en la Tabla 3.1 se determina todos los parámetros y características que esta bomba posee.

Figura 3.1 Bomba centrífuga



Fotografía Bomba centrífuga, Fuente: Marlon Potosí y Grace Ruiz.

Tabla 3.1 Especificaciones técnicas Bomba Centrífuga

CARACTERÍSTICA	UNIDAD
Caudal Nominal	40 [l/min]
Corriente Nominal	5 [A]
H máxima de succión	40 [m]
Voltaje Nominal	110 [V]
Potencia	0.5 [Hp]

Especificaciones técnicas Bomba Centrífuga. Fuente: (Espinoza, 2019)

3.1.2 Pantalla Táctil

La pantalla táctil empleada en este proyecto es de marca Nextion como muestra la Figura 3.2 por medio de la misma se realiza una interfaz hombre máquina (HMI) desde la cual se puede maniobrar, modificar, interactuar y observar sobre todo la variable del proceso de control, por lo tanto, se puede ingresar el valor de las variables de entrada y dar inicio al proceso. Este panel táctil ayuda a tener un ambiente didáctico para el control del módulo de nivel, así también en la Tabla 3.2 se especifica las características técnicas.

Figura 3.2 Pantalla Táctil Nextion



Fotografía Pantalla Táctil, Fuente: Marlon Potosí y Grace Ruiz

Tabla 3.2 Características Pantalla táctil Nextion.

CARACTERÍSTICAS	DESCRIPCIÓN
Modelo	NX8048
Tamaño	7 [in]
Touch panel	Resistivo
Resolución	800*480 [Colores]
Color	65536 [Píxeles]
Memoria RAM	3584 [Bite]
Frecuencia	48 [MHz]
Memoria FLASH	16 [MB]

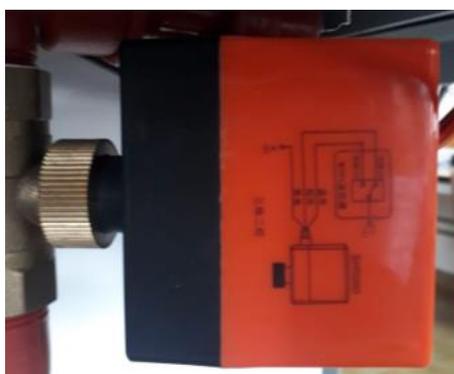
Características Pantalla táctil, Fuente: (Espinoza, 2019)

3.1.3 Servo Válvula

La válvula cuenta con un cierre y apertura con forma de bola, con un acople a un servo motor, es así como por medio de un rango de frecuencia esta realiza la acción de apertura o cierre del paso del fluido.

En la Figura 3.3 se observa la válvula instalada en el módulo y en la Tabla 3.3 se muestra las características de la misma.

Figura 3.3 Servo Válvula



Fotografía Servo Válvula, Fuente: Marlon Potosí y Grace Ruiz

Tabla 3.3 Características Servo Válvula

CARACTERÍSTICA	DESCRIPCIÓN
Modelo	TK-260
Voltaje Nominal	5 [V]
Tamaño	½" DN15
Material	Latón
Presión Nominal	1,6 [MPa]
Abrir/Cerrar	0,18[s]/180[°]

Características Servo Válvula, Fuente: (Espinoza, 2019)

3.1.4 Sensor de presión diferencial (SPD)

El SPD es un dispositivo que se encarga de generar una tensión a su salida al tener una variación de presión en su entrada, al ser un dispositivo piezoresistivo de silicio con un diafragma monolítico el mismo proporciona una salida altamente precisa y lineal, directamente proporcional a la presión ejercida.

En la Figura 3.4 se observa el sensor mientras que en la Figura 3.5 se muestra el sensor ya instalado y en la Tabla 3.4 se indica las respectivas características.

Figura 3.4 Sensor de presión diferencial.



Sensor de presión diferencial, Fuente: (Freescale, 2012)

Figura 3.5 Sensor de presión instalado



Fotografía sensor de presión diferencial instalado, Fuente: Marlon Potosí y Grace Ruiz

Tabla 3.4 Características del sensor de presión diferencial.

CARACTERÍSTICA	DESCRIPCIÓN
Modelo	MPX 2050
Voltaje Nominal	16 [V]
Presión	0 ~ 50 [KPa]
Temperatura	0 ~ 85 [°C]
Tiempo respuesta	1 [ms]
Corriente	6 [mA]

Características del sensor de presión diferencial, Fuente: (Freescale, 2012)

3.1.5 Tanque graduado

El módulo cuenta con un tanque de nivel graduado en centímetros como se muestra en la Figura 3.6. En este tanque es donde se realiza las pruebas de nivel y control del proceso de llenado y vaciado del mismo, además en la Tabla 3.5 se indican las respectivas características.

Figura 3.6 Tanque graduado



Fotografía tanque graduado, Fuente: Marlon Potosí y Grace Ruiz

Tabla 3.5 Características tanque graduado

CARACTERÍSTICA	DESCRIPCIÓN
Altura	500 [mm]
Ancho A	400 [mm]
Ancho B	400 [mm]
Material	Acrílico
Volumen	82 [l]
Tubería	½ [in]

Características tanque graduado, Fuente: (Espinoza, 2019)

3.1.6 Tanque de reserva

El tanque de reserva es un depósito en el cual se estanca o guarda el líquido que va a utilizarse o a desecharse durante el proceso de control, el mismo está conectado con el tanque principal por medio de tubería de PVC y cuenta con una tubería externa la que ayuda a conocer el nivel de agua de este tal como se presenta en la Figura 3.7 y en la Tabla 3.6 se dan a conocer sus características.

Figura 3.7 Tanque de reserva



Fotografía Tanque de reserva, Fuente: Marlon Potosí y Grace Ruiz

Tabla 3.6 Características Tanque de reserva

CARACTERÍSTICA	DESCRIPCIÓN
Altura	650 mm
Diámetro	400 mm
Material	Acero
Volumen	80 litros
Tubería	1/2 [in]

Características del Tanque de reserva, Fuente: (Espinoza, 2019)

3.1.7 Variador de frecuencia

El variador de frecuencia es un dispositivo electrónico, el cual es programable para regular la velocidad de operación de un motor eléctrico, en este caso la bomba centrífuga, se caracteriza por operar a la bomba en los dos sentidos de giro, variando su velocidad al depender de una entrada analógica de voltaje o corriente.

En la Figura 3.8 se observa el variador de frecuencia montado dentro del tablero de control del módulo y en la Tabla 3.7 se encuentra todas las especificaciones técnicas del mismo.

Figura 3.8 Variador de frecuencia.



Fotografía Variador de frecuencia. Fuente: Marlon Potosí y Grace Ruiz

Tabla 3.7 Características variador de frecuencia

ESPECIFICACION	DESCRIPCIÓN
Modelo	VFD007EL11A
Voltaje de IAV	0-10 [VAC]
Frecuencia	50/60 [Hz]
Amperaje	18,0 [A]
Voltaje de Salida	0-240 [VAC]
Rango de Frecuencia	0,1-600 [Hz]
Potencia	0,75 [kW]

Características variador de frecuencia, Fuente: (Espinoza, 2019)

3.2 DISEÑO DE LA TARJETA DE CONTROL

Para el enlace entre los dispositivos a controlar, actuadores, sensores y display se necesita una conexión física entre ellos, debido a esto se optó por realizar una placa electrónica la cual ayuda con la juntura de todos estos dispositivos, y consecuentemente con una programación del sistema de control que se realiza.

A continuación, se presentan los elementos principales de la tarjeta.

3.2.1 Controlador

El microcontrolador AT -MEGA 2560 es un controlador digital que cuenta con un microchip de 8 bits de alto rendimiento idóneo para la ejecución de este proyecto, una memoria aproximada de 25000 bytes. Es apta para realizar el control genético e instalación de varias librerías que permite un control simultaneo de varios dispositivos.

En la Figura 3.9 se observa la estructura física del microcontrolador, dejando en la Tabla 3.8 la disponibilidad de sus características.

Figura 3.9 Controlador



Controlador, Fuente: (Espinoza, 2019)

Tabla 3.8 Características Controlador

CARACTERÍSTICA	DESCRIPCIÓN
Microcontrolador	8 [bit]
Memoria flash	253 [Kb]
Memoria EEPROM	1 [Kb]
Memoria RAM	8 [Kb]
Frecuencia	20 [MHz]
Entradas Analógicas	13 [pins]
Entrada/Salida	54 [pins]
Material	Aluminio

Características controlador, Fuente: (Espinoza, 2019)

3.2.2 Conversor de Voltaje DC-DC

El conversor de voltaje es un dispositivo Step – Down, el cual tiene como objetivo entregar un voltaje de salida constante inferior al voltaje de entrada, en el caso se ocupa para la alimentación de toda la placa de control y dispositivos aledaños a esta.

En la Tabla 3.9 se muestra en las características del conversor DC-DC.

Tabla 3.9 Conversor DC-DC

CARACTERÍSTICA	DESCRIPCIÓN
Modelo	LM 2596
Voltaje Entrada	4.5 ~ 40 [V]
Voltaje Salida	1.23 ~ 27 [Kb]
Frecuencia de Trabajo	150 [KHz]
Potencia de Trabajo	25 [W]
Eficiencia	92 [%]
Corriente de Salida	3 [A]

Conversor DC-DC, Fuente: Marlon Potosí y Grace Ruiz.

3.2.3 Conexiones y acondicionamiento de señales en el microcontrolador

En el diseño de la placa del controlador, se tomó varios puntos en cuenta, las salidas y las entradas al microcontrolador, la conexión serial que necesita la pantalla los cuales se indican en la Tabla 3.10

Tabla 3.10 Conexiones Microcontrolador

Dispositivo conectar	IN/ OUT Controlador	Pines	Tipo de pin
Sensor	IN	1	Analógico
Variador 1	OUT	1	Analógico / PWM
Variador 2	OUT	1	Digital
Pantalla	Comunicación	2	TX /RX
Válvula	OUT	1	PWM

Conexiones detalladas necesarias para el Microcontrolador, Fuente: Marlon Potosí y Grace Ruiz.

