



UNIVERSIDAD POLITECNICA SALESIANA

SEDE GUAYAQUIL

CARRERA: INGENIERÍA ELECTRÓNICA

Tesis previa a la obtención del título de:

INGENIERO ELECTRÓNICO

CON MENCIÓN EN SISTEMAS INDUSTRIALES

TEMA:

“DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO CONTROLADOR DE TEMPERATURA, SALINIDAD Y OXÍGENO DISUELTO PARA LA SUPERVIVENCIA DEL CAMARÓN USANDO CONTROLADORES P, PI, PID Y LÓGICA DIFUSA.”

AUTORES:

Bryan Ignacio Vera Cabanilla

Edison Marcelo Cadena Mite

TUTORA:

MsC. Monica Miranda Ramos

Guayaquil, marzo del 2019

CERTIFICADO DE RESPONSABILIDAD Y AUDITORIA

Los conceptos desarrollados en el presente proyecto de titulación, el desarrollo del tema, el análisis y respectivas conclusiones de este, corresponden exclusivamente a **Bryan Ignacio Vera Cabanilla** con cedula de identificación 0922538277 y **Edison Marcelo Cadena Mite** con cedula de identificación 0804319010 y el patrimonio intelectual del mismo a la **Universidad Politécnica Salesiana**

Guayaquil, marzo del 2019

(f) Bryan Vera

C.I.0922538277

(f) Edison Cadena

C.I. 0804319010

CERTIFICADO DE RESPONSABILIDAD Y AUTORIA DEL TRABAJO DE TITULACIÓN

Nosotros **Bryan Ignacio Vera Cabanilla** con cedula de identificación **0922538277** y **Edison Marcelo Cadena Mite** con cedula de identificación **0804319010**, manifestamos a nuestra voluntad y cedemos a la **Universidad Politécnica Salesiana** la titularidad sobre los derechos del actual proyecto de titulación en virtud de que somos los autores del tema: **“DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO CONTROLADOR DE TEMPERATURA, SALINIDAD Y OXÍGENO DISUELTO PARA LA SUPERVIVENCIA DEL CAMARÓN USANDO CONTROLADORES P, PI, PID Y LÓGICA DIFUSA.”**, mismo que ha sido realizado para la obtención del título de: **INGENIERO ELECTRÓNICO**, quedando la universidad facultada para ejercer y usar los derechos cedidos anteriormente

En aplicación a lo determinado en la Ley de Propiedad Intelectual Del Ecuador, en nuestra condición de autores nos reservamos los derechos morales de la obra anteriormente citada.

En concordancia, suscribimos este documento en el momento que hagamos entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.

Guayaquil, marzo del 2019

(f) Bryan Vera

C.I.0922538277

(f) Edison Cadena

C.I. 0804319010

CERTIFICADO DE DIRECCION DEL TRABAJO DE TITULACION

Yo, Msc. Mónica Miranda certifico que bajo mi dirección y asesoría fue desarrollado el proyecto técnico de titulación **DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO CONTROLADOR DE TEMPERATURA, SALINIDAD Y OXÍGENO DISUELTO PARA LA SUPERVIVENCIA DEL CAMARÓN USANDO CONTROLADORES P, PI, PID Y LÓGICA DIFUSA**, realizado por el Sr. **Bryan Ignacio Vera Cabanilla** y el Sr. **Edison Marcelo Cadena Mite** para la obtención del título de: **INGENIERO ELECTRÓNICO**

Por medio de la presente certifico que el documento cumple con los requisitos establecidos en el Instructivo para el Estructura y Desarrollo de Trabajos de Titulación para pregrado de la Universidad Politécnica Salesiana.

En virtud de lo anterior, autoriza su presentación y aceptación como una obra autentica y de valor académico

Guayaquil, marzo 2019

(f) Mónica Miranda

DEDICATORIA

La dedicación, tiempo y esfuerzo invertido en este proyecto va dedicado a mi madre Karym Cabanilla Quimi quien me enseñó que el trabajo duro y el esmero llevan a todo a ser al triunfo y satisfacción plena. Cada triunfo en esta vida va dedicado a ti, por el gran ejemplo que me has dado a lo largo de mi vida.

Dedico también este proyecto de grado a mi bisabuela Gloria Panchana Jordan, quien en vida fue mi segunda madre, es inmensurable el amor que me diste en tu vida e incalculable también la falta que me haces.

Llegaremos a saludarnos nuevamente, en la infinidad de las estrellas.

Bryan Ignacio Vera Cabanilla

DEDICATORIA

Este proyecto está dedicado a la memoria de Mercedes Mite Quijje, quien estuvo presente en todo momento en mi vida brindándome su cariño y atención, siendo una segunda madre para mí.

"Llevo en mi memoria cada momento compartido, cada tierna caricia, cada palabra de consuelo. Llevo tu corazón cerca del mío y al sentir una repentina ráfaga de viento sé que eres tú dándome una señal de que sigues conmigo"

Gracias por todo y mucho más tía.

Edison Marcelo Cadena Mite

AGRADECIMIENTO

Agradezco a mi hermana Arlene, por saber comprender que el tiempo apremiaba y que antes de su partida no tuve los días suficientes para despedirme de ella, gracias por comprenderme y confiar en mí.

A mi tía Angela Cabanilla, por ser mi guía académica, mi madre y mi amiga. El amor a las ciencias me lo he ganado por tu dedicación al enseñarme cuando era apenas un niño, no sabría decir que fuese de mí sin tu presencia.

A mi abuela Gloria Quimi Panchana, por su rectitud y disciplina al enseñarme el camino del autoaprendizaje.

A mi prima Angie Molina, por darme sus horas para compartir ideas, metas y apoyo constante, y quien ha sido mi mejor estudiante a lo largo de mi vida.

A mi gran amigo y compañero Edison Cadena, con quien he compartido años de colegio y los mas importantes proyectos en mi carrera universitaria. Gracias por la paciencia que has conllevado a lo largo del tiempo debido a mi terquedad y visión, pero este proyecto no hubiese tenido el mismo resultado sin tu presencia y constancia. Espero que este sea solo el comienzo de proyectos más grandes.

A mi pequeña hermana Andrea y a mi familia, quienes con su amor me dan la fuerza necesaria para seguir adelante cuando todo se ve oscuro.

A los Ingenieros Lelys Ugalde, Byron Lima, Luis Neira y Mónica Miranda por tantos conocimientos impartido y apoyo a lo largo de la carrera.

Bryan Ignacio Vera Cabanilla

AGRADECIMIENTO

Agradezco a mi familia, el pilar fundamental de mi vida, son lo más preciado que tengo.

A mi madre Jenny Mite, por ser mi guía, maestra y amiga, estando siempre para mí mostrándome que, aunque se tenga mil adversidades, todo se puede lograr con esfuerzo y dedicación.

A mis tíos, primos y abuelos gracias por el apoyo y amor incondicional que me brindan, sé que siempre estarán para mí y yo para ustedes.

A mis amigos Andrés, Adrián, Erick, Álvaro, Shakir, Jose por estar presentes desde el colegio hasta la actualidad brindando su amistad y dando una mano cuando se necesita.

A mi gran amigo y compañero Bryan Vera, en estos 12 años de amistad hemos realizados muchos proyectos siendo este el más importante. Nos costó amanecidas y muchas bebidas energizantes, pero valió la pena. Gracias por haber compartido conocimientos, ideas y horas de trabajo continuo en cada uno de ellos. Vendrán más.

A los Ingenieros Byron Lima, Luis Neira y la Ingeniera Mónica Miranda, por compartir sus conocimientos, consejos y aclarar nuestras inquietudes a lo largo de la carrera. Conocimientos que me han servido de mucho en el ámbito profesional.

Edison Marcelo Cadena Mite

RESUMEN

AÑO	ALUMNOS	TUTOR DEL PROYECTO	TEMA DEL PROYECTO
2019	VERA CABANILLA BRYAN IGNACIO CADENA MITE EDISON MARCELO	ING. MONICA MIRANDA RAMOS Msc.	DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO CONTROLADOR DE TEMPERATURA, SALINIDAD Y OXÍGENO DISUELTOS PARA LA SUPERVIVENCIA DEL CAMARÓN USANDO CONTROLADORES P, PI, PID Y LÓGICA DIFUSA

El presente proyecto desarrolla un prototipo cuyo objetivo es el control de las variables de salinidad, oxígeno disuelto y temperatura hacia una unidad remota la cual permite visualizar el estado de las variables mediante una interfaz HMI, los estados de las variables son obtenidas mediante sensores colocados en la piscina.

La interfaz HMI fue realizado en el software LabVIEW, la cual consta de símbolos de actuadores, alarmas de conexión, graficas de valores y generación de archivos históricos. La metodología planteada contempla el proceso de análisis matemático sobre las variables, elección de sensores, formulación de función de transferencia en base a modelo simulado, análisis y comparativa de controladores clásicos y difuso para elección en base a su respuesta en el tiempo.

Finalmente, el proyecto ha sido desarrollado e implementado en base a controladores difusos debido a que son los que mejores se adaptan ante la sensibilidad de las variables.

Palabras Claves: Fuzzy Logic, PID, LabView, Matlab, Sensor de salinidad, Sensor de Oxígeno, Sensor de Temperatura, SSR, Arduino, XBee

ABSTRACT

YEAR	STUDENTS	PROJECT TUTOR	PROJECT THEME
2019	VERA CABANILLA BRYAN IGNACIO CADENA MITE EDISON MARCELO	ING. MONICA MIRANDA RAMOS Msc.	DESIGN AND IMPLEMENTATION OF A TEMPERATURE, SALINITY AND DISSOLVED OXYGEN PROTOTYPE FOR SHRIMP SURVIVAL USING P, PI, PID AND DIFFUSE LOGIC CONTROLLERS

The present project develops a prototype whose objective is the control of the variables of salinity, dissolved oxygen and temperature towards a remote unit which allows to visualize the state of the variables by means of an HMI interface, the states of the variables are obtained by means of sensors placed in the prototype pool.

The HMI interface was developed in LabVIEW software, which consists of actuator symbols, connection alarms, value graphs and generation of historical files.

The proposed methodology contemplates the process of mathematical analysis of the variables, choice of sensors, formulation of transfer function based on simulated model, analysis and comparison of classic controllers and fuzzy controllers for choice based on their response in time.

Finally, the project has been developed and implemented based on the Fuzzy controller because they are the best adapted to the sensitivity of the variables.

Keywords: Fuzzy Logic, PID, LabView, Matlab, Salinity Sensor, Oxygen Sensor, Temperature Sensor, SSR, Arduino, XBee

INDICE GENERAL

1. INTRODUCCIÓN.....	1
1.1. Descripción del problema.....	1
1.2. Antecedentes	1
1.2.1. Introducción al cultivo de camarones.....	2
1.2.2. Control y visualización de la calidad de agua en piscina artesanal.	4
1.3. Importancia y alcance	5
1.3.2. Alcance	6
1.4. Delimitación del problema.....	6
1.4.1 Temporal.....	6
1.4.2. Espacial	6
1.4.3. Académica	6
1.5. Objetivos.....	7
1.5.1. Objetivo general	7
1.5.2. Objetivos específicos.....	7
1.6. Marco metodológico	7
1.6.1. Método deductivo	8
1.6.3. Método inductivo	8
1.6.3. Método experimental.....	8
1.6.3. Método teórico.....	8
1.6.4. Técnicas.....	8
1.7. Impacto.....	8
1.8. Funcionalidad	9

2. MARCO TEÓRICO	9
2.1 Sistema de control automático	9
2.1.1 Sistema de Control en lazo abierto.....	9
2.1.2 Sistema de Control en lazo cerrado.....	10
2.2 Controlador	10
2.3 Señal Pseudo Aleatoria Binaria (PRBS).....	11
2.4 Método de Ziegler Nichols.....	12
2.5 Controlador PID	14
2.5.1 Acción de control Proporcional	14
2.5.2 Acción de control Integral.....	15
2.5.3 Acción de control Derivativa	16
2.6 Lógica Difusa	18
2.6.1 Conjuntos Difusos	18
2.6.2 Funciones de Membresía.....	18
2.6.2.1 Función Triangular	19
2.6.2.3 Función Gama.....	20
2.6.2.4 Función Sigmoidal.....	20
2.6.2.5 Función Gaussiana	21
2.6.3 Operaciones con conjuntos difusos	22
2.6.3.1 Unión.....	22
2.6.3.2 Intersección.....	23
2.6.3.3 Negación o Complemento	23
2.6.4 Modelo lingüístico.....	24

2.6.5 Variable Lingüística	24
2.6.6 Inferencia Difusa	25
2.7. Sistemas Difusos.....	25
2.7.1. Bloque Fusificador.....	26
2.7.2 Máquina de inferencia	26
2.7.3 Bloque Defusificador	26
2.7.4. Centro de Gravedad.....	27
2.7.5. Promedio de los Centros	27
2.7.6. Método del Máximo	27
2.8. Arduino.....	28
2.9. Módulo Xbee	29
2.10. Sensor.....	29
2.10.1. Sensor de Conductividad Atlas Scientific.....	29
2.10.1.1. Circuito para medición de conductividad EZO Atlas Scientific	30
2.10.2. Sensor de Oxígeno disuelto Atlas Scientific	31
2.10.2.1. Circuito para medición de oxígeno disuelto EZO Atlas Scientific	32
2.10.3 Sensor de temperatura waterproof ONE WIRE	32
2.11. Matlab.....	33
2.12. Labview	34
3. DISEÑO.....	34
3.1 Diseño Electrónico	35
3.1.1 Módulo de sensores.....	36
3.2 Diseño Eléctrico	44
3.3 Actuadores utilizados	45