Para obtener la mayor eficiencia del variador de frecuencia, en su entrada analógica (IAV), se necesita tener un voltaje de 10 [V], debido a que el microcontrolador no cuenta con una salida de este voltaje, se empleó un amplificador operacional con una ganancia de 2 en la configuración no inversora. Las Ecuaciones 3.1 y 3.2 muestran el cálculo de los valores implementados en el amplificador.

$$V_s = E \left(1 + \frac{R_f}{R_g} \right) \quad \text{Ec. (3.1)}$$

$$V_s = 5[V] \left(1 + \frac{1}{1} \right) = 10 [V] \quad \text{Ec. (3.2)}$$

Donde R_f es la resistencia de realimentación a la entrada inversora y R_g es una resistencia a tierra de la misma entrada.

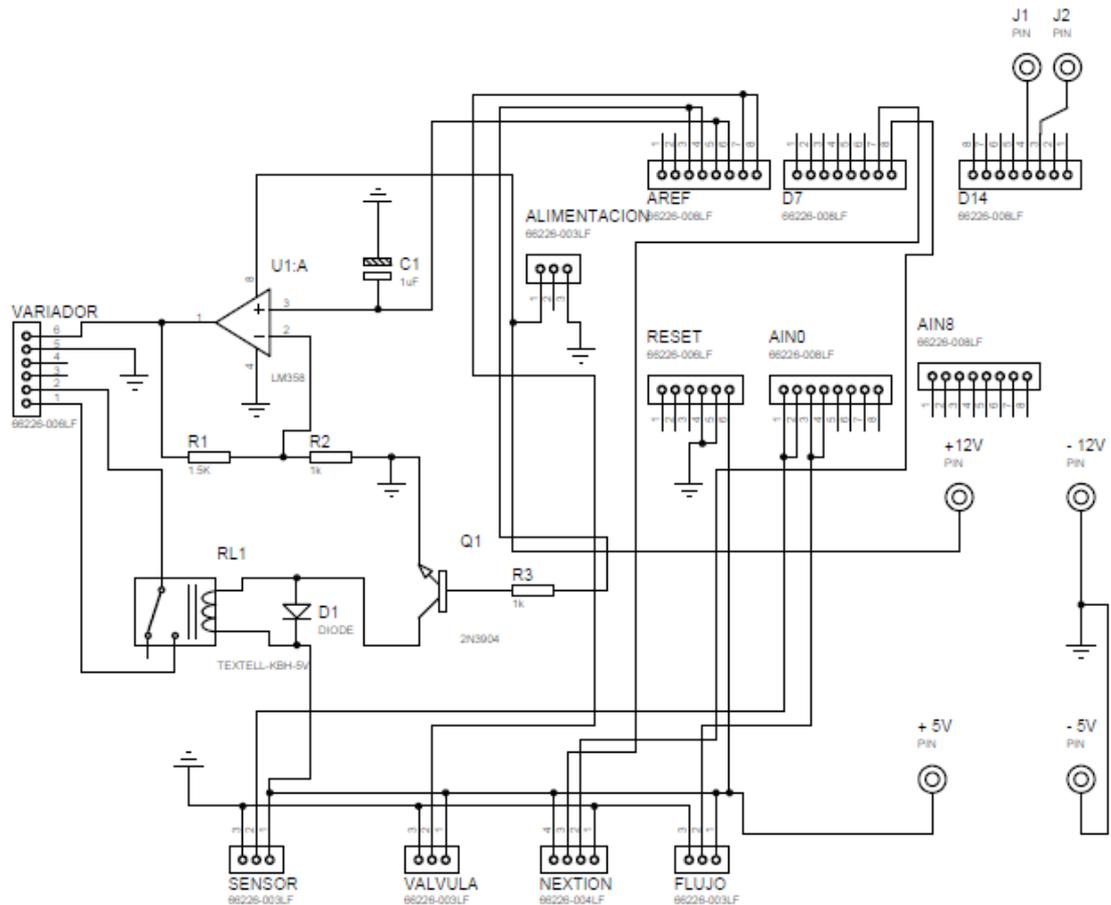
La entrada número dos del variador es una entrada tipo switch que debe tener una conexión con los pines (MI1) y (DCM) para activar el sentido de giro 1, para esto se emplea un relé el cual va a unir estos dos pines por medio de su contacto normalmente abierto, que al momento de activarse por medio del controlador realizara la acción pertinente.

El sensor de presión, necesita de una alimentación y una conexión a un pin analógico del microcontrolador para su lectura.

La válvula necesita ser conectada a un pin tipo PWM para el accionamiento de esta.

En la Figura 3.10 se presenta todo el diseño del circuito a colocar en la placa, tomando en cuenta todas las conexiones del microcontrolador y pines de conexión de dispositivos a controlar.

Figura 3.10 Diseño circuito placa de control



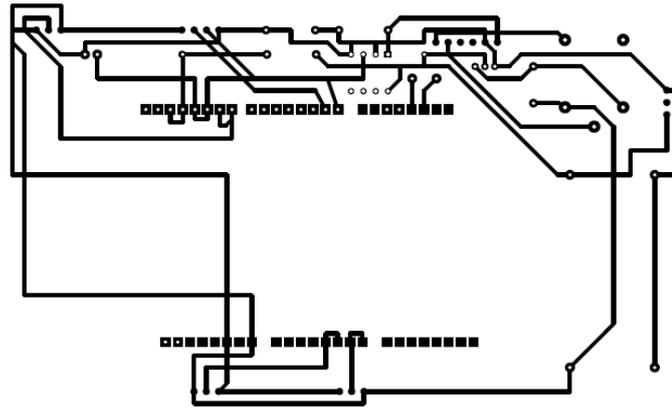
Diseño del circuito para la placa de control, Fuente: Marlon Potosí y Grace Ruiz

3.2.4 Desarrollo del layout de la tarjeta

Para el layout de la tarjeta, se ocupó una placa de baquelita de doble lado, debido a que el microcontrolador posee un gran número de pines a conectar, en la Figura 3.11 se muestra el diseño a trazar en la baquelita para su conexión de dispositivos pasivos y pines de conexión a los dispositivos exteriores a la placa.

En la conexión de la placa hacia los dispositivos externos se emplea conectores tipo pin, en donde se facilita una conexión y desconexión rápida.

Figura 3.11 Diseño placa del controlador

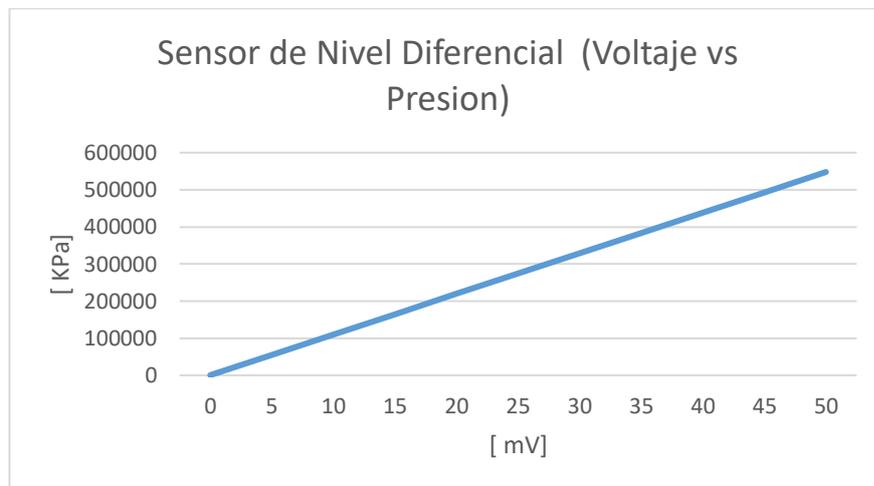


Diseño de la placa del controlador, Fuente: Marlon Potosí y Grace Ruiz

3.3 DISEÑO DEL ALGORITMO DIFUSO Y OPTIMIZACIÓN

Uno de los elementos repotenciados en el sistema es el sensor de nivel, el cual es de principio diferencial y comportamiento lineal como se muestra en la Figura 3.12

Figura 3.12 Gráfica de prueba de relación lineal



Gráfica de prueba de relación lineal, Fuente: Marlon Potosí y Grace Ruiz

La conversión de presión a nivel se la realiza mediante la Ecuación 3.3:

$$\text{Presion} = \left(\frac{V_{\text{sensor}} - 0.04 * V_{\text{fuente}}}{0.09 * V_{\text{fuente}}} \right) + \text{tol} \quad \text{Ec. (3.3)}$$

En donde:

V sensor: es el voltaje que proporciona el sensor.

V fuente: es el voltaje al cual está trabajando el sensor.

Con la Ecuación 3.3 se obtiene la presión del agua que mide el sensor de forma real y para obtener el nivel se realiza una adecuación a la lectura del sensor por medio de la Ecuación 3.4.

$$\text{Nivel} = \left(\frac{\text{Presion} * 10000}{\text{Den} * g} \right) * 100 \quad \text{Ec. (3.4)}$$

En donde:

Presión: es la presión calculada en la ecuación 3.3.

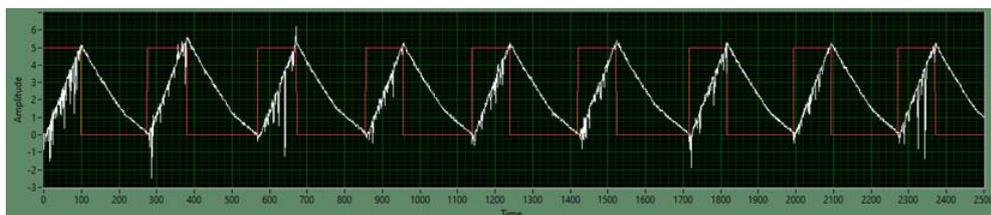
Den: es la densidad del agua.

g: gravedad a la que se está realizando la medición

3.3.1 Identificación del sistema

Continuando con el proceso de control, es indispensable sacar la función de transferencia (FT) de la planta. Matlab es el software el cual se va a ocupar como herramienta para este proceso, se ingresan los datos que fueron tomados cada 500 ms, aplicando una señal pseudoaleatoria de tipo escalón a la planta, la cual va a ser encendida al llegar a un nivel de 10 cm y va a ser apagada en un nivel de 20 cm, como se presenta en la Figura 3.13.

Figura 3.13 Gráfica de la señal pseudoaleatoria de tipo escalón y respuesta del sistema



Gráfica de la señal pseudoaleatoria de tipo escalón y respuesta del sistema, Fuente: Marlon Potosí y Grace Ruiz

Se toma el tiempo que tarda en llegar de un punto de llenado a otro, por medio de Excel se realiza una tabla de entrada y salida en la planta y como siguiente paso se procede a ocupar el Toolbox de identificación de Matlab para obtener la FT del sistema, como detalla en la Ecuación 3.5.

$$FT = \frac{1.5459}{49.499 S+1} \quad \text{Ec. (3.5)}$$

La función obtenida de la planta da un 70% de validación gráfica con respecto a la señal real de la respuesta del sistema.

3.3.2 Controlador Difuso

El control difuso por optimizar cuenta con dos entradas y una salida; la primera entrada indica el nivel al que se desea que el tanque llegue, tomando como mínimo 0 cm y como máximo 40 cm, que es la graduación que posee el tanque principal, para ello se ocupa conjuntos triangulares para cada una de las 5 variables lingüísticas a utilizar en la entrada.

Como segunda entrada al sistema de control se optó por tomar la variable de la válvula, la cual representa el porcentaje de apertura que posee esta, mínimo 0 y máximo 100% en donde se indica la apertura completa de la válvula, dado por conjuntos triangulares que representan a las 5 variables lingüísticas de esta entrada como se muestra en la Tabla 3.11.

Tabla 3.11 Variables Lingüísticas.

VARIABLES LINGÜÍSTICAS		
Nivel (Entrada)	Válvula (Entrada) Apertura en porcentaje	Motor (Salida)
P – Pequeño	0	L – Lento
MP – Medianamente Pequeño	25	M – Medio
N – Normal	50	ML – Medianamente Lento
MG – Medianamente Grande	75	MR – Medianamente Rápido
G – Grande	100	R – Rápido

Descripción de las Variables Lingüísticas, Fuente: Marlon Potosí y Grace Ruiz.

En la salida, así como en la entrada se emplea conjuntos triangulares, para cada una de las variables lingüísticas, al tener un porcentaje del rendimiento de la bomba centrífuga de 0 como mínimo y 100 como máximo.

La tabla FAM en donde se representan las reglas difusas se muestran en un máximo de 5 veces de repetición de cada conjunto de salida, presentado en la Tabla 3.12 en donde se toma un conocimiento previo de la planta para realizar este arreglo.

Tabla 3.12 Reglas Difusas

Válvula/Nivel	P	MP	N	MG	G
0	L	L	M	ML	MR
25	L	L	M	ML	MR
50	L	M	ML	MR	R
75	M	M	MR	R	R
100	ML	ML	MR	R	R

Descripción de las Reglas difusas por medio de la matriz FAM, Fuente: Marlon Potosí y Grace Ruiz.

En la Figura 3.14 se detalla en un diagrama de flujo la programación implementada en el control difuso.

Figura 3.14 Diagrama de flujo del control difuso.

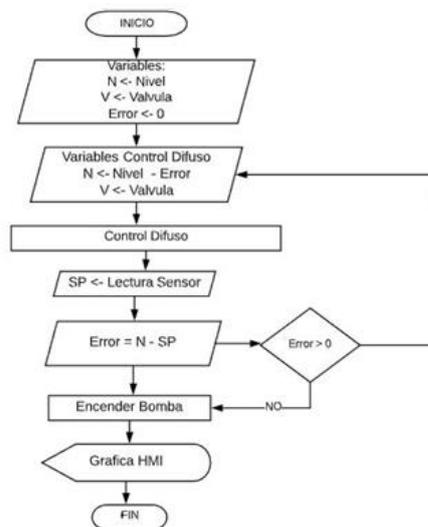
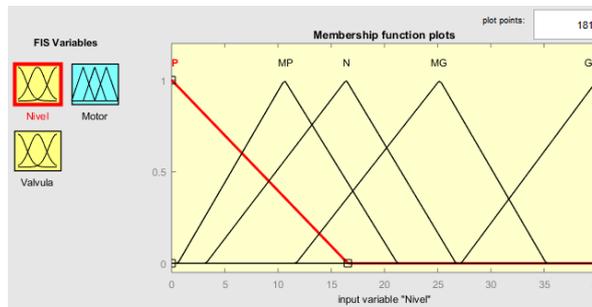


Diagrama de flujo del control difuso, Fuente: Marlon Potosí y Grace Ruiz

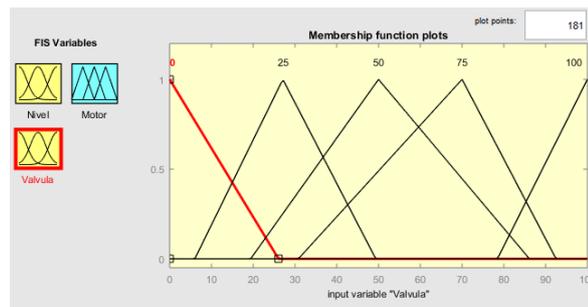
Teniendo diseñado el control difuso se procede a generar el archivo fis con la ayuda del software de Matlab, en donde se puede visualizar la forma de las variables lingüísticas y conjunto universo de cada una de las entradas y de la salida tal como se muestran en las Figuras 3.15, 3.16 y 3.17.

Figura 3.15 Variable de entrada Nivel.



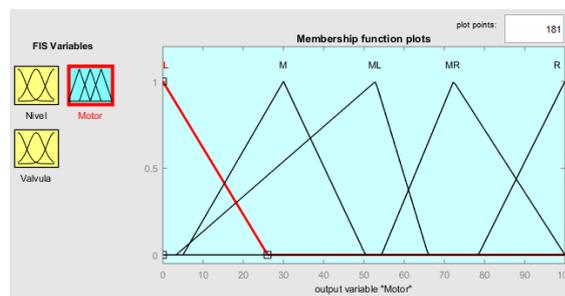
Variable de entrada Nivel, Fuente: Marlon Potosí y Grace Ruiz

Figura 3.16 Variable de entrada Válvula.



Variable de entrada Válvula, Fuente: Marlon Potosí y Grace Ruiz

Figura 3.17 Variable de salida Motor.



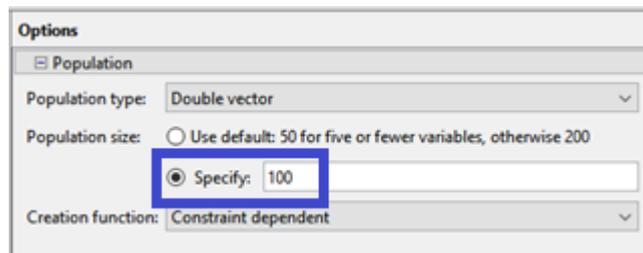
Variable de salida Motor, Fuente: Marlon Potosí y Grace Ruiz

3.3.3 Software de optimización genético difuso

La optimización del control difuso se lo realiza mediante el Toolbox de optimización en el software Matlab, en el cual se ingresa el control realizado previamente con la extensión fis, para posteriormente optimizarlo por medio de un algoritmo genético.

El primer paso de la optimización mediante el algoritmo genético, es seleccionar la población que va a ser optimizada. Es habitual en un control difuso tener una población para las entradas, como en este caso es el nivel y el porcentaje de apertura y cierre de la válvula. En la Figura 3.18 se presenta la selección de la población universo que va a ser optimizada.

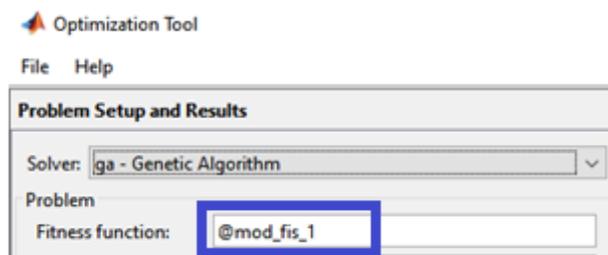
Figura 3.18 Selección de la Población AG - Matlab



Selección de la Población AG – Matlab, Fuente: Marlon Potosí y Grace Ruiz

Al haber determinado la población a ser optimizada, se establece la función fitness, con la cual se va a evaluar los mejores conjuntos de las funciones de pertenencia. La Figura 3.19 presenta la declaración de la función.

Figura 3.19 Función Fitness



Función Fitness, Fuente: Marlon Potosí y Grace Ruiz – Matlab

La función fitness dentro de la optimización va a utilizar varias funciones entrelazadas, la primera de ellas realiza una modificación de los parámetros para evaluar con este un nuevo conjunto de salida, el cual con una función secundaria entrelazada va a comparar el error cuadrático medio de este con otro conjunto de salida.

$$\text{fuz.input(i).mf(j).params(1)} = \text{fuzfsys.input(i).mf(j).params(1)} + \text{delta(delta_num)} \quad \text{Ec. (3.6)}$$

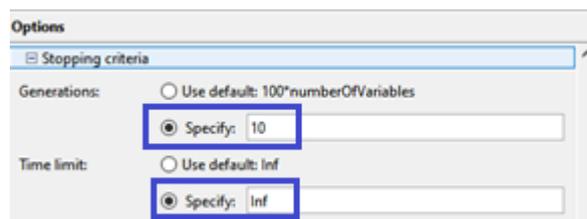
En la Ecuación 3.6, **input** selecciona que entrada va a trabajar, **mf** elige la función de membresía y **params** selecciona los parámetros, tomando en cuenta que al trabajar con funciones de membresía tipo triangulares estos cuentan con 3 parámetros en cada función.

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y})^2}{n}} \quad \text{Ec. (3.7)}$$

La Ecuación 3.7 detalla el error cuadrático medio donde compara el nuevo arreglo de salidas modificadas, con un arreglo previamente subido de las salidas del sistema de control difuso original sin modificar.

Se parametrizan los demás componentes del algoritmo genético, tales como las iteraciones (que es el número de veces que va a tomar de manera aleatoria un dato, lo va a combinar y va a comparar para reconocer cual es la mejor opción), así como el tiempo que se va a demorar en contabilizar estas 10 iteraciones, tal y como se indica en la Figura 3.20.