3.3.1 Bombas	45
3.3.2 Electroválvulas.....	46
3.3.3 Resistencias	46
3.3.4 Bombas de Oxígeno	48
3.4 Diseño de Estructura.....	48
3.5 Desarrollo de modelos matemáticos.....	49
3.5.1 Generación de señales PRBS	49
3.5.2 Adquisición de Datos	51
3.5.3 Identificación de plantas de temperatura, salinidad y oxígeno disuelto. ..	52
3.5.4 Cálculo de parámetros para controladores PID.	56
3.5.5 Diseño de controladores clásicos.....	60
3.5.6 Diseño de controladores difusos	64
3.5.6.1 Funciones de membresía.....	64
3.5.6.1.1 Variable Error.....	64
3.5.6.1.2 Variable Setpoint.....	65
3.5.6.2 Diseño de Controlador Difuso para planta de Temperatura	65
3.5.6.3 Diseño de Controlador difuso para planta de Oxígeno Disuelto	68
3.5.6.4 Diseño de Controlador difuso para planta de Salinidad.....	70
3.6 Software de monitoreo y control	73
4. RESULTADOS	79
4.1. Análisis de resultados	79
4.1.1 Planta Temperatura	79
4.2.2 Planta Salinidad.....	83
4.2.3 Análisis de resultado del oxígeno disuelto	85
Conclusiones	88
Recomendaciones	90

Índice de Figuras

Figura 1: Litopenaeus Vannamei (IctioTerm, 2005)	3
Figura 2: Señal PRBS (Mathworks, 2018).....	11
Figura 3: Método de Ziegler Nichols en planta de temperatura (Picuino, 2017)	12
Figura 4: Segundo método Ziegler Nichols aplicado a sistema oscilatorio (UNAM, 2006) ...	13
Figura 5: Respuesta de un sistema de control proporcional con $k_p=50$ (Picuino, 2017)	15
Figura 6: Respuesta de un sistema de control proporcional con $k_p=10$ (Picuino, 2017)	15
Figura 7: Respuesta proceso de nivel de agua, usando controladores P y PI (Smith,1997)	16
Figura 8: Comparación valores de K_d en sistema con oscilación en salida (Picuino, 2017)	17
Figura 9: Función Trapezoidal con a y $b = -\infty$ (Olmo, 2008).....	19
Figura 10: Función Trapezoidal con c y $d = \infty$ (Olmo, 2008)	20
Figura 11: Función Gama (Olmo, 2008)	20
Figura 12: Función Sigmoidal (Olmo, 2008)	21
Figura 13: Función Gaussiana (Olmo, 2008).....	21
Figura 14: Operación Unión (Castillo, 2015).....	22
Figura 15: Operación Intersección (Castillo, 2015)	23
Figura 16: Operación Complemento (Castillo, 2015)	24
Figura 17: Descripción de variable lingüística (Carlos & Motoche, 2012)	25
Figura 18: Sistema difuso	26
Figura 19: Bloque Fusificador	26
Figura 20: Controlador Arduino MEGA 2560 (Arduino,2015)	28
Figura 21:Modulo Xbee (Morillo, 2017).....	29
Figura 22: Sensor Probe K1.0 (Atlas Scientific, 2017)	30
Figura 23: Sensor de temperatura ONE WIRE DS18B20 (DFRobot, 2008)	32
Figura 24: Software MATLAB	33
Figura 25: Ejemplo de Código fuente de Labview (Jolly, 2017)	34
Figura 26: Diagrama de bloques de módulos electrónicos.....	35
Figura 27: Diagrama de bloques de planta Temperatura	35
Figura 28: Diagrama de bloques de planta Salinidad.....	35
Figura 29: Diagrama de bloques de planta Oxígeno Disuelto	36

Figura 30: Sensor de Temperatura instalado en la planta.....	36
Figura 31: Circuito Esquemático adaptador sensor de temperatura (DFRobot, 2018)	37
Figura 32: Sensor de salinidad instalado en la planta	37
Figura 33:Diagrama de circuito conexión de sensor de salinidad (Atlas Scientific, 2018)	38
Figura 34: Soluciones para Sensor de Salinidad (Atlas Scientific, 2018).....	38
Figura 35: Calibración de sensor de salinidad.....	39
Figura 36: Sensor de oxígeno instalado en la planta	40
Figura 37:Diagrama de circuito conexión de sensor de oxigeno (Atlas Scientific, 2018).....	40
Figura 38: Solución Electrolítica de oxígeno Disuelto (Atlas Scientific, 2018)	41
Figura 39 Calibración de sensor de oxigeno	41
Figura 40: Módulo de comunicación XBEE	42
Figura 41: Instalación de Módulo XBEE en la planta	42
Figura 42: Módulo de Relés instalados en la planta.....	43
Figura 43: Relevador de Estado Sólido instalado en la planta	44
Figura 44: Tablero Eléctrico de la planta	45
Figura 45: Bombas de agua utilizada en la planta.....	46
Figura 46: Electroválvula instalada en el sistema de tuberías de la planta	46
Figura 47: Resistencias instaladas en la planta	48
Figura 48:Bombas de oxigeno utilizada en la planta	48
Figura 49: Estructura de la planta con todos sus componentes.....	49
Figura 50: Script para señal PRBS desarrollado en Matlab	50
Figura 52: Señal PRBS generada para planta de Salinidad.....	50
Figura 51: Señal PRBS generada para planta de Temperatura	50
Figura 53: Señal PRBS generada para planta de Salinidad.....	51
Figura 54: Tabla de valores de datos de señal PRBS.....	51
Figura 55: Lectura de sensor de temperatura.....	52
Figura 56: Funciones de transferencias estimadas para planta de Oxigeno	54
Figura 57: Validación de función de transferencia planta temperatura	55
Figura 58: Validación de función de transferencia planta Salinidad	56
Figura 59: Validación de función de transferencia planta Oxigeno	56

Figura 60: Planta de temperatura en lazo abierto con una entrada escalón.....	57
Figura 61: Respuesta de planta de temperatura con una entrada escalón en lazo abierto ..	57
Figura 62: Respuesta de la planta de salinidad con una entrada escalón en lazo abierto ...	57
Figura 63: Respuesta planta oxígeno ante una entrada escalón en lazo abierto	58
Figura 64: Planta de temperatura en lazo cerrado aplicada una entrada escalón	58
Figura 65: Respuesta de la planta de temperatura con $K_p=1,5$	59
Figura 66: Respuesta de la planta de Salinidad con $K_p=1$	59
Figura 67: Respuesta de la planta de Oxígeno con $K_p=1.52$	59
Figura 68: Respuesta controladores Clásicos con setpoint de 30°C. Planta Temperatura ..	60
Figura 69: Valores de K_p y K_d para controlador PD. Planta Temperatura.....	61
Figura 70: Respuesta de controladores Clásicos setpoint de 30 PPT. Planta Salinidad	62
Figura 71: Valores de K_p , K_i y K_d para controlador PID. Planta Salinidad	62
Figura 72: Respuesta de controladores Clásicos setpoint de 8 PPM. Planta de Oxígeno....	63
Figura 73:VI de respuestas en lógica difusa.....	64
Figura 74: Función de membresía entrada Setpoint. Planta de Temperatura	66
Figura 75: Función de membresía salida %. Planta de Temperatura	66
Figura 76: Función de membresía entrada Error. Planta de Temperatura.....	67
Figura 77: VI de simulación de control Difuso	67
Figura 78: Respuesta de simulación de controlador difuso. Planta Temperatura.....	68
Figura 79: Función de membresía entrada Setpoint. Planta de Oxígeno.....	69
Figura 80: Función de membresía entrada Error. Planta de Oxígeno	69
Figura 81: Función de membresía salida % . Planta de Oxígeno.....	69
Figura 82: Respuesta del controlador Fuzzy en planta de Oxígeno	70
Figura 83: Función de membresía entrada Setpoint. Planta de Salinidad	71
Figura 84: Función de membresía entrada Error. Planta de Salinidad	72
Figura 85: Función de membresía salida %. Planta de Salinidad	72
Figura 86: Respuesta de simulación de controlador difuso. Planta Salinidad.....	73
Figura 87: Flujo de datos UART del protocolo ZipBEE (DIGI, 2018)	74
Figura 88:Esquema de transmisión XBEE. (DIGI, 2018).....	74
Figura 89: Adaptador UART XBee.....	75

Figura 90: Topología PAR (Belback, 2017)	75
Figura 91: Etapas de comunicación entre PC y la planta.....	76
Figura 92: Panel frontal del sistema de control y monitoreo.....	77
Figura 93: Visualización de actuadores en panel frontal.	77
Figura 94: Grafica de valor de temperatura	78
Figura 95: Valores de Setpoint de las variables de la planta.....	78
Figura 96: Activación manual de los actuadores de la planta	79
Figura 97: Controlador Fuzzy programado en VI principal de planta	80
Figura 98: Curva temperatura vs tiempo. Valores reales de la planta de Temperatura.....	80
Figura 99: Señal de control dada por el controlador Fuzzy en planta Temperatura	81
Figura 100 Activación de resistencias caloríficas	81
Figura 101 Curva temperatura vs tiempo en perturbación	82
Figura 102: Respuesta señal de control ante perturbación en planta de Temperatura.	82
Figura 103: Activación de resistencias caloríficas en perturbación	83
Figura 104: Curva Salinidad (ppt) vs Tiempo (sg)	83
Figura 105: Activaciones de bomba de agua salada	84
Figura 106 Curva salinidad vs tiempo en perturbación	84
Figura 107 Activaciones de bomba dulce ante perturbación en planta de Salinidad.....	85
Figura 108 Curva oxígeno vs tiempo	85
Figura 109 Respuesta de la señal de control en planta de oxígeno.....	86
Figura 110 Activaciones de bomba de aire.....	86
Figura 111: Curva oxígeno vs tiempo en perturbación	87
Figura 112: Respuesta de la señal de control ante perturbación en planta de Oxígeno	87
Figura 113: Activación de bomba de aire en perturbación	88

INDICE DE TABLAS

Tabla 1: Rangos de concentración de OD en ecosistémicas frecuentes. (Goyenola, 2007) ..	4
Tabla 2: Cálculos de parámetros KP Ti y Td mediante Ziegler Nichols (Jenkins, 2016)	12
Tabla 3: Valores de sintonización del segundo método Ziegler-Nichols. (UNAM, 2006)	13
Tabla 4: Especificaciones Generales Sensor Probe K1.0	30
Tabla 5: Especificaciones Generales Sensor DO	31
Tabla 6: Especificaciones de sensor DS18B20 (DFRobot, 2018)	33
Tabla 7: Comparativa entre controladores clásicos. Planta Temperatura	60
Tabla 8: Comparativa entre controladores clásicos. Planta Salinidad	61
Tabla 9: Comparativa entre controladores clásicos. Planta Oxígeno	63
Tabla 10: Rangos de operación de plantas de temperatura, oxígeno y salinidad	65
Tabla 11: Variables lingüísticas, entrada Setpoint.	65
Tabla 12: Variables lingüísticas, entrada Error	66
Tabla 13: Reglas controlador Difuso Temperatura	67
Tabla 14: Características de control Fuzzy. Planta Temperatura	68
Tabla 15: Reglas controlador Difuso Oxígeno	69
Tabla 16: Características de control Fuzzy. Planta Oxígeno	70
Tabla 17: Variables lingüísticas, entrada Setpoint	71
Tabla 18: Variables lingüísticas, entrada Error	71
Tabla 19: Conjunto de reglas para controlador difuso. Planta de Salinidad	72
Tabla 20: Características de control Fuzzy. Planta Salinidad	73

1. INTRODUCCIÓN

1.1. Descripción del problema

Los camarones son criaturas delicadas a los cambios en su entorno, por lo tanto la calidad del agua en su ecosistema juega un papel fundamental en el desarrollo de los mismos debido a que las malas condiciones de las variables internas pueden conllevar a efectos de estrés y consecutivamente aumentar la tasa de mortandad de estos seres.

Se toma como experiencia la visita técnica a la empresa AquaTropical ubicada en Mar Bravo, Prov. La libertad. AquaTropical es un laboratorio que cuenta con las instalaciones e infraestructura necesaria para el proceso de producción de postlarvas de camarón *Litopenaeus Vannamei*. Durante la visita dirigida por el Biólogo Juan Macías, supervisor general de todo el proceso de producción, se especificaron las 3 variables más importantes a considerar: Temperatura, salinidad y nivel de oxígeno, los cuales están presentes en todos los procesos y los cuales se consideran cruciales para la vida óptima de los crustáceos.

Por lo tanto, los subprocesos elegidos para este proyecto: recepción de reproductores, maduración, reproducción y desove; mantienen las variables de manera que, la temperatura se mantiene en un margen de 26-29 °C en cada una de las etapas. La salinidad idónea ronda los 30 ppt (Partes por mil) conservando siempre un rango de nivel de oxígeno de 6 a 10 mg/L en todos los estanques para evitar la muerte de los crustáceos.

1.2. Antecedentes

En Ecuador, la acuicultura representa una de las actividades más importantes del sector productivo desde su origen en 1968. Actualmente, la Organización de las

Naciones Unidas para la Alimentación (FAO) define a la acuicultura como la explotación de organismos acuáticos, incluyendo peces, moluscos, crustáceos y plantas acuáticas.

En nuestro país, el producto acuícola principal de explotación es el camarón. La industria camaronera convirtió al Ecuador en un importante productor y exportador en el mercado internacional, llevando a desarrollar mecanismos y sistemas automatizados que intervengan en el proceso de cría para asegurar calidad e incremento en la producción.

1.2.1. Introducción al cultivo de camarones

El camarón es una especie marina de gran relevancia dentro del comercio exterior, Ecuador es el mayor productor de camarón en cautiverio del hemisferio Occidental y es también el segundo productor a escala mundial, luego de Tailandia; el 96% de la producción camaronera proviene del cultivo y el 4% de la pesca artesanal (Aspectos básicos del cultivo de camarón en el Ecuador, 2015). El cultivo de camarón consta de las siguientes etapas: recepción, maduración, reproducción, desove y cría, que se han logrado reproduciendo en cautiverio los procesos biológicos naturales de estos crustáceos. (Cortez, 2006).