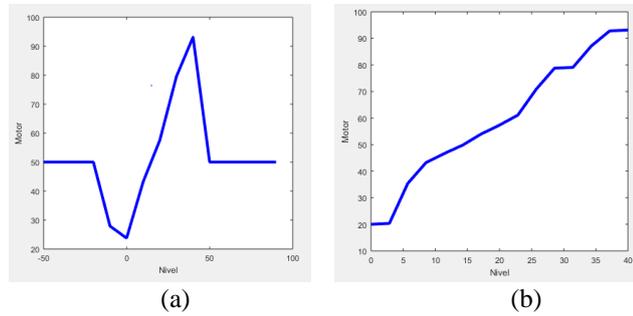
Figura 3.20 Parámetros de optimización - Matlab



Parámetros de optimización en el software Matlab, Fuente: Marlon Potosí y Grace Ruiz

Al finalizar todo el proceso se tiene un arreglo de varios datos de las 10 iteraciones dejando al final la mejor opción, y es esta la que va a ser usada para generar un nuevo sistema de control difuso, el cual pasa a tomar el nombre de Genético Difuso.

Figura 3.21 Optimización Población Nivel – Matlab



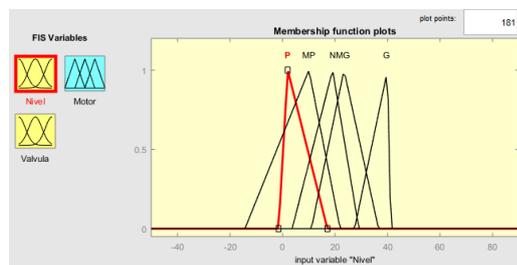
Optimización Población Nivel – Matlab, Fuente: Marlon Potosí y Grace Ruiz

Como un preámbulo del análisis se tiene que la Figura 3.21(a) tiene una mayor linealidad en comparación con la Figura 3.21(b), dando una idea de cómo el control difuso ha sufrido un cambio en su población inicial de la entrada nivel.

3.3.4 Controlador Genético Difuso

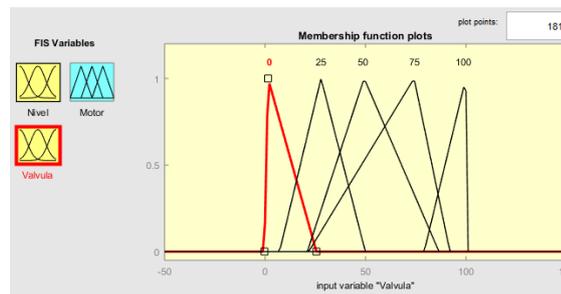
El controlador difuso sufrió un cambio en su población de universo de entrada, tanto de nivel como de válvula, obteniendo una variación en su linealidad, lo cual es reflejado en la Figura 3.22 y 3.23, como en la posición de las variables lingüísticas y los límites de cada conjunto.

Figura 3.22 Control Genético Difuso Entrada Nivel – Matlab



Control Genético Difuso Entrada Nivel – Matlab, Fuente: Marlon Potosí y Grace Ruiz

Figura 3.23 Control Genético Difuso Entrada Válvula – Matlab



Control Genético Difuso Entrada Válvula – Matlab, Fuente: Marlon Potosí y Grace Ruiz

Tanto en el nivel como en la válvula, los nuevos límites de inicio de los conjuntos cuentan con un intervalo de incremento, el cual ha modificado el algoritmo genético para corregir los errores que tiene el control difuso al inicio y al final de la población de universo de cada entrada, dando así valores continuos y evitando un cierto rango que da un valor constante. En la Figura 3.24 se detalla en un diagrama de flujo el controlador genético difuso.

Figura 3.24 Diagrama de flujo del Control Genético Difuso

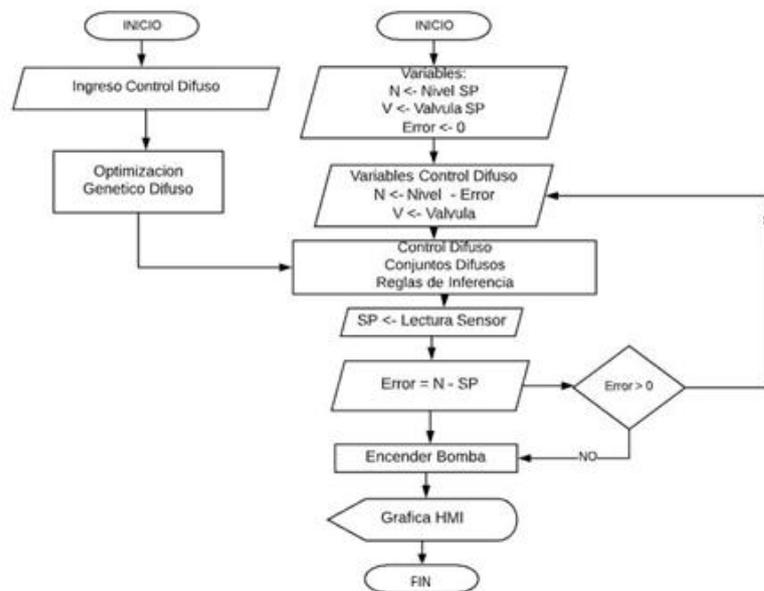


Diagrama de flujo del Control Genético Difuso, Fuente: Marlon Potosí y Grace Ruiz

3.4 DESCRIPCIÓN DEL SOFTWARE DE AUTOMATIZACIÓN

Para implementar el nuevo control genético difuso, se optó por utilizar el microcontrolador Arduino Mega, el cual cuenta con una conexión serial que permite la comunicación con la pantalla táctil. La pantalla contiene una interfaz gráfica para el ingreso de los datos de control en la planta.

3.4.1 IDE Controlador Arduino Mega

Para la programación del sistema de control se ha escogido el software IDE de Arduino, el cual es una plataforma de licencia abierta desarrollado bajo el lenguaje C++, que facilita y proporciona un amplio número de herramientas para desarrollar y diseñar proyectos de esta índole. En la Figura 3.25 se observa el ambiente del IDE del controlador con una pequeña fracción del código realizado y en la parte de anexos se puede apreciar el código completo del controlador a implementarse.

Figura 3.25 IDE controlador



```
prueba1.7.20
#include <Nextion.h>
#include <Servo.h>
#include <fuzzy.h>
Servo servoMotor;
fuzzy controlfuzzy; // Creamos un objeto de tipo fuzzy.
int variable=0,ser;
int currentPage = 0;
int nivset =0, valset=0 ,zout=0 ,lev=0 ,extra=0;
float n=0,v=0,z=0;
const int led = 13;
double level, Vout, Pre, Vs=5.0;
double aux;
double tol=0.015; //tolerancia-ajusta la medida de presión
int i, den=997; //densidad del agua
double gra=9.8; //gravedad

float MOTOR[] = {0,100}; // Definimos el universo de discurso (rango)de la variable salida
float NIVEL[] = {0,45}; // Definimos universo de discurso de la variable de entrada
float VALVULA[] = {0,100}; // Definimos universo de discurso de la segunda variable de entrada
////////////////////////////////////
float P[] = [-1.5,1.998,17.1,'T']; //Definimos los conjuntos difusos, colocando 'T' si es triangular, o 'R' si es trapezoidal
float MP[] = [-14.3,10.05,21.96,'T']; // Conjuntos difusos de entrada
float N[] = {3.677,19.08,29.09,'T'};
```

IDE controlador, Fuente: Marlon Potosí y Grace Ruiz

3.4.2 Edición de la pantalla

La pantalla cuenta con una interfaz la cual es programada en un ambiente propio de su marca denominada Nextion Editor, en la que se puede realizar cualquier tipo de modificación, construcción, edición y montaje de todo el ambiente que el usuario va a poder observar al manipular el controlador. La Figura 3.26 muestra el editor utilizado y la carátula que se implementó en el HMI.

Figura 3.26 Nextion Editor



Nextion Editor, Fuente: Marlon Potosí y Grace Ruiz

Este software de desarrollo es propio de GUI el cual es libre y solo es aplicable a productos de la misma compañía.

3.4.3 Interfaz Hombre—Máquina (HMI)

La comunicación entre el hombre y la máquina es el primer paso en una automatización y control debido a que si esta se la realiza de manera amigable el operario podrá dar un uso eficiente a la planta, por lo que al ser implementado dentro del módulo el nuevo controlador genético difuso, tiene la presentación de la Figura 3.27.

Figura 3.27 Pantalla 1 HMI

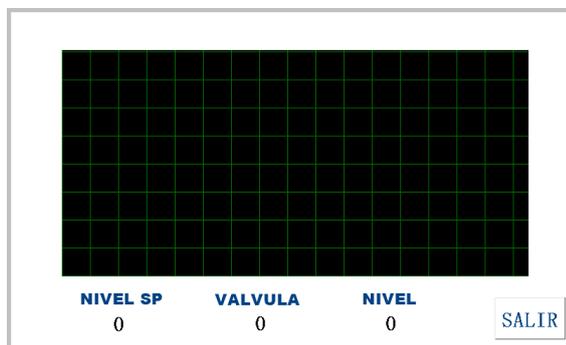


Pantalla 1 HMI, Fuente: Marlon Potosí y Grace Ruiz

En la Figura 3.27 se presenta la interfaz de la primera pantalla en donde se realiza la elección del nivel del tanque y el porcentaje de apertura de la válvula, así como un botón que va a dar inicio al proceso y un mensaje en donde reconoce cuando el controlador tiene comunicación con la pantalla.

En la Figura 3.28 se presenta una segunda pantalla de la interfaz en donde se observa tres indicadores del nivel seteado, apertura de la válvula y el nivel real que se encuentra el tanque, así como una waveform en donde se grafica la forma de onda que va a tener el sistema.

Figura 3.28 Pantalla 2 HMI



Pantalla 2 HMI, Fuente: Marlon Potosí y Grace Ruiz

CAPÍTULO 4

PRUEBAS Y RESULTADOS

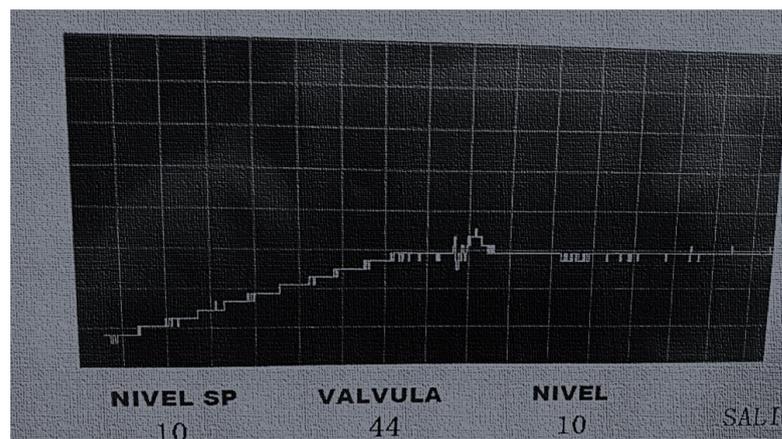
En el presente capítulo se analiza las pruebas realizadas al controlador Fuzzy convencional (CF) y al controlador generado mediante la sintonización por AG es decir el Controlador Genético Difuso (CGD).

Dicho análisis se realiza evaluando el comportamiento de los controladores en base al índice de desempeño IAE y el test de Wilcoxon.

4.1 PRUEBA DEL CONTROLADOR FUZZY CONVENCIONAL

Para esta prueba se ingresa un Set point de Nivel de 10 cm y para la Válvula de 44% de apertura, de esta manera en la Figura 4.1 se muestra cómo es el comportamiento de la planta con el CF.

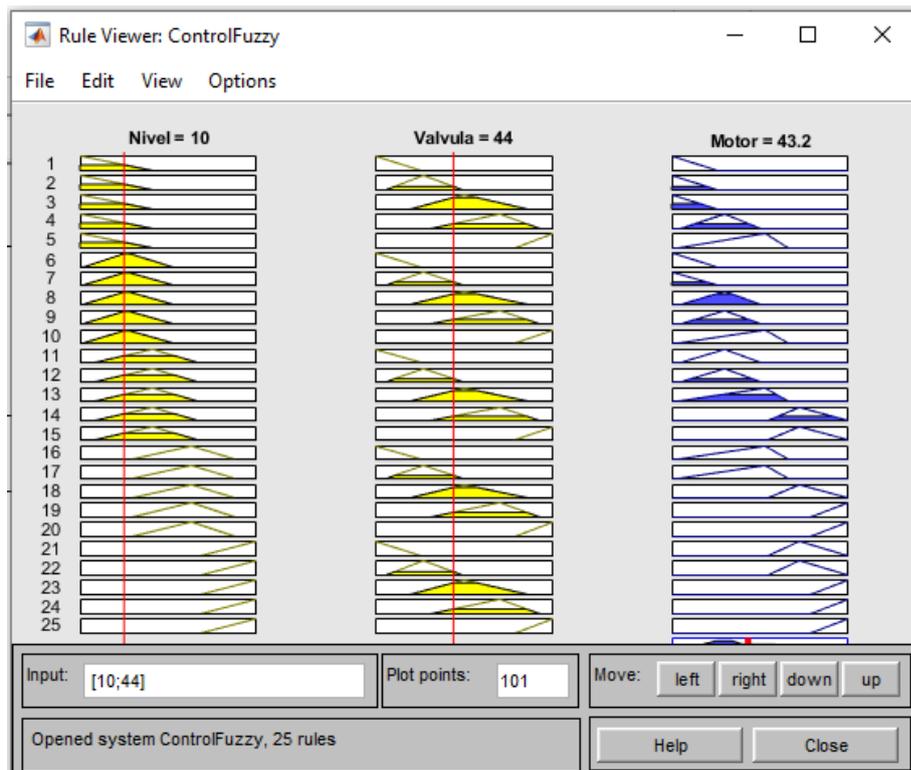
Figura 4.1 Comportamiento de la planta con el CF



Comportamiento de la planta con el CF, Fuente: Marlon Potosí y Grace Ruiz.

En el Toolbox de Matlab de Fuzzy se da los valores de cada una de las variables para así observar el resultado que se espera de la variable salida, en este caso del motor, como se indica en la Figura 4.2.

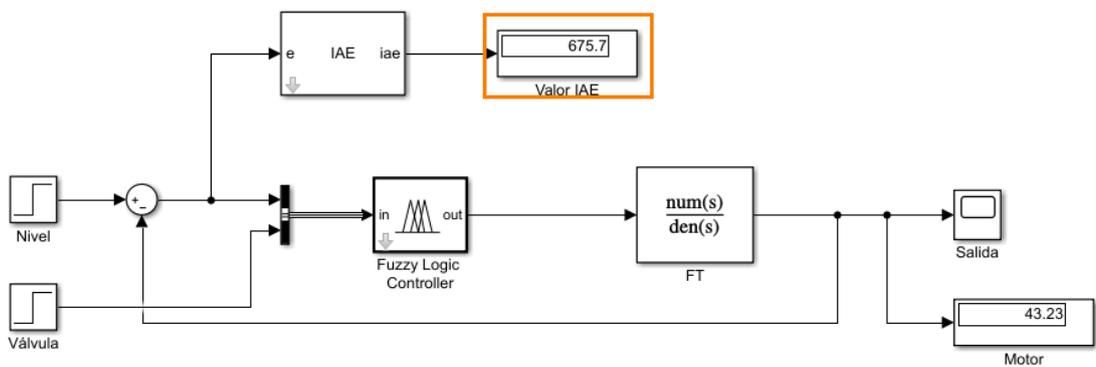
Figura 4.2 Tool box de Matlab de Fuzzy ingreso de datos en las variables



Tool box de Matlab de Fuzzy ingreso de datos en las variables, Fuente: Marlon Potosí y Grace Ruiz.

Una vez conocido estos valores se procede a realizar el proceso en Simulink para obtener el IAE del CF como se presenta en la Figura 4.3.

Figura 4.3 CF en Simulink para obtención del IAE

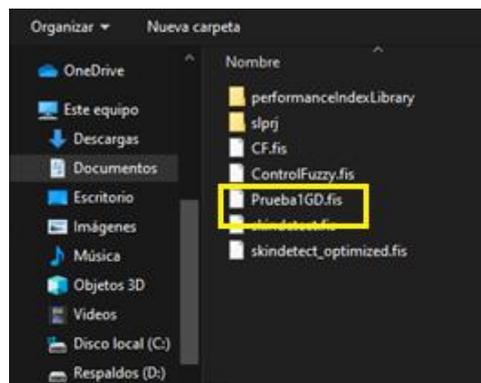


Prueba del CF en Simulink para obtención del IAE, Fuente: Marlon Potosí y Grace Ruiz

4.2 PRUEBAS DEL CONTROLADOR GENÉTICO DIFUSO (CGD)

En este caso mediante el código en Matlab se realiza la respectiva optimización por AG y de esta forma se genera un nuevo grupo de condiciones de los conjuntos antes previstos. Por medio del código en Matlab se genera un nuevo .fis como se muestra en la Figura 4.4 y de esta manera se observará las nuevas condiciones.

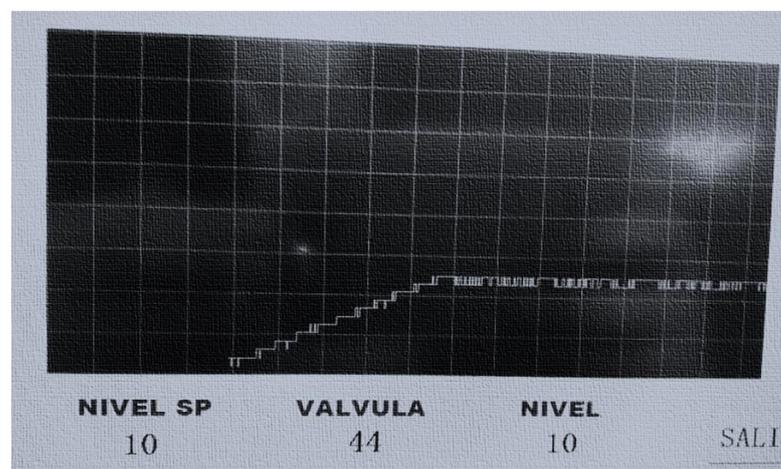
Figura 4.4 Grupo de condiciones generado mediante la optimización por AG



Grupo de condiciones generado mediante la optimización por AG, Fuente: Marlon Potosí y Grace Ruiz.

De esta manera se puede probar ya el nuevo controlador. En la Figura 4.5 se presenta el comportamiento de la planta.

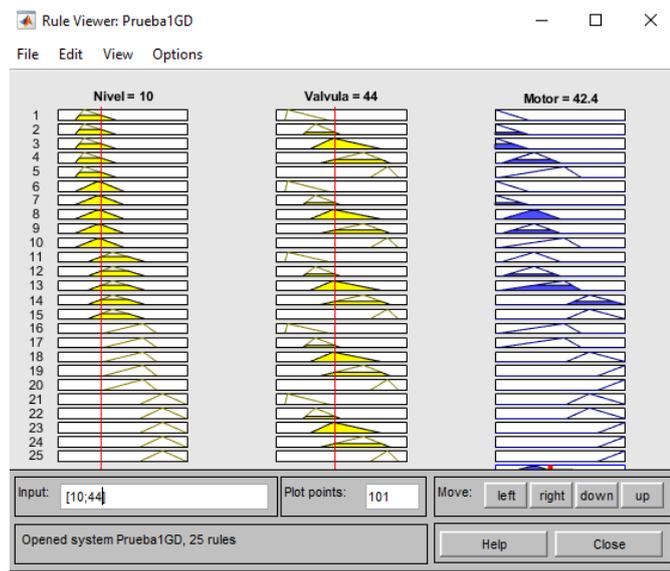
Figura 4.5 Comportamiento de la planta con el CGD



Comportamiento de la planta con el CGD, Fuente: Marlon Potosí y Grace Ruiz.

Para realizar la verificación de las siguientes pruebas obtenidas por la sintonización del AG se utiliza el Toolbox de Fuzzy en Matlab como se muestra en la Figura 4.6, una vez generado el archivo se abre y se debe volver a colocar los valores tal como se realizó con el CF.

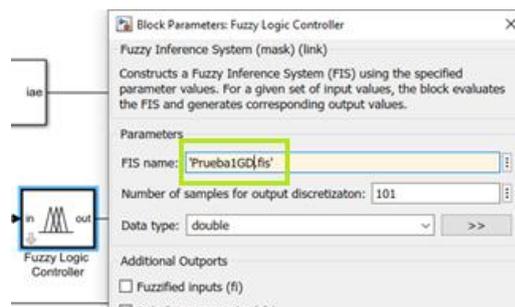
Figura 4.6 Tool box de Fuzzy Matlab Prueba 1 CGD



Tool box de Fuzzy Matlab Prueba 1 CGD, Fuente: Marlon Potosí y Grace Ruiz.