Dada las condiciones meteorológicas la región costera del Ecuador posee dos épocas climatológicas: una época lluviosa de invierno de altas temperaturas (alrededor de 30°C) y una época seca de verano de temperaturas más bajas (alrededor de 25°C). Se debe mantener una temperatura constante para la eficacia del crecimiento del camarón, dado que el cambio brusco de esta puede presentar variaciones en la densidad, viscosidad, solubilidad de gases y al cambio de oxígeno de ingesta de los crustáceos. Dado que cada especie de camarón tiene diferentes capacidades para resistir rangos de temperaturas, se establece la familia *Litopenaeus Vannamei* el cual tiene un rango óptimo de 28 a 30 C. La descarga de

los ríos en las costas produce aguas estuarinas lo que implica la existencia de un alto rango de salinidades.



Figura 1: Litopenaeus Vannamei (IctioTerm, 2005)

De modo que la salinidad es un tema importante para el correcto crecimiento de los camarones, esta es considerada como concentración total de todos los iones disueltos expresados en partes por mil (ppt) en el agua. La salinidad está determinada principalmente por sólidos disueltos, como: fosfatos, bicarbonatos, sulfatos, nitratos y otros. Dado que el agua de mar es la fuente utilizada la misma tiene un rango de 25 a 36 ppt durante los doce meses de año.

El oxígeno disuelto es parámetro vital también para este ecosistema la variable más crítica para la calidad del agua de un estanque, dado que sin una buena provisión de oxígeno los organismos pueden ser vulnerables a enfermedades, o morir.

La solubilidad del oxígeno en el agua depende de la temperatura, a mayor temperatura menos oxígeno se disuelve. Por otra parte, si el agua está contaminada, tiene muchos microorganismos, materia orgánica alta, el oxígeno disuelto tiende a disminuir.

La saturación de oxígeno también descende a medida que incrementa la salinidad, pero este efecto no es muy alto en los rangos de salinidad en acuicultura de agua

dulce. Con salinidades altas el agua retiene menos oxígeno disuelto que con salinidades bajas (Boyd, Consideraciones sobre la calidad de agua y suelo en cultivos de camarón, 1992)

[OD] mg/L	Condición	Consecuencias
0	Anoxia	Muerte masiva de organismos aerobios.
0-5	Hipoxia	Desaparición de organismos y especies sensibles.
5-8	Aceptable	[OD] adecuadas para la vida de la gran mayoría de especies de peces y otros organismos acuáticos.
8-12	Buena	
>12	Sobresaturada	Sistemas en plena producción fotosintética.

Tabla 1: Rangos de concentración de OD y consecuencias ecosistémicas frecuentes. (Goyenola, 2007)

1.2.2. Control y visualización de la calidad de agua en piscina artesanal.

Las variables por considerar para el sistema son: temperatura, oxígeno disuelto y salinidad, puesto que son primordiales para el óptimo crecimiento del crustáceo.

En la actualidad las camaroneras artesanales realizan la lectura y registro de variables de manera manual, muchas de estas usando medidores analógicos. Estos datos sirven de referencia para un registro total diario, con el afán de precautelar las condiciones del crecimiento del crustáceo. El monitoreo de las

variables primordiales tiene como finalidad efectivizar el control de las variables físicas y obtener una mejor producción.

De modo que se debe mantener un constante censo de las mismas para un óptimo control, esto sin embargo significaría un mayor coste de horas de trabajo para los operadores al tener que realizar el registro en múltiples piscinas y está ligado al factor de error humano.



Figura 2: Piscinas artesanales en AQUATROPICAL.

1.3. Importancia y alcance

1.3.1. Importancia

La propuesta de este proyecto es brindar al sector acuícola, un instrumento de control y registro fiable, rápido y preciso.

Basado en la comparativa de diferentes algoritmos controladores, desarrollado en un sistema de código abierto y con la facilidad que ofrece la comunicación inalámbrica.

1.3.2. Alcance

El proyecto titulado como “Diseño e implementación de un prototipo controlador de temperatura, salinidad y oxígeno disuelto para la supervivencia del camarón usando controladores P, PI, PID y lógica difusa.”, se basa en analizar el algoritmo de control más eficaz que presenta el sistema ante influencias externas que pueden afectar su comportamiento. Se va a diseñar un circuito de control tal que mediante sensores de temperatura, salinidad y oxígeno disuelto podremos obtener el valor real que circula por la piscina prototipo, la tarjeta Arduino realizará la adquisición de datos para posteriormente enviar los mismo por medio de un módulo inalámbrico XBEE y se receptorá en el software Matlab, para analizar el sistema y desarrollar las plantas de control para luego realizar el monitoreo por medio del software LabVIEW.

1.4. Delimitación del problema

1.4.1 Temporal

Este proyecto de titulación tuvo una duración de 12 meses contados desde la fecha de aprobación del tema.

1.4.2. Espacial

Este proyecto de titulación tiene como lugar de destino las instalaciones de la Universidad Politécnica Salesiana sede Guayaquil.

1.4.3. Académica

El modelado del sistema de control y monitorización permite comprender mejor el comportamiento de la planta analizada y adquirir nuevos conceptos en base a la investigación de modelos matemáticos que juntos a los adquiridos en las aulas de la Universidad Politécnica Salesiana sede Guayaquil en las materias de teoría de control, sensores y transductores, informática industrial, circuitos eléctricos industriales, física permitirán tener una mejor comprensión sobre estos sistemas y ayudarán a entender por qué son fundamentales en la era electrónica moderna.

1.5. Objetivos

1.5.1. Objetivo general

Diseñar diferentes modelos de controladores P, PI, PID y lógica difusa para el control de temperatura, salinidad y oxígeno disuelto utilizando como controlador una tarjeta Arduino y la tecnología inalámbrica Xbee, y obtener un análisis comparativo entre los diferentes controladores.

1.5.2. Objetivos específicos

- Identificar el modelo matemático simulado de las variables: salinidad, temperatura y oxígeno disuelto, usando el software MATLAB.
- Desarrollar e implementar la adquisición de datos en la placa Arduino MEGA para las variables de salinidad, oxígeno disuelto y temperatura.
- Diseñar un sistema de lógica difusa en el software Labview para el control simulado de cada una de las variables: temperatura salinidad y oxígeno disuelto.
- Diseñar controladores clásicos P, PI, PD, PID haciendo uso del software Matlab para el control simulado de las variables de salinidad oxígeno disuelto y temperatura.
- Implementar en el prototipo el controlador elegido para la variable temperatura en base al análisis simulado comparativo.
- Realizar un análisis simulado comparativo en base a los controladores P, PI, PID y lógica difusa para las variables salinidad, oxígeno disuelto y temperatura.

1.6. Marco metodológico

Para el desarrollo del presente proyecto se emplean las siguientes técnicas de investigación.

1.6.1. Método deductivo

Este método permitió conseguir un enfoque total de los temas investigados en el proyecto de titulación, con este método se pudo relacionar los conocimientos adquiridos en el curso de la carrera con los investigados para el proyecto.

1.6.3. Método inductivo

Este método permitió que la observación de un fenómeno o problema particular, en el presente caso la necesidad de mantener una temperatura, salinidad y nivel de oxígeno constante en cada etapa, en base del registro del prototipo se puede generalizar para obtener un sistema lo suficientemente robusto para una industria de gran escala.

1.6.3. Método experimental

Este método permitió que, en base a las pruebas pertinentes del prototipo, tanto hardware como software, se desarrolló el proceso de desarrollo del sistema, un enfoque total de los temas investigados en el proyecto de titulación.

1.6.3. Método teórico

En base a este método se analizó el levantamiento de datos de los diferentes valores que se encuentran en los sensores, el desarrollo de funcionamiento de los diferentes sensores para la placa Arduino MEGA, y el proceso de crecimiento del crustáceo.

1.6.4. Técnicas

Se analizó los valores de entrada y salida del sistema durante un periodo de tiempo para poder obtener un análisis de datos comprensible y posterior sistema robusto.

1.7. Impacto

El proyecto de titulación detallado aquí pretende analizar la innovación en la inclusión de sistema autónomo, robusto (en base a la toma de diferentes controladores),

inalámbrico, y de acceso totalmente abierto, dado que el hardware y software a utilizar puede ser moldeado a conveniencia del investigador.

1.8. Funcionalidad

El estudio de los sistemas de control para pequeñas industrias es un tema emergente dado que ayuda a la rentabilidad de la producción haciendo uso de sistemas especializados sin embargo el alto coste de estos es muchas veces rechazados o no se dan por el nivel de conocimiento previo que se necesita.

Con este proyecto se podrá investigar y analizar un sistema haciendo uso de la tecnología Open Source, además permitirá desarrollar en Matlab y LabVIEW la programación necesaria para hacer una herramienta de control robusta y rápida.

2. MARCO TEÓRICO

2.1 Sistema de control automático

Un sistema de control se puede definir como el conjunto de técnicas o métodos, los cuales son ejecutados en un determinado proceso para asegurar el normal funcionamiento de las variables implicadas dentro de los rangos de operación de este. (Sanjuán, 2007)

Entre estos sistemas se pueden diferenciar dos clases: Sistemas de control en lazo abierto y Sistemas de control en lazo cerrado.

2.1.1 Sistema de Control en lazo abierto

En un sistema de control de lazo abierto, la salida de este no influye sobre el comportamiento de la señal de control, es decir que el sistema no recibe ninguna realimentación por parte de la salida resultante para ser comparada con la señal de

entrada evitando que el sistema pueda corregir su funcionamiento ante una perturbación. (Ogata, 2010)

Estos sistemas de control son utilizados en procesos en los cuales se conoce la relación entrada – salida sin la presencia de perturbaciones externas que afecten el funcionamiento normal del mismo, a cada salida se le asigna una condición de operación estándar. Generalmente procesos en los cuales el sistema de control funcione por ciclos de tiempo definidos se consideran sistemas en lazo abierto un ejemplo de esto es el control de semáforos en un intercambiador de tráfico.

2.1.2 Sistema de Control en lazo cerrado

También llamados Sistemas de control realimentados. En un sistema de control de lazo cerrado, la salida influye directamente sobre el comportamiento de la señal de control, el sistema recibe una realimentación por parte de la salida siendo esta comparada con la entrada generando una señal de error, la cual es ingresada al controlador con el propósito de reducir dicho valor y llegar a la referencia deseada. (Ogata, 2010)

Estos sistemas de control son utilizados en procesos donde se suelen presentar perturbaciones impredecibles que puedan afectar al funcionamiento normal de los mismos, corrigiéndose de forma autónoma si se ven afectados por perturbaciones internas o externas en sus parámetros, haciéndolos más estables en comparación a los sistemas de lazo abierto. El control de temperatura de un cuarto es un ejemplo de control en lazo cerrado, el termostato, mide la temperatura de la habitación y la compara con la temperatura de referencia aumentando o disminuyendo la intensidad del aire hasta llegar a su valor deseado.

2.2 Controlador

Un controlador es un instrumento el cual mediante un algoritmo de control cargado previamente en su memoria interna toma como entrada el valor de la señal de error

del sistema y se encarga de devolver una señal de control adecuada para mejorar el funcionamiento del sistema, permitiendo que este llegue a su setpoint, disminuyendo el valor de error.

Una especificación requerida para que un controlador funcione de manera óptima es que el lazo de realimentación siempre esté cerrado, es decir, que el controlador reciba continuamente la información transmitida dentro de este.

2.3 Señal Pseudo Aleatoria Binaria (PRBS)

Para la estimación de sistema de identificación es muy común hacer uso de una señal PRBS dado que la misma crea un gran espectro de frecuencias al actuador, generando una prueba de estrés en diferentes rangos.

En base a estas señales es posible un modelamiento experimental, es ampliamente utilizado en sistemas donde los modelos matemáticos contienen estructuras analíticas complejas y no son fiables en su totalidad debido a la respuesta variable del sistema. (Charlton, 1968)

Esta señal produce valores binarios (0 o 1), la cual es receptada por el sistema y por medio del actuador reproduce estos valores que luego serán tratados por el sistema de identificación

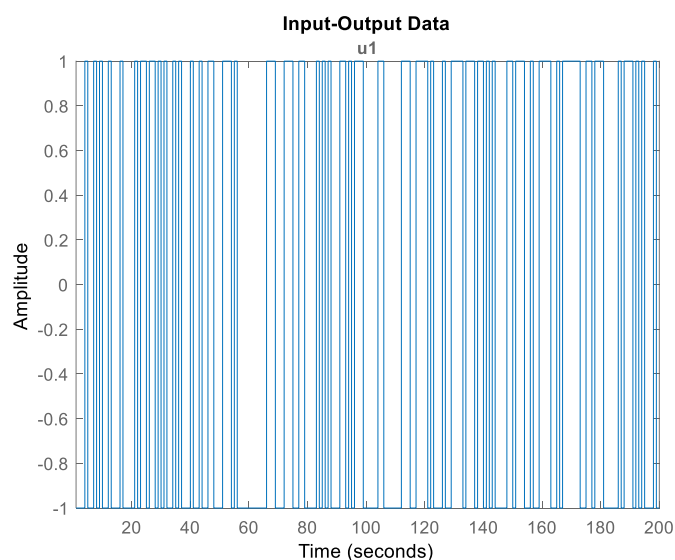


Figura 2: Señal PRBS (Mathworks, 2018)

2.4 Método de Ziegler Nichols

Es un método basado en la respuesta transitoria de una señal paso, con el cual se puede adaptar un controlador PID de forma empírica. Representa una “S” alargada sin presentar sobrepaso en lazo abierto, siempre que el sistema no contenga integradores o polos conjugados complejos. (Jenkins, 2016)

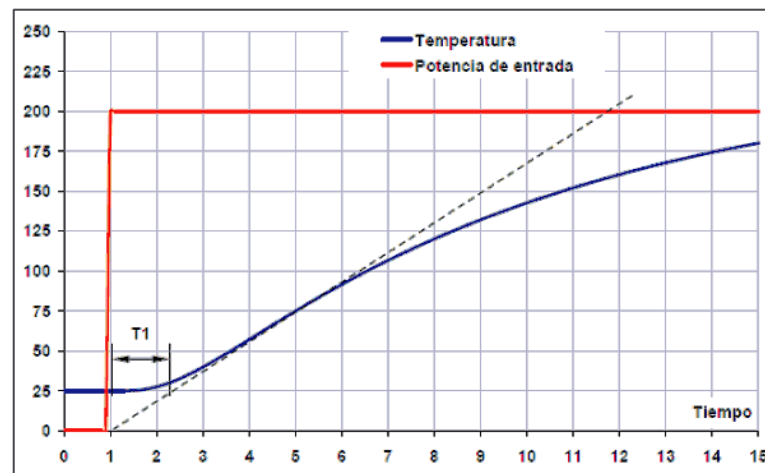


Figura 3: Método de Ziegler Nichols en planta de temperatura (Picuino, 2017)

Se puede observar en la imagen la curva de reacción, esta se caracteriza por dos constantes, tiempo de delay o tiempo muerto “L” y tiempo constante o tiempo de subida “T”, que son determinadas en base a una línea tangencial dibujada en el punto de inflexión de la curva y encontrando las intersecciones de la tangente con el eje de tiempo y el valor de estado estable.

Controlador	K_p	T_i	T_d
P	T/L	∞	0
PI	$(0.9) T/L$	$L/0.3$	0
PID	$(1.2) T/L$	$2L$	$0.5L$

Tabla 2: Cálculos de parámetros K_p T_i y T_d mediante Ziegler Nichols, primer método (Jenkins, 2016)

En base a las fórmulas citadas en la tabla 2 se puede analizar los respectivos valores de los controladores.

Si el sistema posee oscilaciones continuas, se emplea el segundo método de Ziegler Nichols, en el cual se debe suprimir todo efecto de parte integral y derivativa. Luego, haciendo uso de la ganancia K_p , se espera que el sistema mantenga oscilaciones continuas.

Este valor de la ganancia que logra desarrollar una ganancia continua tiene el nombre de ganancia crítica K_{cr} , el cual tiene en su haber un periodo crítico P_{cr} . (UNAM, 2006)

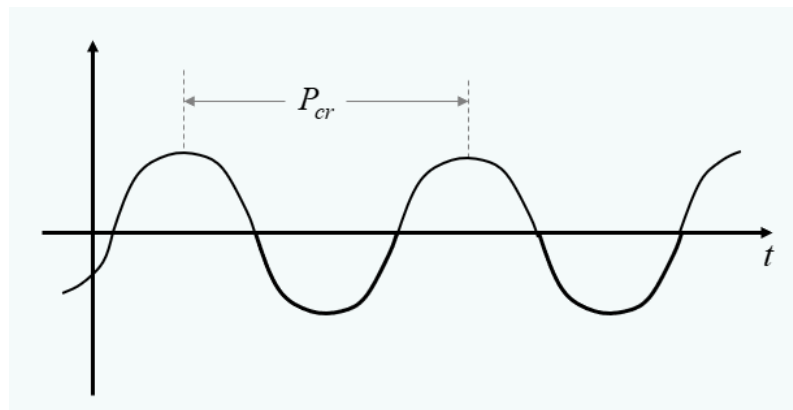


Figura 4: Segundo método Ziegler Nichols aplicado a sistema oscilatorio (UNAM, 2006)

Una vez encontrados los valores K_{cr} y P_{cr} se pueden calcular los valores de las ganancias k_p , k_i , k_d partiendo de la siguiente tabla.

Controlador	K_p	T_i	T_d
P	$0.5 K_{cr}$	∞	0
PI	$0.45 K_{cr}$	$0.5 P_{cr}$	0
PID	$0.6 K_{cr}$	$0.5 P_{cr}$	$0.125 P_{cr}$

Tabla 3: Valores de sintonización del segundo método Ziegler-Nichols. (UNAM, 2006)

2.5 Controlador PID

El controlador PID es uno de los controladores universalmente aceptado en el campo del control de la industria. Esta popularidad puede ser atribuida a su robustez en el rango operativo de control y simplificada funcionalidad.

Los controladores PID se basan en la combinación de sus parámetros proporcional, derivativa e integral mediante un algoritmo de control

$$u(t) = K_c(\epsilon(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt}) = P + I + D \quad (1)$$

Como se especifica en la anterior fórmula (1), el algoritmo PID consiste en la suma de tres acciones; proporcional(P), integral (I) y derivativo (D) los cuales se varían para obtener una respuesta óptima.

2.5.1 Acción de control Proporcional

La acción proporcional devuelve una señal de control que es proporcional a la entrada del controlador (señal de error)

$$u(t) = K_p * e(t) \quad (2)$$

Donde:

$u(t)$: salida del controlador

K_p : ganancia proporcional

$e(t)$: señal de error

El objetivo del control proporcional es aproximar el error de estado estable a cero, modificando el valor de la ganancia K_p . Si se fija K_p en un valor alto puede generar valores superiores al valor seteado causando un oscila miento en la señal de salida

(overshoot) hasta llegar al setpoint (figura 5), en el caso de fijar un valor muy bajo se obtendrá una respuesta lenta ante variaciones de valores de setpoint (figura 6) ya que el control proporcional no toma en cuenta el tiempo en que se tarda el sistema en llegar a su valor seteado (setting time) (Villajulca, 2011)

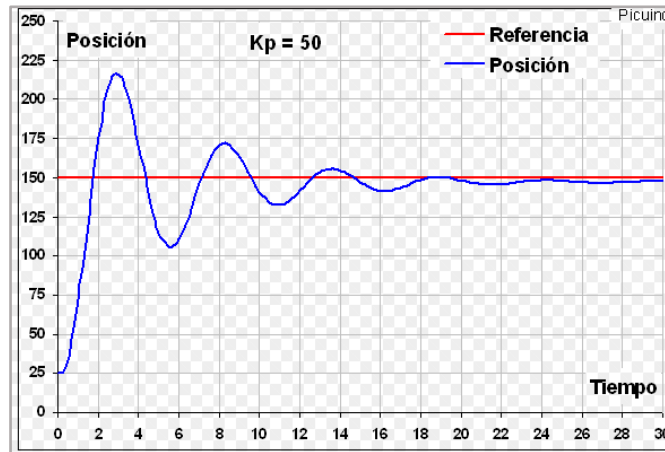


Figura 5: Respuesta de un sistema de control proporcional con $k_p=50$ (Picuino, 2017)

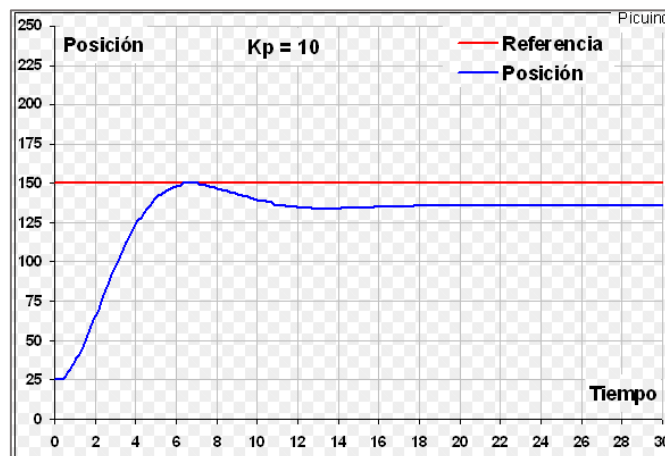


Figura 6: Respuesta de un sistema de control proporcional con $k_p=10$ (Picuino, 2017)

2.5.2 Acción de control Integral

El objetivo del control integral es calcular la integral de la señal de error para eliminar el error en estado estable producido por perturbaciones, a más de disminuir el valor de la desviación de la referencia (offset) que se produce cuando se aplica un control

proporcional. Generalmente estos dos controladores se usan juntos convirtiéndose en un control PI el cual posee las ventajas de ambos controladores teniendo la siguiente ecuación.

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(\tau) d\tau \quad (3)$$

Donde:

$u(t)$: salida del controlador

K_p : ganancia proporcional

T_i : Tiempo integral

$e(t)$: señal de error

El tiempo integral T_i ajusta la acción integral, la unidad de medida de este es minutos, a menor valor de T_i el controlador integrará más rápido.

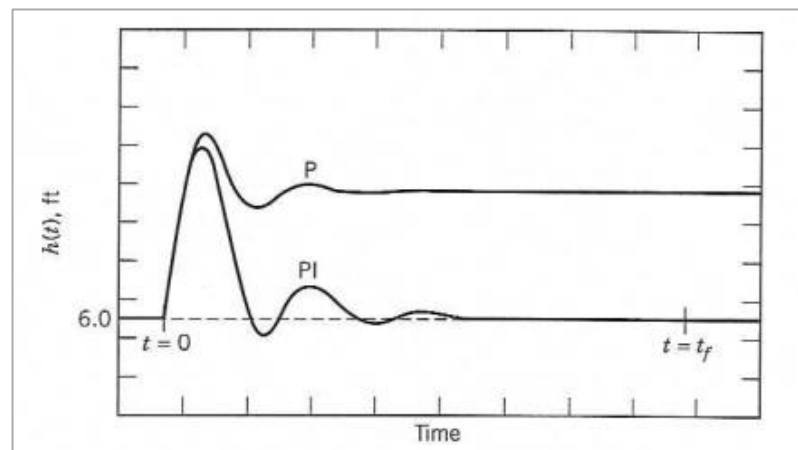


Figura 7: Respuesta de un proceso de nivel de agua, usando un controlador P y PI (Smith-Corripio, 1997)

2.5.3 Acción de control Derivativa

La acción derivativa, como su nombre lo indica, actúa sobre la derivada del error, la cual es considerada la velocidad de este, esta va a depender de la inercia propia del sistema. Si un sistema tiene una velocidad alta (ej el accionamiento de un cilindro

neumático), al ser enviada la señal de control para que esta llegue a su setpoint, por la naturaleza del sistema este responderá de una manera rápida, superando el setpoint deseado, produciéndose una oscilación alrededor del punto de referencia sin que este llegue a estabilizarse. (Picuino, 2017)

El objetivo del control derivativo es estabilizar la respuesta del sistema, evitando demasiadas oscilaciones. Se rige por la siguiente ecuación (4).

$$u(t) = K_d * \frac{de(t)}{dt} \quad (4)$$

Donde:

$u(t)$: salida del controlador

K_d : constante derivativa

$\frac{de(t)}{dt}$: derivada de la señal de error.

En la siguiente figura (figura 5) se pueden observar el cambio en la respuesta del sistema ante un valor de K_d , minorando las oscilaciones y estabilizando un poco más el sistema.

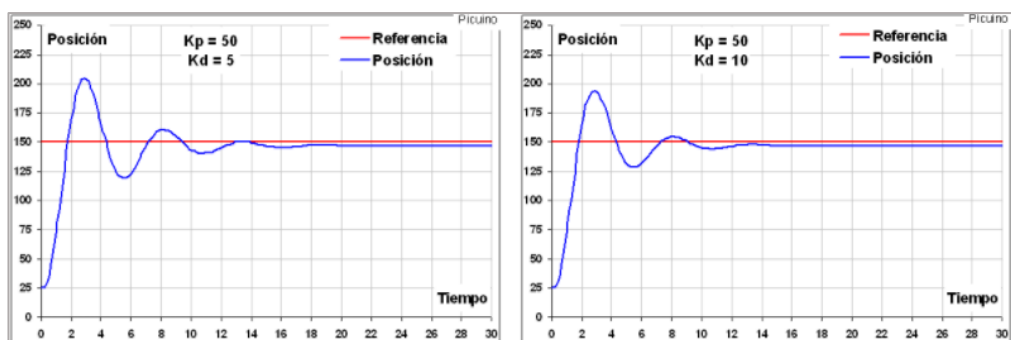


Figura 8: Comparación de valores de K_d en un sistema con oscilación en su salida (Picuino, 2017)

Estas tres acciones en conjunto brindan al controlador PID estabilidad, respuesta rápida y un error en estado estable bajo.

2.6 Lógica Difusa

La lógica difusa es considerada un sistema lógico destinado a suministrar un modelo de razonamiento similar al pensamiento humano, siendo este más aproximado que exacto.

Este modelo se aplica para conjuntos difusos en los cuales las definiciones de “Verdadero o Falso”, “0 o 1”, “Si o No”, son muy limitadas para definir si un elemento posee o no posee una propiedad. Por ejemplo: La cantidad de agua vertida en un vaso para que este se considere lleno o vacío.

El modelado de la información se realiza a base de reglas lingüísticas que aproximan una función de membresía basándose en la relación entre la entradas y salidas del sistema devolviendo un valor de salida dentro del rango entre 0 a 1. (Ponce, 2010)

2.6.1 Conjuntos Difusos

Un conjunto es un grupo de elementos que cumplen una propiedad determinada. En la lógica clásica un elemento de un universo dado pertenece o no pertenece al conjunto, esto cambia en la lógica difusa. Un conjunto difuso permite la pertenencia parcial de sus elementos pudiendo estos pertenecer a uno o más conjuntos con distintos grados de pertenencia.

Estos grados se definen mediante su función de membresía $f_A(x)$ asociada al conjunto, la cual proporcionará un grado de pertenencia del elemento x al conjunto A , dentro del intervalo $[0,1]$. (Benito & Isabel, 2010)

2.6.2 Funciones de Membresía.

Como se habló anteriormente, la función de membresía indica el grado en que cada elemento del universo dado pertenece a un conjunto. Para un conjunto con límites normalizados la función de membresía está perfectamente definida. No sucede así con un conjunto difuso ya que su función dependerá de muchos factores, entre ellos, el universo definido, la aplicación, el diseñador, etc.