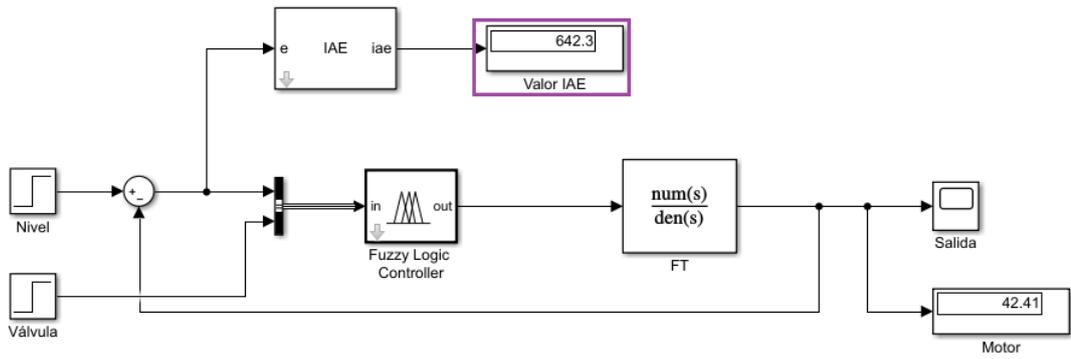
Así mismo se colocará el archivo nuevo generado de la Prueba 1 en Simulink (como se observa en la Figura 4.7) para la obtención del IAE (como se presenta en la Figura 4.8).

Figura 4.7 Carga de archivo Prueba 1 en Simulink



Carga de archivo Prueba 1 en simulink, Fuente Marlon Potosí y Grace Ruiz.

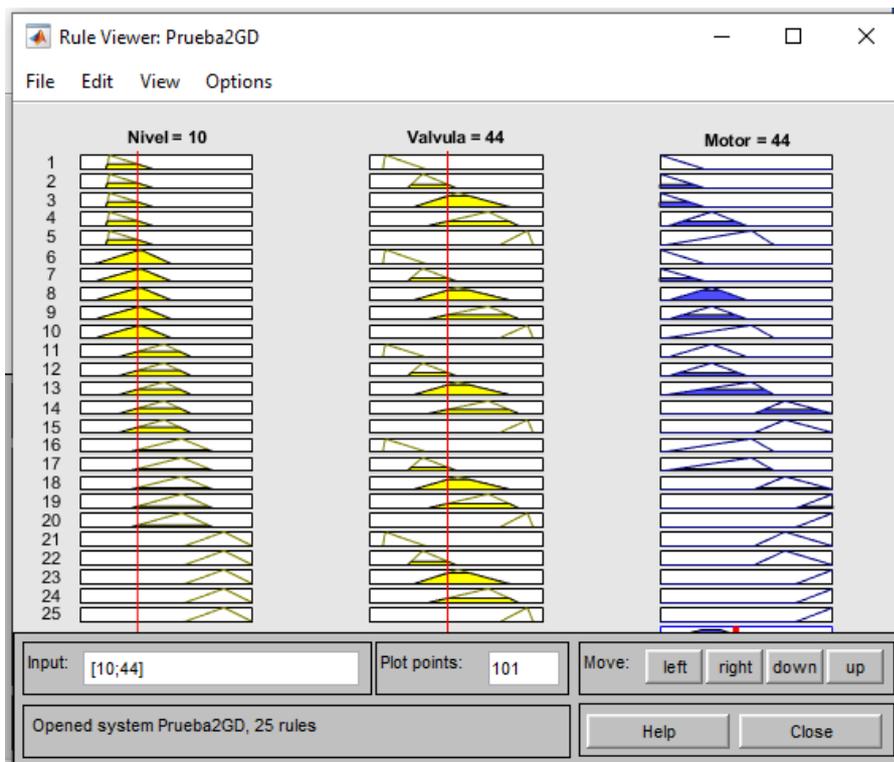
Figura 4.8 Prueba 1 en Simulink para obtención del IAE



Prueba 1 en Simulink para obtención del IAE, Fuente: Marlon Potosí y Grace Ruiz.

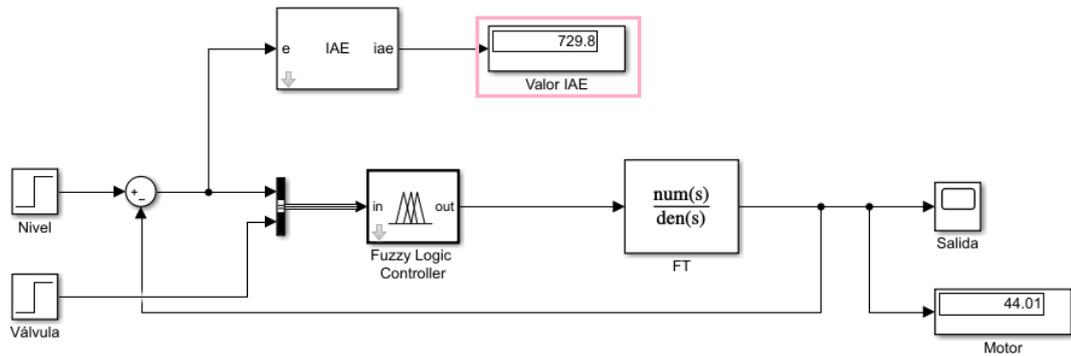
Se muestra a continuación una segunda prueba realizada mediante Matlab como en la Figura 4.9 y en Simulink con en la Figura 4.10.

Figura 4.9 Tool box de Fuzzy Matlab Prueba 2 CGD



Tool box de Fuzzy Matlab Prueba 2 CGD, Fuente: Marlon Potosí y Grace Ruiz.

Figura 4.10 Prueba 2 en Simulink para obtención del IAE

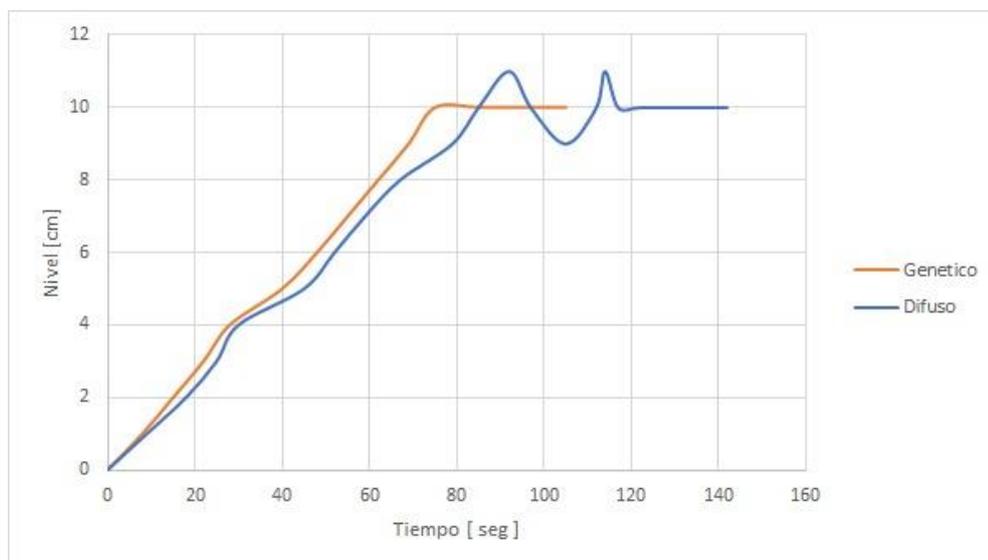


Prueba 2 en Simulink para obtención del IAE, Fuente: Marlon Potosí y Grace Ruiz.

4.3 ANÁLISIS COMPARATIVO DE CONTROLADORES

Después de cada una de las pruebas realizadas se puede observar en las siguientes imágenes el comportamiento del controlador implementado, para lo cual en la figura 4.12 se puede observar que el controlador Genético Difuso tiene una mejor respuesta que el controlador Fuzzy clásico.

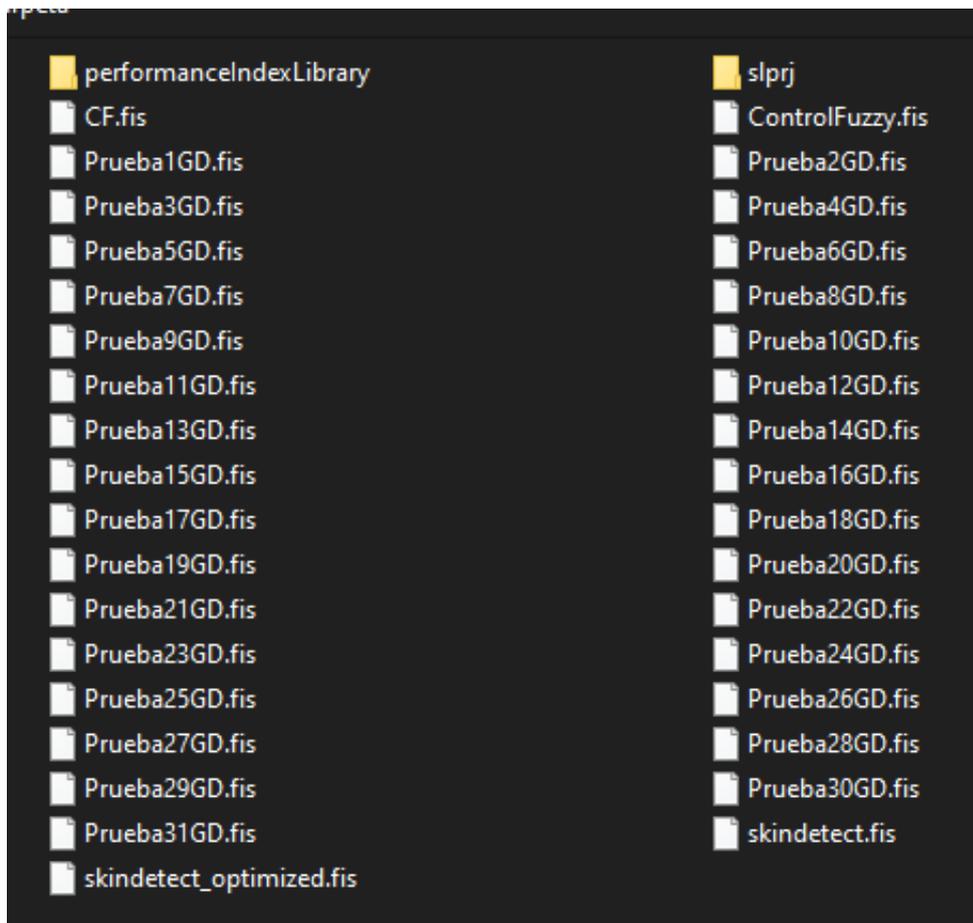
Figura 4.11 Comportamiento de los controladores estudiados.