Al momento de determinar la función de membresía se eligen funciones más sencillas de tal forma que los cálculos se puedan simplificar, las más comunes son las siguientes:

2.6.2.1 Función Triangular

Esta función se asemeja a dicha figura geométrica, teniendo una pendiente positiva hasta que alcanza un punto máximo, cambiando su pendiente a negativa. No necesariamente debe ser asimétrica.

2.6.2.2 Función Trapezoidal

La función Trapezoidal está definida por sus dos límites inferiores y dos superiores (a, b, c, d) tales que $a < b < c < d$. Esta es la forma general de la función. Si se modifican los valores de los límites se pueden obtener diferentes variantes de dicha función. Si los límites b y c tienen el mismo valor, la función se transforma en una triangular.

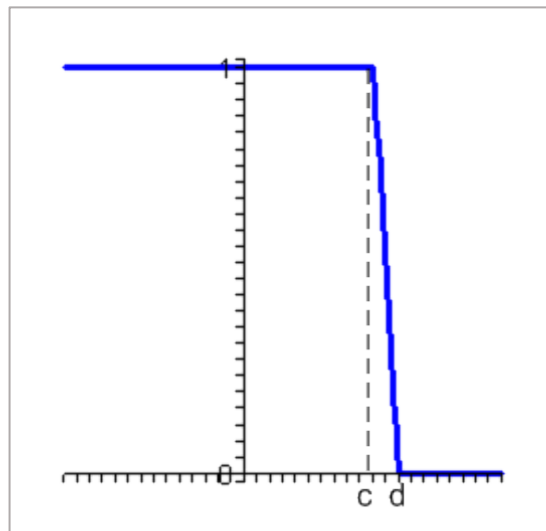


Figura 9: Función Trapezoidal con a y $b = -\infty$ (Olmo, 2008)

Si los límites “ a, b ” o “ c, d ” toman valores de $-/+$ infinito respectivamente se obtendrán las siguientes funciones trapezoidales.

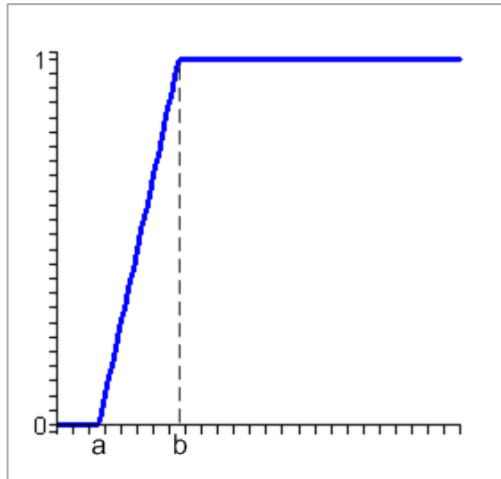


Figura 10: Función Trapezoidal con c y $d = \infty$ (Olmo, 2008)

2.6.2.3 Función Gama

Posee un límite inferior “a”, un valor “k”>0 y una asíntota en 1. Cuanto más grande es el valor de k el crecimiento de la función es más rápido.

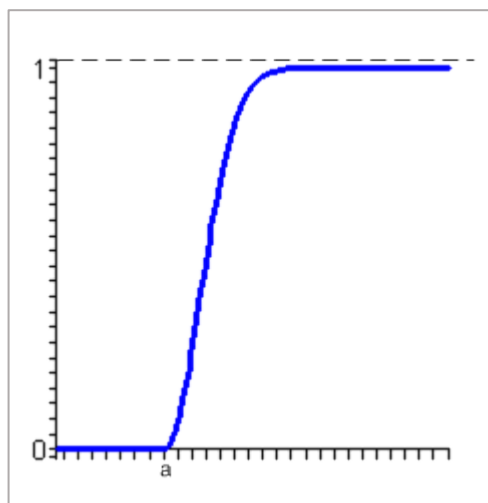


Figura 11: Función Gama (Olmo, 2008)

2.6.2.4 Función Sigmoide

Definida por 2 límites, superior e inferior (a y b) y un punto de inflexión $a < m < b$ donde:

$$m = \frac{(a+b)}{2} \tag{5}$$

La función posee un crecimiento más lento entre mayor sea la distancia entre sus límites a , b .

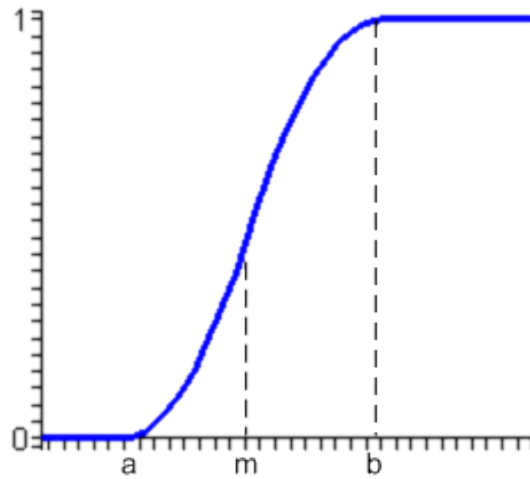


Figura 12: Función Sigmoide (Olmo, 2008)

2.6.2.5 Función Gaussiana

Describe la Campana de Gauss, con su valor medio " m " y un parámetro " k ", cuanto más grande sea el valor de k , más estrecha es la campana.

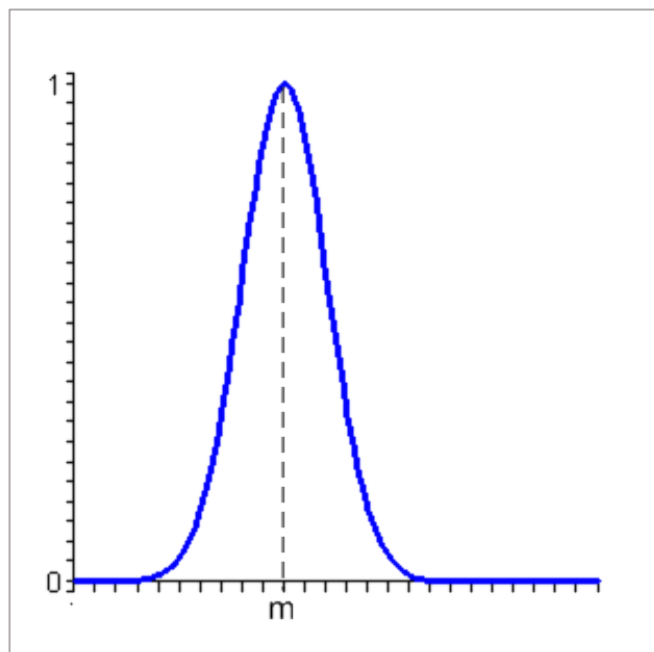


Figura 13: Función Gaussiana (Olmo, 2008)

2.6.3 Operaciones con conjuntos difusos

Como las operaciones entre conjuntos clásicos, es posible realizar operaciones con conjuntos difusos. Se pueden definir tres operaciones básicas: unión, intersección y complemento o negación. Estas pueden estar relacionadas a las operaciones lógicas (OR, AND, NOT) pero ya que se está trabajando con variables difusas, estas deben ser redefinidas conocidas como normas triangulares o normas T.

Dado dos conjuntos difusos A y B asociados a una variable lingüística X, cuando se realiza una operación entre estos el resultado es un nuevo conjunto difuso

2.6.3.1 Unión

La unión entre dos conjuntos difusos (A y B) genera un nuevo conjunto difuso C el cual se obtiene aplicando la operación T-Conorma.

$$C = A \cup B \quad (6)$$

$$\mu_C(x) = T(\mu_A(x), \mu_B(x)) \quad (7)$$

$$\forall x \in U \quad (8)$$

$$C = \max(\mu_A(x), \mu_B(x)) \quad (9)$$

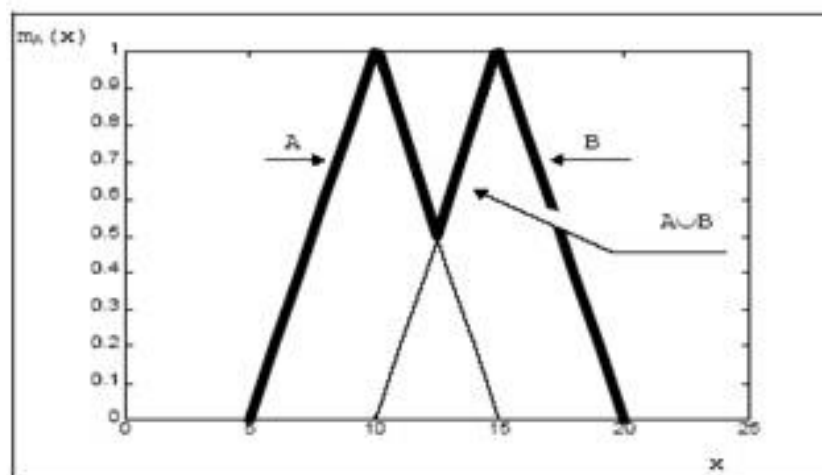


Figura 14: Operación Unión (Castillo, 2015)

2.6.3.2 Intersección

La intersección entre dos conjuntos difusos (A y B) genera un nuevo conjunto difuso C el cual se obtiene aplicando la operación T-Norma.

Se considera que un elemento pertenece al nuevo conjunto si este perteneció a ambos conjuntos anteriores.

$$C = A \cap B \quad (10)$$

$$\mu_C(x) = T(\mu_A(x), \mu_B(x)) \quad (11)$$

$$\forall x \in U \quad (12)$$

$$C = \min(\mu_A(x), \mu_B(x)) \quad (13)$$

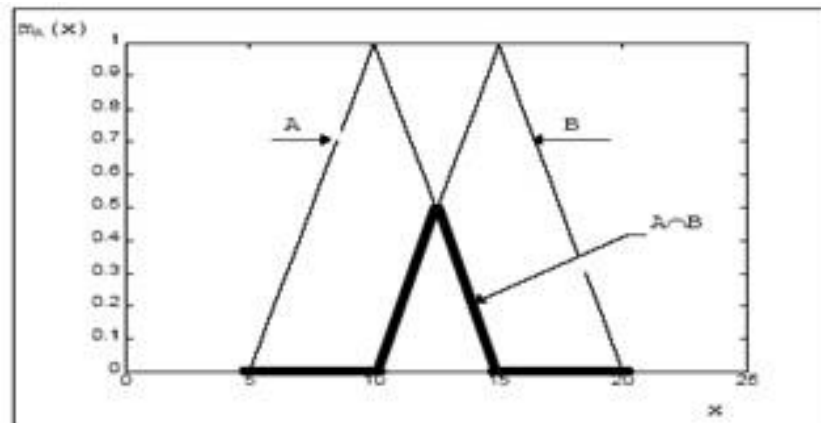


Figura 15: Operación Intersección (Castillo, 2015)

2.6.3.3 Negación o Complemento

El complemento de un conjunto difuso (A) genera un nuevo conjunto difuso (\bar{A}) el cual se define por:

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \quad (14)$$

$$\forall x \in U \quad (15)$$

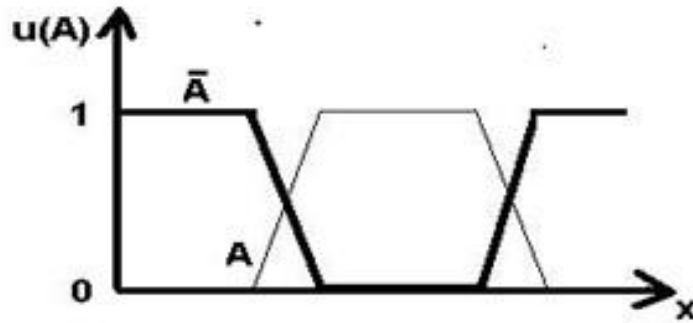


Figura 16: Operación Complemento (Castillo, 2015)

2.6.4 Modelo lingüístico

Un modelo lingüístico es capaz de recolectar conocimientos basados en reglas IF - THEN.

$$IF x= A_i THEN y = B_i \quad i= 1,2, 3...k \quad (16)$$

Donde:

x = variable lingüística de entrada

A_i = etiqueta lingüística antecedente

B_i = etiqueta lingüística consecuente

Las etiquetas lingüísticas representan términos cualitativos para describir una relación por medio de reglas lingüísticas.

2.6.5 Variable Lingüística

Como su nombre lo indica, los valores de una variable lingüística se representan en términos lingüísticos cuyo significado será determinado de acuerdo a los conjuntos difusos. Las variables lingüísticas capturan variables de incertidumbre, las cuales se ajustan más a la realidad (alto, bajo, etc..). Una variable está definida por:

$$L = (x, A, X, G, m) \quad (17)$$

Donde:

x : nombre de la variable

A: {A1,A2...An}: conjunto de etiquetas o valores lingüísticas

X: Universo de discurso de x

G: Regla sintáctica para generar etiquetas

m: Regla semántica que asigna un significado a cada etiqueta.

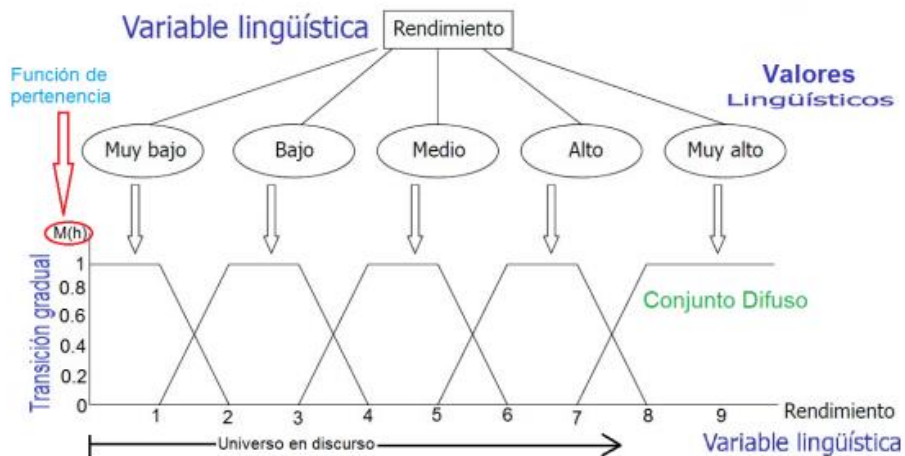


Figura 17: Descripción de variable lingüística (Carlos & Motoche, 2012)

2.6.6 Inferencia Difusa

Es el proceso el cual permite obtener un conjunto difuso de salida dados los valores de entrada y el uso de reglas dadas utilizando la lógica difusa.

La inferencia en un modelo lingüístico está basada en la regla de inferencia composicional de Zadeh, la cual es capaz de generar una conclusión mediante el conocimiento de un antecedente parcialmente conocido o de un hecho ocurre un hecho similar, pero no idéntico. (Lima, 2016)

2.7. Sistemas Difusos.

Los sistemas difusos están basados en su conocimiento, dicho conocimiento se genera a partir de la combinación e interpretación de las reglas IF-THEN, esto supone el uso de un dispositivo de inferencia difusa el cual posee una serie de bloques

necesarios para el tratamiento de datos de entrada y de salida los cuales se explican a continuación.(Meza, 2003)



Figura 18: Sistema difuso

2.7.1. Bloque Fusificador

Este bloque se encarga de asignar un valor de pertenencia a la/s variables de entradas del sistema, en referencia al/ los conjunto/s difusos definidos anteriormente convirtiendo así los valores reales de la entrada a valores difusos.



Figura 19: Bloque Fusificador

2.7.2 Máquina de inferencia

Tiene como finalidad asociar conjuntos difusos de entrada con los conjuntos difusos de salida para representar las reglas difusas que regirán la respuesta del sistema. Esto lo realiza un software (Matlab, Labview, etc) el cual puede procesar datos a mayor velocidad y con mayor eficiencia.

2.7.3 Bloque Defusificador

Este bloque, mediante el uso de los métodos de defusificación, devuelve un valor determinado de la variable de salida a partir del conjunto difuso obtenido como respuesta de la máquina de inferencia. Esta respuesta está dada en variable

lingüística, los métodos de defusificación convierten dicha variable a una variable escalar que es la que reconoce el proceso a controlar. Existen diversos métodos los cuales se muestran a continuación.

2.7.4. Centro de Gravedad

Consiste en calcular el centro de área del conjunto difuso obtenido por la máquina de inferencia mediante la siguiente fórmula:

$$y^* = \frac{\sum_{j=1}^F u(y_j)y_j}{\sum_{j=1}^F u(y_j)} \quad (18)$$

un valor único.

2.7.5. Promedio de los Centros

La salida se calcula mediante la siguiente ecuación, esta representa el promedio de los centros de las funciones de membresía que intervienen en la salida.

$$y^* = \frac{\sum_i b_i \mu_{hi}}{\sum_i \mu_{hi}} \quad (19)$$

Donde:

b_i = Centros de las funciones de membresía

μ_{hi} = La altura de la función de membresía con respecto al centro.

2.7.6. Método del Máximo

Este método escoge el valor máximo de la altura de la función característica del conjunto difuso que empleado para la salida. Si la altura contiene más de un punto se puede utilizar las siguientes alternativas:

Valor ínfimo de la altura:

$$y^{*1} = \inf\{y \in \text{altura } y^{\text{dif}}\} \quad (20)$$

Valor supremo de la altura:

$$y^*2 = \sup\{y \in \text{altura } y^{\text{dif}}\} \quad (21)$$

Valor promedio de la altura:

$$y^*3 = \text{prom}\{y \in \text{altura } y^{\text{dif}}\} \quad (22)$$

2.8. Arduino

Arduino es una herramienta de hardware y software libre cuyo circuito impreso integra un microcontrolador y un entorno de desarrollo (IDE), en donde se programa la placa. Esta herramienta se enfoca en facilitar el uso de la electrónica y programación de sistemas embebidos en proyectos multidisciplinarios.

El hardware contiene un microcontrolador, usualmente Atmel AVR, puertos digitales y analógicos de entrada/salida, los cuales se conectan a placas de expansión (shields), que amplían las características de funcionamiento de la placa. Asimismo, posee un puerto de conexión USB desde donde se puede alimentar la placa y establecer comunicación con el computador.

El entorno de desarrollo (IDE) está basado en Processing y lenguaje de programación basado en Wiring, así como en el cargador de arranque (bootloader) que es ejecutado en la placa. El microcontrolador de la placa se programa mediante un computador, usando una comunicación serial mediante un convertidor de niveles RS-232 a TTL serial.

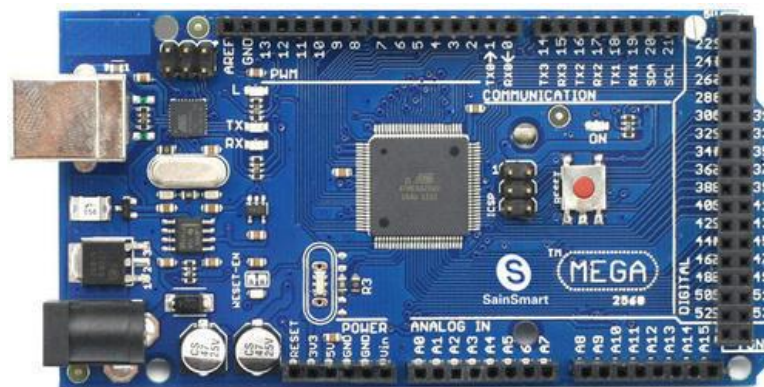


Figura 20: Controlador Arduino MEGA 2560 (Arduino,2015)

2.9. Módulo Xbee

Según la empresa Digi los módulos Xbee permiten implementar un medio inalámbrico en el cual servirá para conectar y comunicar diversos dispositivos suplantando el uso de un medio físico. Los Xbee usan el protocolo IEEE 802.15.4 (Zigbee), creando redes punto-punto y punto- multipunto para aplicaciones de alto tráfico de datos bajo coste y sincronización de comunicación (Morillo, 2017)

Las comunicaciones con el protocolo Zigbee se realizan a 2.4 Ghz a través de un solo canal. El alcance depende de la potencia de emisión del dispositivo y antena que se utilice, normalmente 100 m para módulos Xbee y 1.6km para Xbee Pro. Posee un bajo consumo de corriente menor a 50 mA en funcionamiento y 10 uA cuando no se está utilizando.



Figura 21:Modulo Xbee (Morillo, 2017)

2.10. Sensor

Un sensor es un sistema conjunto formado por un transductor, elemento que convierte una señal física a otra señal física, y un transmisor, elemento que recepta la variable física y la transforma en una señal de voltaje o corriente para que pueda ser leído por un sistema.

2.10.1. Sensor de Conductividad Atlas Scientific

El sensor Probe K 10 distribuido por la empresa Atlas Scientific, es un sensor que mide la conductividad eléctrica en una solución. Es comúnmente usado sistemas de acuicultura para monitorear la cantidad de nutrientes o impurezas del agua.

El funcionamiento es el siguiente: Consta de dos electrodos colocados en forma opuesta uno al otro, a los cuales se les aplica un voltaje AC haciendo que los cationes viajen hacia el electrodo negativamente cargado y los aniones donde el electrodo positivamente cargado. Mientras más electrolitos libres contenga el líquido más alto es la conductividad eléctrica. (Atlas Scientific, 2017)



Figura 22: Sensor Probe K1.0 (Atlas Scientific, 2017)

Material	Grafito
Rango	5-200000 uS/cm
Tiempo de Respuesta	90% en 1seg
Presión máxima	2447 KPA (500 PSI)
Profundidad máxima	343 m
Rango de Temperatura	1-110°C

Tabla 4: Especificaciones Generales Sensor Probe K1.0

2.10.1.1. Circuito para medición de conductividad EZO Atlas Scientific

Este circuito, junto con el sensor Probe K1.0 ofrece y asegura la correcta medición de la conductividad aparte de entregar otros valores de variables como salinidad, total de solidos disueltos y gravedad específica (solo para agua de mar). Tiene como

protocolos UART e I2C el primero viene por defecto programado en el circuito. Dicho protocolo UART trabaja a una Baudrate de 9600, semejante al controlador Arduino por lo cual se logra establecer una conexión de comunicación. (Atlas Scientific, 2017)

2.10.2. Sensor de Oxígeno disuelto Atlas Scientific

El sensor DO distribuido por la empresa Atlas Scientific, es un sensor que consiste de una sonda de oxígeno disuelto galvánica en una membrana de politetrafluoroetileno, un ánodo revestido en un electrolito y un cátodo. Las moléculas de oxígeno se desactivan a través de la membrana de las sondas a una velocidad constante (sin la membrana, la reacción sucede rápidamente). Una vez que las moléculas de oxígeno han cruzado la membrana, se reducen en el cátodo y se produce un pequeño voltaje. Si no hay moléculas de oxígeno presentes, la sonda emitirá 0 mV. A medida que aumenta el oxígeno, también lo hace la salida de mV de la sonda. (Atlas Scientific, 2017)

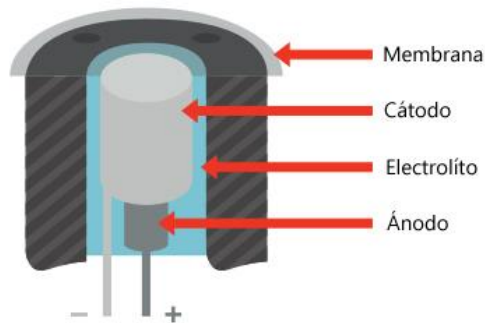


Figura 22: Sensor DO (Atlas Scientific, 2017)

Material	Galvano
Rango	0.01-100+ mg/L
Tiempo de Respuesta	100% en 1seg
Presión máxima	3447 KPA (500 PSI)
Profundidad máxima	343 m
Rango de Temperatura	1-50°C

Tabla 5: Especificaciones Generales Sensor DO

2.10.2.1. Circuito para medición de oxígeno disuelto EZO Atlas Scientific

Este circuito, junto con el sensor DO ofrece y asegura la correcta medición de la lectura del oxígeno disuelto. Tiene como protocolos UART e I2C. Dicho protocolo UART trabaja a una baudrate de 9600, semejante al controlador Arduino por lo cual se logra establecer una conexión de comunicación. (Atlas Scientific, 2017)

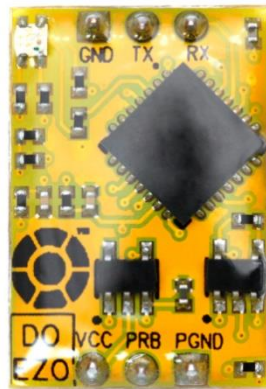


Figura 23: Circuito de medición DO (Atlas Scientific, 2017)

2.10.3 Sensor de temperatura waterproof ONE WIRE

El sensor de temperatura ONE WIRE ofrece una medición adecuada de temperaturas de líquidos en tanques. Su cableado de aislamiento, de 2 metros de largo, posee una conexión tipo 3 hilos y está protegido por recubrimiento de acero inoxidable. Opera en un rango de -55 a 125 °C. Este dispositivo hace uso de la tecnología ONE WIRE la cual solo necesita un conductor para poder realizar la comunicación.



Figura 23: Sensor de temperatura ONE WIRE DS18B20 (DFRobot, 2008)

Rango de Temperatura	-55° C – 125°C
Protección	Acero inoxidable
Longitud del cable	91 cm
Longitud del tubo protector	50 mm

Tabla 6: Especificaciones de sensor DS18B20 (DFRobot, 2018)

2.11. Matlab

Matlab (Matrix Laboratory) es un software matemático desarrollado por Cleve Moler en conjunto con la empresa Mathworks, proporciona herramientas para la resolución de problemas matemáticos de ingeniería con la ayuda de los denominados ToolBox, paquetes de funciones con soluciones específicas enfocadas a diseños de sistemas de control, identificación de sistemas, procesamiento de señales, redes neuronales etc. (León, 2014)

Matlab posee dos herramientas que son de gran ayuda para la interpretación de sistemas y visualización de datos mediante una interfaz, estos son Simulink (plataforma de simulación multidominio) y GUIDE (editor de interfaces de usuario). Una ventaja de este software es la capacidad de crecimiento, ya que los usuarios pueden desarrollar y compartir sus propias aplicaciones (scripts) adaptando y modificando el código para a ser aplicados en determinadas situaciones,



Figura 24: Software MATLAB

2.12. Labview

Labview es un entorno de desarrollo, creado por la empresa National Instruments, enfocado para el diseño de sistemas de automatización y control de procesos industriales y la adquisición, procesamiento y visualización de datos de los mismos. Su lenguaje de programación gráfico, más conocido como lenguaje G, ofrece muchas ventajas al minimizar el tiempo de programación simplificándolo graficas al uso de sus diversas librerías de funciones (Jolly, 2017).

Al contrario de la programación convencional a base de líneas de código, Labview posee sus funciones en forma de íconos, los cuales son bloques de datos y estos se unen mediante “cables” que representan los flujos de datos (Figura 25), brindando la posibilidad al usuario de poder realizar programas sencillos sin la necesidad de ser un experto en programación.