Comportamiento de los controladores Genético Difuso y Fuzzy clásico, Fuente: Marlon Potosí y Grace Ruiz.

Se realizaron 30 pruebas mediante AG para la obtención del nuevo Controlador Genético Difuso (como se muestra en la Figura 4.12). En la Tabla 4.1 se puede apreciar cada una de las pruebas realizada con su respectivo IAE para después con estos datos realizar el test de Wilcoxon para validar el nuevo controlador generado.

Figura 4.12 Generación de 30 pruebas mediante optimización de AG



Generación de 30 pruebas mediante optimización de AG para el CGD, Fuente: Marlon Potosí y Grace Ruiz.

Tabla 4.1 Datos IAE del CF y datos IAE del CGD

Prueba	IAE Controlador Fuzzy	IAE Controlador Genético Difuso
1	675.7	642
2	675.7	729.8
3	675.7	642.2
4	675.7	452
5	675.7	685.7
6	675.7	760.5
7	675.7	646.3
8	675.7	658.8
9	675.7	817.7
10	675.7	667.2
11	675.7	655.5
12	675.7	666.3
13	675.7	680.9
14	675.7	626.3
15	675.7	758.6
16	675.7	643.3
17	675.7	615.8
18	675.7	584.6
19	675.7	673.6
20	675.7	907.2
21	675.7	668.2
22	675.7	536
23	675.7	605.9
24	675.7	643.5
25	675.7	673.3
26	675.7	668.8
27	675.7	568.4
28	675.7	637.9
29	675.7	657.3
30	675.7	665.1

Datos IAE del CF y datos IAE del CGD, Fuente: Marlon Potosí y Grace Ruiz.

Por medio del Test de Wilcoxon y los datos obtenidos se prueba si el IAE del CF es idéntico o distinto del IAE del CGD.

Para lo cual se debe considerar las siguientes hipótesis:

- IAE del CF = IAE del CGD, entonces no se ha conseguido la optimización deseada.
- IAE del CF > IAE del CGD, entonces se evidenciaría que se ha optimizado el controlador Fuzzy, debido a que la energía reflejada sería menor en el nuevo sistema.
- IAE del CF < IAE del CGD, entonces se evidenciaría que no se ha optimizado el sistema, debido a que la energía reflejada estaría aumentando y no disminuyendo como el caso anterior.

4.4 TEST DE WILCOXON: CF VS. CGD

En primer lugar, se plantea una Hipótesis nula (H_0) y una Hipótesis alternativa (H_a).

Hipótesis nula

$$\text{IAE de CF} = \text{IAE de CGD}$$

Hipótesis alternativa

$$\text{IAE de CF} > \text{IAE de CGD}$$

Para proceder con el proceso del Test de Wilcoxon se realiza las operaciones pertinentes como la diferencia entre IAE de cada controlador, los valores de la diferencia se ordenan de menor a mayor y luego se asigna un rango, así como se presenta en la Tabla 4.2.

Tabla 4.2 Test de Wilcoxon

N. Pruebas	IAE de CF	IAE de CGD	Diferencia	Ordena	Rangos de orden
1	675,7	642	33,7	2,1	1
2	675,7	729,4	-53,7	2,4	2
3	675,7	642,2	33,5	5,2	3
4	675,7	452	223,7	6,9	4
5	675,7	685,7	-10	7,5	5
6	675,7	760,5	-84,8	8,5	6
7	675,7	646,3	29,4	9,4	7
8	675,7	658,8	16,9	10	8
9	675,7	817,7	-142	10,6	9
10	675,7	667,2	8,5	16,9	10
11	675,7	655,5	20,2	18,4	11
12	675,7	666,3	9,4	20,2	12
13	675,7	680,9	-5,2	29,4	13
14	675,7	626,3	49,4	32,2	14
15	675,7	758,6	-82,9	32,4	15
16	675,7	643,3	32,4	33,5	16
17	675,7	615,8	59,9	33,7	17
18	675,7	584,6	91,1	37,8	18
19	675,7	673,6	2,1	49,4	19
20	675,7	907,2	-231,5	53,7	20
21	675,7	668,2	7,5	59,9	21
22	675,7	536	139,7	69,8	22
23	675,7	605,9	69,8	82,9	23
24	675,7	643,5	32,2	84,8	24
25	675,7	673,3	2,4	91,1	25
26	675,7	668,8	6,9	107,3	26
27	675,7	568,4	107,3	139,7	27
28	675,7	637,9	37,8	142	28
29	675,7	657,3	18,4	223,7	29
30	675,7	665,1	10,6	231,5	30

Operaciones pertinentes para el Test de Wilcoxon, Fuente: Marlon Potosí y Grace Ruiz.

Una vez obtenido los rangos se obtienen los parámetros siguientes:

a) T (+): Suma de rangos correspondientes a diferencias positivas.

$$T (+) = 395$$

b) T (-): Suma de rango correspondientes a diferencias negativas.

$$T (-) = 65$$

c) $T = \text{mín.}(T+, T-) = 65$

d) Estadística Z

$$Z = \frac{T - n \frac{(n+1)}{4}}{\sqrt{n(n+1)(2n+1)/24}} \quad \text{Ec. (4.1)}$$

$$Z = \frac{65 - 30 \frac{(30 + 1)}{4}}{\sqrt{30(30 + 1)(2 * 30 + 1)/24}}$$

$$Z = -3.44$$

A continuación, se trabaja con un nivel de confianza del 95%, por tanto, la estadística se compara con la constante de $Z_a = 1.96$

En la Tabla 4.3 se detalla cómo se toma la decisión para validar el nuevo controlador.

Tabla 4.3 Decisión de validez

Comparación	Hipótesis	Validación
$Z \leq Z_a$	Ho aceptada	X
$Z \geq Z_a$	Ha aceptada	

Comparación de datos obtenidos para validar los controladores, Fuente: Marlon Potosí y Grace Ruiz.

CONCLUSIONES

La caracterización de un sistema permite entender su dinámica, lo que ayuda en la planificación de la estrategia de control a utilizar.

La implementación de un nuevo sensor de nivel se basó principalmente por la facilidad de uso y adaptabilidad al control de nivel propuesto, además por su bajo costo a diferencia de otros y la confiabilidad que brinda por su relación lineal, tal como se identificó en la Figura 3.13 y también a que su principio de funcionamiento es fácil de aplicar.

Existen diferentes diseños de Algoritmos Genéticos, sin embargo, la eficacia para una optimización aceptable se basa en adjudicar valores idóneos en los parámetros principales como el tamaño de la población, número de iteraciones, tiempo de ejecución y sobre todo emplear la función fitness adecuada.

El control en tiempo real obliga el uso adecuado de un sistema embebido apto para el proceso de control, como el controlador AT MEGA 2560 de la placa Arduino Mega que por su capacidad de memoria y poder de cómputo (entre otras características detalladas en la Tabla 3.8) en comparación de otros sistemas embebidos es idóneo para esta estrategia de control.

Por medio del test empleado (95% de nivel de confianza) se determinó que se acepta la hipótesis nula establecida, es decir que el IAE del CF es igual al IAE del CGD, por lo tanto, se interpreta que no hay una diferencia significativa entre los dos controladores para este caso de control de nivel.

Los Algoritmos Genéticos Difusos requieren de una mayor tasa de cálculo computacional, lo cual obliga a trabajar con funciones de pertenencia triangulares para alivianar el tiempo de procesamiento.

Uno de los puntos críticos en la identificación de un sistema de control es la recolección y tratado de datos, debido a que si las muestras no contienen información importante de la evolución del sistema se obtendrán funciones de transferencia incongruentes a la realidad del proceso.

RECOMENDACIONES

Para llevar a cabo el diseño del control Fuzzy convencional se estableció usar funciones de pertenencia triangulares debido a que es recomendado usar funciones simples las cuales simplifican los cálculos matemáticos y no pierden exactitud, además que al estar determinando conceptos borrosos (o difusos) el uso de funciones complejas no proporciona mayor precisión.

Para la identificación de sistemas de control es recomendable emplear un tiempo de muestreo menor o igual a la quinta parte del tiempo de establecimiento de la respuesta del sistema.

En los procesos de nivel en donde se traslade caudal es recomendable antes de encender el sistema chequear que el actuador o bomba esté cebada.

En un proceso real es recomendable establecer un pequeño número de iteraciones para la rápida convergencia del algoritmo de optimización.

Se recomienda cambiar el agua de los tanques y limpiar periódicamente los mismos para evitar el almacenamiento de materiales que puedan obstruir el tubo colocado al sensor y así prevenir lecturas erróneas del sensor de presión diferencial.

Si se va a retirar la conexión física entre los elementos y la placa tomar en cuenta la polarización al momento de volver a reconectar los mismos para evitar daños eléctricos.

Para el adecuado funcionamiento de la planta se recomienda el uso no simultaneo de las dos pantallas de control, debido a que al funcionar estas al instante se genera errores de comunicación en cada una de las pantallas con sus respectivos controladores, por lo que cuenta con varios interruptores, los cuales va a evitar estas fallas.

El tablero principal de control, al igual que el cableado cuenta con un aislamiento y protección contra rasgados, para así evitar el desgaste y maltrato del módulo, por lo que se recomienda tener mucha precaución al momento del llenado o vaciado del tanque de reserva ya que las instalaciones no cuentan con una protección impermeable a líquidos con los que se vaya a trabajar.