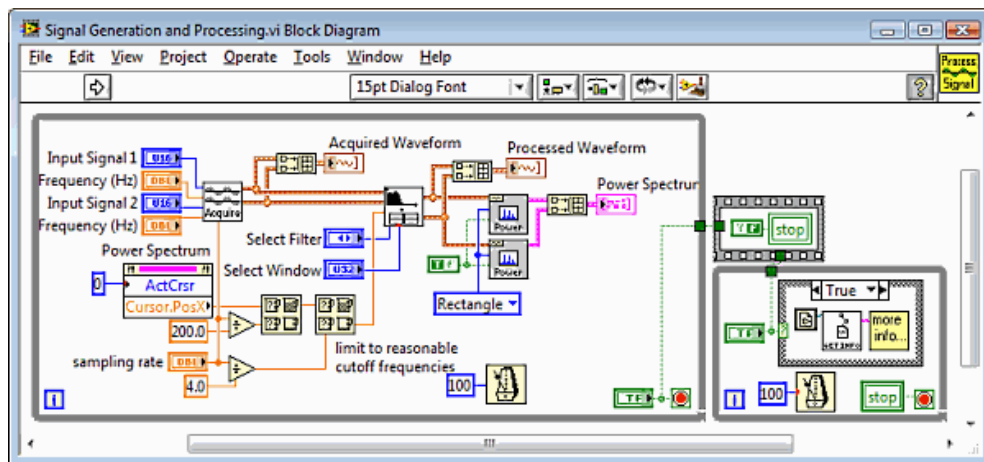


Figura 25: Ejemplo de Código fuente de Labview (Jolly, 2017)

3. DISEÑO

En el siguiente capítulo se detalla la elaboración de los diferentes diseños tanto eléctricos como electrónicos y de actuadores implementados para el correcto funcionamiento de la planta.

3.1 Diseño Electrónico

Consta de tres módulos enlazados entre sí, los cuales permitirán la recolección, tratamiento, envío y recepción de la información hacia el sistema SCADA,

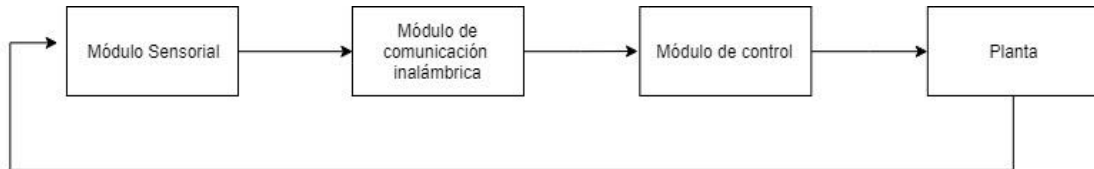


Figura 26: Diagrama de bloques de módulos electrónicos

A continuación, se describen los diagramas de bloques de las plantas de Temperatura, salinidad y oxígeno disuelto.

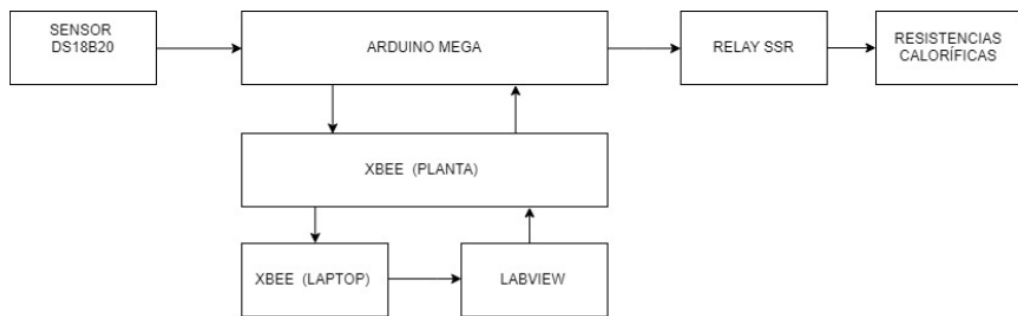


Figura 27: Diagrama de bloques de planta Temperatura

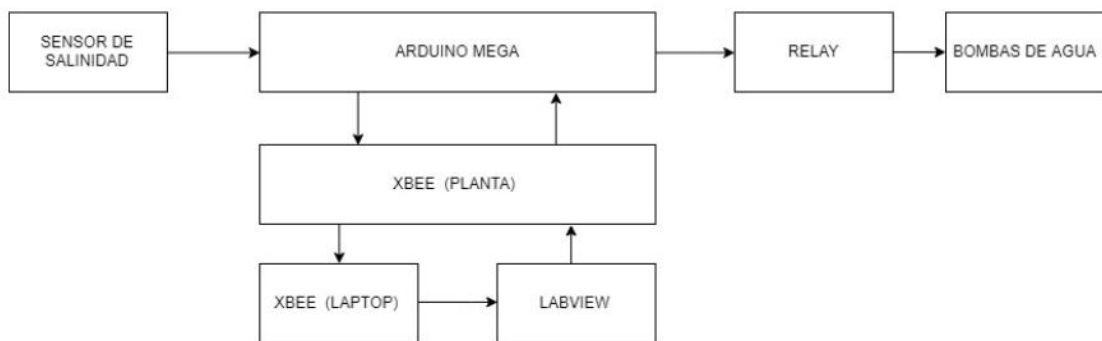


Figura 28: Diagrama de bloques de planta Salinidad

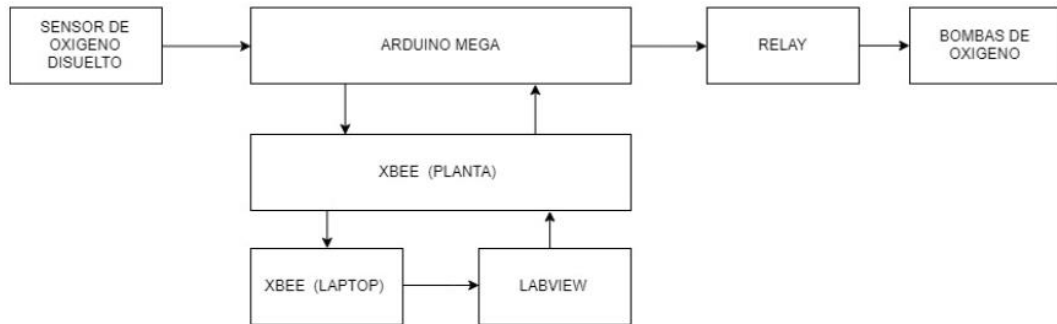


Figura 29: Diagrama de bloques de planta Oxígeno Disuelto

3.1.1 Módulo de sensores

Conformado por sensores de temperatura, salinidad y oxígeno disuelto. Se encargan de medir las variables físicas antes descritas y transformarlas en señales eléctricas, para ser leídas por el controlador Arduino.

3.1.1.1 Sensor de Temperatura

Consta del sensor de temperatura DS18B20 junto a su circuito terminal. Dicho circuito recibe el valor de temperatura medido por la probeta y lo transmite hacia el controlador Arduino por medio de una salida DATA de tipo Binario. El rango de operación del sensor va desde 55° C a 125°C.



Figura 30: Sensor de Temperatura instalado en la planta

3.1.1.2 Esquemático

El circuito terminal del módulo de temperatura está conformado por dos resistencias de pull up, para eliminar el ruido en la señal y delimitar un estado de entrada cuando el cable de datos no esté conectado.

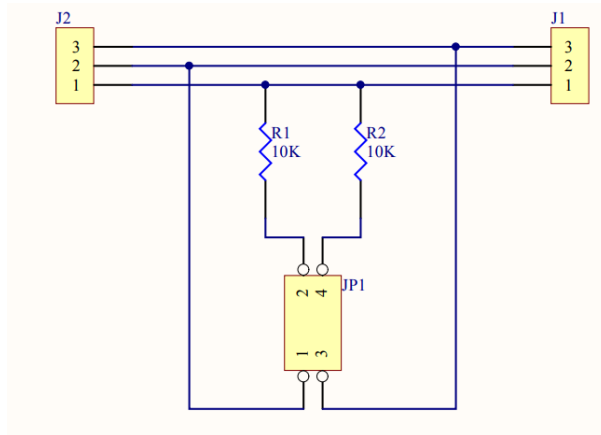


Figura 31: Circuito Esquemático adaptador sensor de temperatura (DFRobot, 2018)

3.1.1.3 Sensor de Salinidad Probe K1.0

Para el control de salinidad se utiliza el sensor Atlas Scientific junto a su circuito amplificador, el cual se conecta a los terminales BNC de la probeta del sensor. El circuito se alimenta a 5 VDC, y realiza la transmisión de valores por medio del protocolo UART al Arduino MEGA para la posterior etapa de control. El rango de operación de la probeta va desde 5 a 200000 uS/cm

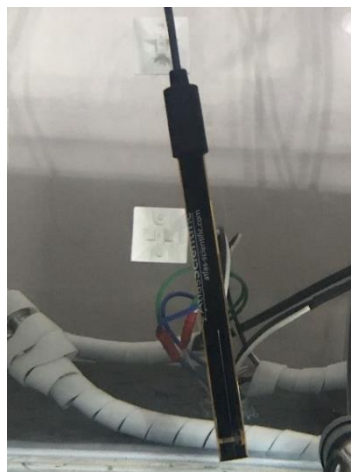


Figura 32: Sensor de salinidad instalado en la planta

3.1.1.4 Esquemático

Se representa el circuito amplificador. El sensor se comunica a través del conector BNC enviando los datos hacia el circuito (pones PRB Y PGND). El cual envía los valores hacia el controlador por medio de comunicación serial (pines RX-TX).

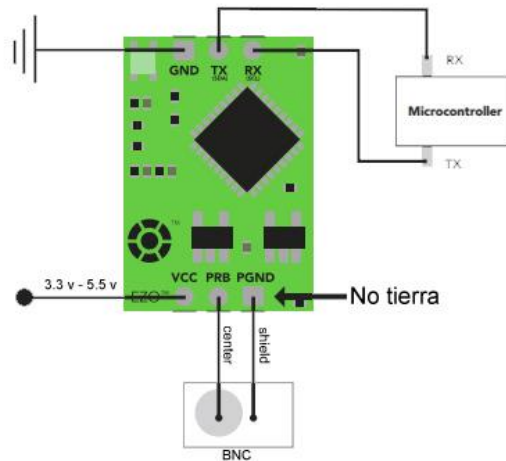


Figura 33: Diagrama de circuito de conexión de sensor de salinidad (Atlas Scientific, 2018)

3.1.1.4.1 Calibración de sensor de Salinidad

Para realizar la calibración del sensor de salinidad se utilizan dos soluciones salinas las cuales forman parte del KIT del sensor de salinidad C.O Atlas Scientific. Estas soluciones poseen dos valores para la calibración, uno para realizar la calibración a nivel bajo con un valor de 12,88 uS y 80,000 uS para la calibración a nivel alto (figura 34).



Figura 34: Soluciones para Sensor de Salinidad (Atlas Scientific, 2018)

La calibración que se realiza es a nivel bajo realizando los siguientes pasos:

- Se ingresa la probeta en un recipiente con Agua Potable (salinidad 0) para eliminar impurezas o residuos de sales que contenga la probeta
- Desde el monitor serie del controlador Arduino, una vez establecida la comunicación con el sensor C O Atlas Scientific se digita el comando "Cal,0" para calibrar el valor del sensor en 0
- Se ingresa la probeta en la solución de valor bajo, agitándola para asegurar que no queden burbujas atrapadas en la superficie.
- El valor de lectura del sensor variará entre el 1 al 40% de su valor real durante un tiempo, luego de esto se estabiliza en el valor de la solución
- Se ingresa la probeta en la solución de valor alto (12880 uS), agitándola para asegurar que no queden burbujas atrapadas en la superficie como se puede observar en la figura 35.
- El valor de lectura del sensor variará entre el 1 al 40% de su valor real durante un tiempo, luego de esto se estabiliza en el valor de la solución

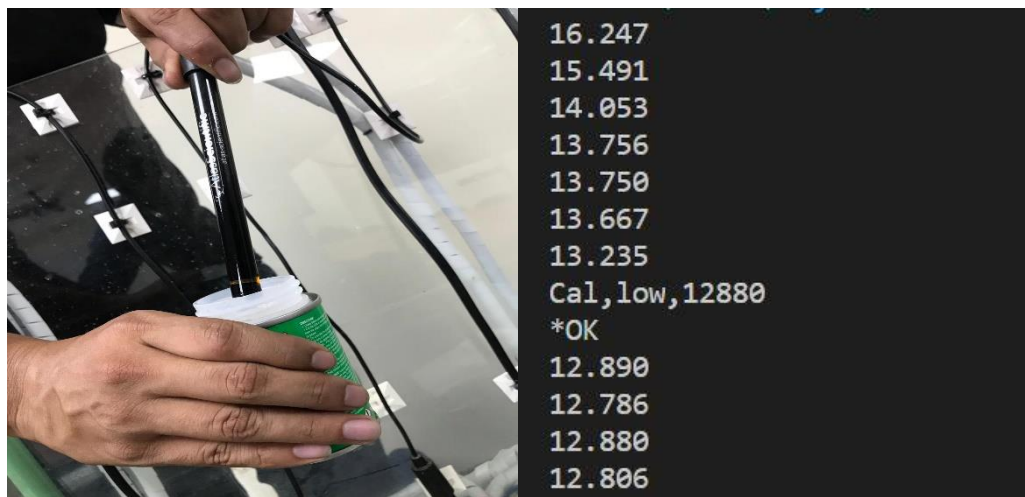


Figura 35: Calibración de sensor de salinidad

3.1.1.5 Sensor de Oxígeno disuelto DO ATLAS Scientific

Para el control de oxígeno disuelto se utiliza el sensor DO de Atlas Scientific (figura 36) acompañado del circuito amplificador que está contenido en el KIT, el cual se conecta a los terminales BNC de la probeta del sensor. Se alimenta a 5 VDC, y realiza la transmisión de valores por medio de UART al Arduino MEGA. El rango de operación de la probeta va desde 0.01 a 100+ mg/L



Figura 36: Sensor de oxígeno instalado en la planta

3.1.1.6 Esquemático

Se representa el circuito amplificador. El sensor se comunica a través del conector BNC enviando los datos hacia el circuito (pones PRB Y PGND). El cual envía los valores hacia el controlador por medio de comunicación serial (pines RX-TX).