REFERENCIAS

- Andrade Sánchez, S. J., & Tipán Jiménez, E. P. (2016). DISEÑO DE UN CONTROLADOR PID DIFUSO PARA LA VARIABLE CAUDAL, DE LA PLATAFORMA DE ENTRENAMIENTO PARA CONTROL CAUDAL, DE LA PLATAFORMA DE ENTRENAMIENTO PARA CONTROL DE PROCESOS CONTINUOS EXISTENTE EN LA UPS, SEDE QUITO - CAMPUS SUR. Quito: Universidad Politécnica Salesiana Sede Quito. Obtenido de <https://dspace.ups.edu.ec/bitstream/123456789/13682/1/UPS-ST002964.pdf>
- Armendáriz, E., Carbo, P., López, J., Matrínez, E., Rocha, E., Rodríguez, J., & Hernández, C. (2014). Ingeniería Bioinspirada. Tamaulipas: Omnia Science.
- Baldovino, R., & Dadios, E. (2015). Control de lógica difusa: diseño de un algoritmo de tabla de matriz asociativa difusa 'Mini' en el control de velocidad del motor. TENCON 2015 - 2015 IEEE Region 10 Conference, 1-4. doi:10.1109/TENCON.2015.7372923
- Boggi, S., Razzitte, A. C., & Fano, W. G. (2014). "Simulación numérica de la permeabilidad magnética aplicada a ferritas utilizando algoritmos genéticos". IEEE Biennial Congress of Argentina (ARGENCON), 202-206. doi:10.1109/ARGENCON.2014.6868496.
- Córdova, F., Cano, A., Dorado, D., & Herrera, J. (2016). "Método computacional para la identificación de funciones de pertenencia en entornos de lógica difusa". 2016 IEEE 11a Conferencia Colombiana de Computación (CCC), 1-8. doi:10.1109 / ColumbianCC.2016.7750790
- De Solminihac T, H., Echaveguren N, T., & Chamorro G, A. (2018). Gestión de Infraestructura Vial (Tercera ed.). Santiago: Ediciones UC.
- Domański, P. D. (2019). Control Performance Assessment: Theoretical Analyses and Industrial Practice (Vols. 245 de Estudios en sistemas, decisión y control.). (S. Nature, Ed.)
- Escaño, J. M., García, J., & Nuevo, A. (2019). Integración de sistemas de automatización industrial. Madrid: Paraninfo.
- Espinoza, E. (2019). DESARROLLO DEL CONTROL ELECTRÓNICO DEL MÓDULO DIDÁCTICO DE NIVEL Y CAUDAL PARA EL LABORATORIO DE ELECTRÓNICA DE LA UPS. Quito: Universidad Politécnica Salesiana.
- Freescale. (10 de 2012). Freescale Semiconductor, Inc. Obtenido de http://www.farnell.com/datasheets/2097509.pdf?_ga=2.37683304.936254475.1594429720-624296604.1594429720
- Glen, A. G. (2016). Computational Probability Applications (Ilustrada ed., Vol. 247 de International Series in Operations Research & Management Science). (Springer, Ed.) Obtenido de <https://books.google.com.ec/books?id=L7uiDQAAQBAJ&pg=PA88&dq=tes>

t+wilcoxon&hl=es-419&sa=X&ved=2ahUKEwiyxvLHzdfqAhVvkeAKHb-
PD78Q6AEwAHoECAAQAq#v=onepage&q=test%20wilcoxon&f=false

- Gómez Ramirez, E., Cortés Ríos, J. C., Ortiz de la Vega, H. A., Melin Olmeda, E. P., & Castillo López, Ó. (2017). El algoritmo de sintonización simple de controladores difusos:(ASSCD). México: De la Salle Ediciones.
- González Palomino, G. (2016). Motores lineales de imanes permanentes: principios de funcionamiento y optimización. (Primera ed.). Santiago de Cali: Universidad Autónoma de Occidente .
- Jiménez, J., Guardado, J., Cabrera, N., Rodríguez, J., & Figueroa, F. (Marzo de 2017). "Sistemas de planificación de expansión de la transmisión utilizando algoritmos genéticos con criterio multiobjetivo". IEEE Latin America Transactions, 15(3), 563-568. doi:10.1109 / TLA.2017.7867609.
- Lopez, J. (diciembre de 2014). Sistemas de control basados en algoritmos genéticos y lógica difusa para la aplicación en procesos industriales. Bogotá: Pontificia Universidad Javeriana.
- Maldonado, C. (2013). Derivas de complejidad: Ciencias sociales y tecnologías convergentes. Bogotá: Universidad del Rosario.
- Mathivet, V. (2018). Inteligencia Artificial para desarrolladores. Conceptos e implementación en C#. Barcelona: ENI.
- MELSEC. (2019). Mitsubishi Products Inc. Obtenido de Mitsubishi Electric Europe B.V: <http://www.farnell.com/datasheets/103812.pdf>
- Morales, D. M., Cappelletti, M., Casas, G., Hasperué, W., & Peltezer y Blancá, E. (2016). "Estudios basados en algoritmos genéticos de celdas solares expuestas a radiación". Congreso Bienal de Argentina IEEE 2016 (ARGENCON), 1-5. doi:10.1109 / ARGENCON.2016.7585262.
- Nedjah, N., & Mourelle, L. d. (2005). Fuzzy Systems Engineering. Theory and Practice. Río de Janeiro: Springer.
- Ogata, K. (2003). Ingeniería de Control Moderna (Cuarta Edición ed.). Madrid: Pearson Educación S.A.
- Pinto, F. J. (2016). Distributed Computing and Artificial Intelligence. 13th International Conference. Advances in Intelligent Systems and Computing., 474, 53-59. doi:10.1007/978-3-319-40162-1_6
- Santos, M., & Miranda, E. (2012). Aplicación de la lógica difusa en el ámbito de las energías renovables. Revista Elementos. Volúmen 2. Número 1, 102-104.
- SIMATIC . (2019). SIMATIC Data Manual. Obtenido de Siemens Products Inc.: https://cache.industry.siemens.com/dl/files/654/1135654/att_24995/v1/Fuzzy_e.pdf

- SYSMAC. (2019). SYSMAC Data Manual. Obtenido de Omron Products Inc.: http://omronkft.hu/nostree/pdfs/plc/c200h/w208e12_c200h-fz001_operation_manual.pdf
- Valdivia Miranda, C. (2012). Sistemas de control continuos y discretos. Madrid: Editorian Paraninfo.
- Valenzuela Hernández, J. G., Montoya Giraldo, O. D., & Giraldo Buitrago, D. (2013). Lógica Difusa Aplicada al Control Local del Péndulo Invertido con Rueda de Reacción. *Scientia Et Technica*, vol. 18, num.4, 623-632. Recuperado el 4 de Julio de 2020, de <https://www.redalyc.org/pdf/849/84929984006.pdf>
- Vásquez Cortés, J. C., Cardona Guio, J. P., & Leal Gómez, J. J. (2015). Automatización Neumática. En Capítulo1. Fundamentos de automatización (págs. 21-22). Bogotá: Ediciones de la U.
- Wiransky, E. (2020). *Genetic Algorithms with Python: Applying genetic algorithms to solve real-world deep learning and artificial intelligence problems*. Birmingham: Packt Publishing Ltd.

ANEXOS

Código implementado controlador genético difuso

```
#include <Nextion.h>
#include <Servo.h>
#include <fuzzy.h>
Servo servoMotor;
fuzzy controlfuzzy; // Creamos un objeto de tipo fuzzy.
////////////////////
double level, Vout, Pre, Vs=5.0;
double aux;
double tol=0.015; //tolerancia-ajusta la medida de presión
int i, den=997; //densidad del agua
double gra=9.8; //gravedad
////////////////////
float MOTOR[] = {0,100}; // Definimos el universo de discurso (rango)de la
variable salida
float NIVEL[] = {0,45}; // Definimos universo de discurso de la variable de
entrada
float VALVULA[] = {0,100}; // Definimos universo de discurso de la segunda
variable de entrada
////////////////////
float P[] = {-1.5,1.998,17.1,'T'}; //Definimos los conjuntos difusos, colocando 'T'
si es triangular, o 'R' si es trapezoidal
float MP[] = {-14.3,10.05,21.96,'T'}; // Conjuntos difusos de entrada
float N[] = {3.677,19.08,29.09,'T'};
float MG[] = {11.01,23.41,36.57,'T'};
float G[] = {27.46,40,41.25,'T'};
////////////////////
void setup() { // Put your setup code here, to run once:
    pinMode(A7,INPUT);
    pinMode(led, OUTPUT);
    nexInit();
```

```

Serial.begin(9600); // Start serial communication at baud=9600

servoMotor.attach(8);

} // End of setup

void loop() { //

aux=0;

//extra=analogRead(A7);

for(i=0; i<10; i++){ //filtro de promedio movil

    aux=aux+(float(analogRead(A7))*5.0/1023.0); //lee pin A0(bytes)-(lo convierte
flotante)-lo transforma en voltaje

    delay(5); //se toma 10 medidas del sensor

}

Vout=aux/10.0; //promedio de las 10

// aplicar formula de la presión

Pre=((Vout-0.04*Vs)/(0.09*Vs))+tol; //kPa

//aplica fórmula del nivel

level=((Pre*1000)/(den*gra))*100; /*1000 para tener P y *100 para tener cm

//regresión lineal

variable1=1.0949*level+0.0356;

lev=variable1;

controlfuzzy.regla_compuesta2(G,CERO,NIVEL,VALVULA,error,derror,MR,
MOTOR,B,tam); //Llamamos a la función para aplicar la regla correspondiente

controlfuzzy.regla_compuesta2(G,VCINCO,NIVEL,VALVULA,error,derror,M
R,MOTOR,B,tam);

controlfuzzy.regla_compuesta2(G,CINCUENTA,NIVEL,VALVULA,error,derr
or,R,MOTOR,B,tam);

```

```
controlfuzzy.regla_compuesta2(G,SCINCO,NIVEL,VALVULA,error,derror,R,
MOTOR,B,tam);
```

```
controlfuzzy.regla_compuesta2(G,CIEN,NIVEL,VALVULA,error,derror,R,M
OTOR,B,tam);
```

```
float res = controlfuzzy.defusi(B,MOTOR,tam); //Se llama a la función para
defusificar el conjunto de salida y obtener el resultado del sistema fuzzy
```

```
zout=res*3.55;
```

```
analogWrite ( 10 , zout ) ;
```

```
analogWrite ( 11 , zout ) ;
```

```
delay(50);}
```

```
////////////////////////////////////
```

```
NexTouch *nex_listen_list[] =
```

```
{
```

```
&b0,
```

```
&b4,
```

```
&h0,
```

```
&page0,
```

```
&page1,
```

```
&page2,
```

```
NULL
```

```
};
```

```
void b0PushCallback(void *ptr)
```

```
{
```

```
if(nivset>0){nivset--;} 
```

```
Serial.print("n3.val=");
```

```
Serial.print(nivset);
```

```
Serial.write(0xff);
```

```
}
```