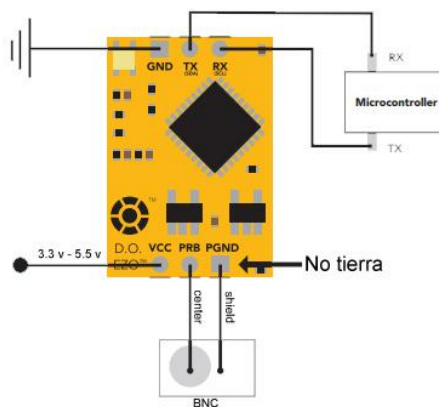


Figura 37: Diagrama de circuito de conexión de sensor de oxígeno (Atlas Scientific, 2018)

3.1.1.61 Calibración de sensor de Oxígeno disuelto

Para realizar la calibración del sensor de oxígeno se utiliza una solución electrolítica (figura 38), la cual forma parte del KIT del sensor de oxígeno D.O Atlas Scientific. Esta solución posee un valor de lectura de 0 mg/L lo cual permite calibrar correctamente el sensor.



Figura 38: Solución Electrolítica de oxígeno Disuelto (Atlas Scientific, 2018)

- Se ingresa la probeta dentro de la solución y se revuelve para remover el aire atrapado dentro de esta, lo que puede causar lecturas muy altas.
- Se deja reposar la probeta dentro de la solución hasta que las medidas se estabilicen, esto puede llegar a tomar de 10seg a 1:30 min.
- Desde el monitor serie del controlador Arduino, una vez establecida la comunicación con el sensor se digita el comando "Cal,0" como se puede observar en la figura X

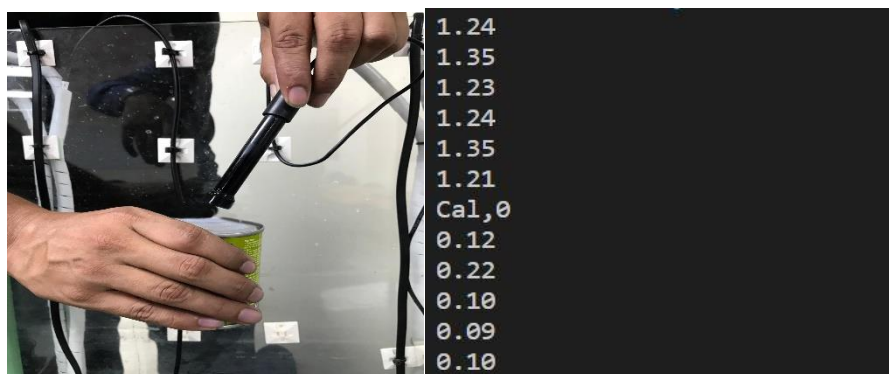


Figura 39 Calibración de sensor de oxígeno

3.1.2 Módulo de Comunicación Inalámbrica

Conformado por dos dispositivos de comunicación inalámbrica (XBEE). Este módulo realiza el envío de los datos brindados por el módulo de sensores al controlador (figura 41) y la recepción de comandos dictados por la computadora central (Sistema SCADA) hacia el controlador. La comunicación es de tipo FULL DUPLEX, es decir, ambos dispositivos XBEE pueden recibir y enviar información al mismo tiempo y se realiza por medio del protocolo de red IEEE 802.15.4 siendo una red Punto-Punto.

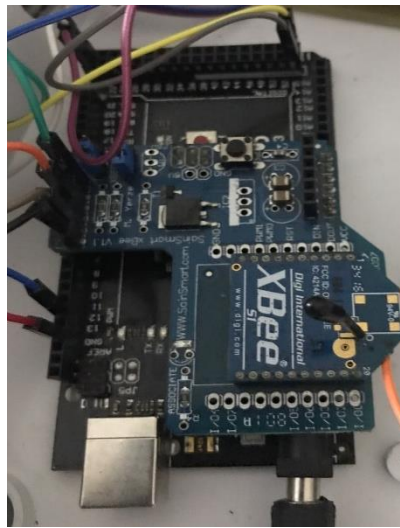


Figura 40: Módulo de comunicación XBEE



Figura 41: Instalación de Módulo XBEE en la planta

3.1.3 Módulo de Control

Es el encargado del procesamiento de los comandos de activación/desactivación de los actuadores de la planta (bombas, electroválvulas, resistencias, etc.) enviados desde el SCADA, por medio del protocolo serial, hacia el XBEE conectado al ARDUINO MEGA en la planta. Este módulo es el enlace entre la electrónica de control y los circuitos de fuerza de los actuadores.

3.1.3.1 Módulo de Relés

Para el control de los actuadores del sistema (electroválvulas, bombas y aireador) se hace uso de un módulo de relés, la placa está compuesta de 8 relés los cuales soportan hasta una corriente de 6 amperios y voltaje de 110VAC en sus contactos. Su accionamiento es activado con un voltaje de 5VDC, señal enviada por el controlador principal, pueden ser activados tanto individual como grupalmente. Su PCB está diseñada para aislar la parte de fuerza (Contactos NO y NC), con la de control (señal enviada del Arduino) y proteger esta última de sobrecargas.

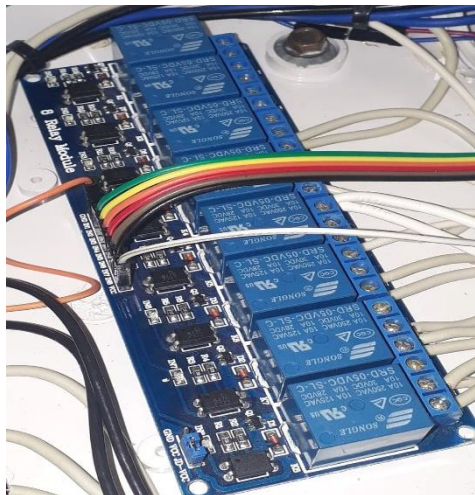


Figura 42: Módulo de Relés instalados en la planta

3.1.3.2 Relevador de Estado Sólido

Para el control de las resistencias de calentamiento se hace uso de un relevador de estado sólido, el cual posee una velocidad de conmutación mucho más rápida que un relé con accionamiento mecánico, permitiendo variar proporcionalmente la frecuencia de operación de las resistencias. El voltaje permitido aplicado a la bobina de accionamiento va desde 3 VDC hasta 32 VDC, para la planta, se utiliza una señal de 5VDC enviada desde el controlador principal. Su salida soporta una corriente máxima de hasta 40 Amperios desde 24VAC a 480 VAC, permitiendo conectar el grupo de resistencias a la misma salida sin presentar problemas de sobre corriente.



Figura 43: Relevador de Estado Sólido instalado en la planta

3.2 Diseño Eléctrico

Para la correcta alimentación de toda la planta, se diseñó un panel de fuerza, el cual posee el breaker principal y los breakers secundarios para cada actuador. Adicional se encuentran los circuitos de relés de fuerza para los actuadores que demandan un mayor consumo de corriente (Bombas y Resistencias).

La capacidad del breaker principal supera la de los breakers secundarios para asegurar la protección contra cortocircuitos de toda la planta. Uno de los breakers

secundarios protege a las dos bombas que consumen individualmente 3,36 Amperios. El otro breaker se encarga de proteger a las cuatro resistencias, estas consumen 1,36 Amperios individualmente.

Los relés RH2B-U, dos en total, soportan un amperaje de hasta 10 Amperios por contacto (posee dos contactos por relé). Cada bomba posee una conexión única con su relé.

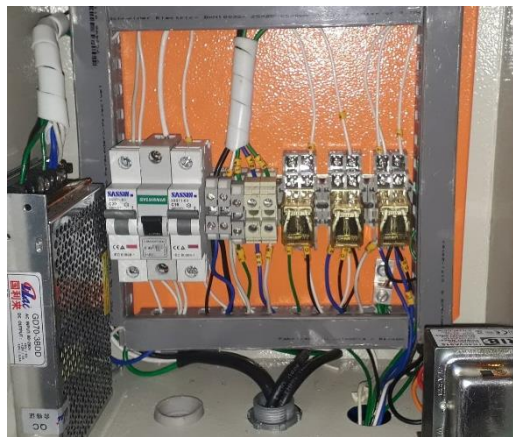


Figura 44: Tablero Eléctrico de la planta

3.3 Actuadores utilizados

Los actuadores utilizados en la planta fueron elegidos de acuerdo con las dimensiones de la piscina de la planta, esta será llenada hasta alcanzar los 150 litros como máximo volumen. Los cuales se detallan a continuación.

3.3.1 Bombas

Para el proceso de salinidad de la piscina se tiene un sistema de bombeo tanto de agua salada como de agua dulce, dependiendo del nivel de salinidad que se tenga el controlador enviará a activar una de las dos bombas con su respectiva electroválvula.

Poseen un caudal de 35 l/min y son alimentadas a 110VAC con una potencia de 0,5 HP.



Figura 45: Bombas de agua utilizada en la planta

3.3.2 Electroválvulas

Trabajan en conjunto con las bombas para suministrar agua dulce como salada. Permiten el flujo de agua desde los reservorios hasta la piscina, si las electroválvulas no están accionadas, las bombas no encenderán. Su señal de control posee voltaje de 12 VDC.



Figura 46: Electroválvula instalada en el sistema de tuberías de la planta

3.3.3 Resistencias

Para el control de temperatura, se realiza el cálculo respectivo de la potencia requerida para el cambio requerido de temperatura que se requiere. El setpoint

máximo en la planta serán 29.5°C. A continuación, se presentan los rangos mínimo y máximo de temperatura para calcular el delta máximo de temperatura.

$$T_i = 24 \text{ C} \quad (23)$$

$$T_{\text{max}} = 29.5\text{C} \quad (24)$$

$$\Delta \text{ MaxTemp} = T_i - T_{\text{max}} \quad (25)$$

$$\Delta \text{ MaxTemp} = 5.5 \quad (26)$$

Donde:

T_i : Temperatura inicial del sistema

T_{max} : Temperatura máxima de trabajo del sistema.

$\Delta \text{MaxTemp}$: variación máxima de temperatura

El fluido utilizado en la planta (agua), utiliza un total de 1000 calorías para aumentar su temperatura 1°C a razón de 1Lt, calculando las calorías necesarias para aumentar el delta máximo se tiene lo siguiente:

$$5.5 * 1000 \text{ cal/Lt } ^\circ\text{C} = 5500 \text{ cal/lit} \quad (27)$$

Como la capacidad de la piscina es de 150Lts se tiene:

$$5500 * 150 \text{ lit} = 825000 \text{ cal} = 825 \text{ kcal} \quad (28)$$

Una vez obtenida las Kcal necesarias se realiza la conversión de Kcal a Kwh

$$825 \text{ Kcal} * \frac{1\text{Kwh}}{860\text{Kcal}} = 0,96 \text{ Kwh} \quad (29)$$

Por lo tanto, se necesitarán 960 WATTS para elevar la temperatura de 24°C a 29.5°C.

Para cubrir la carga necesaria y reducir el tiempo de calentamiento se utilizaron cuatro resistencias de valores comerciales. Alimentadas con un voltaje de 120VAC con una potencia de 300W individualmente, varían su frecuencia de activación

proporcionalmente para elevar la temperatura en la planta, hasta llegar al setpoint deseado.



Figura 47: Resistencias instaladas en la planta

3.3.4 Bombas de Oxígeno

Para el suministro de oxígeno, se utiliza dos aireadores de oxígeno que cubren la demanda total de la planta. Se alimentan con un voltaje de 120VAC y proporcionan un caudal de 1700 L/h brindando una recirculación de agua dentro de la planta.



Figura 48: Bombas de oxígeno utilizada en la planta

3.4 Diseño de Estructura

La estructura de la planta está hecha a base de hierro negro, para sostener el peso de la piscina, los reservorios y el sistema de tuberías. Con una medida de 1,50 metros de largo por 0,80 metros de ancho y 0,85 metros de alto, posee dos niveles.

En el nivel inferior van colocados los reservorios y las bombas con sus tuberías, que son conducidas hacia el nivel superior, donde va ubicada la piscina con todos los sensores y actuadores conectados hacia el tablero de control.



Figura 49: Estructura de la planta con todos sus componentes

3.5 Desarrollo de modelos matemáticos

A continuación, se detalla la serie de pasos que se realizan para la identificación de los modelos matemáticos para los procesos de transferencia de calor (temperatura), salinidad y nivel de oxígeno disuelto.

3.5.1 Generación de señales PRBS

Para la identificación de los modelos matemáticos se hace uso de señales pseudo aleatorias binarias (PRBS), generándolas mediante un script en el software Matlab (figura 50).

```

PRBS.m x +
4   %% SCRIPT PRBS
5   close all;
6   clear all;
7   clc;
8   fclose('all');
9   delete(instrfindall);
10  filename = 'datos_pbrs_temperatura_ULTIMATE_continuacion.xlsx';
11  delete(filename);
12  %%borrar archivo excel
13
14  %% creacion de objeto serial
15  s = serial('COM16');
16  s.Terminator = 'CR/LF';
17  s.Baudrate=9600;
18  plotTitle='PRBS Temperatura';
19  xlabel='Tiempo (s)';
20  ylabel='Centigrados (C)';
21  y2Label='Activacion';
22  plotGrid = 'on';
23  delay = 0.01;
24
25  %Definir variables de funcion

```

Figura 50: Script para señal PRBS desarrollado en Matlab

Las señales PRBS generadas para las variables de Temperatura, Salinidad y Oxígeno disuelto se presentan en las figuras 51,52 y 53.

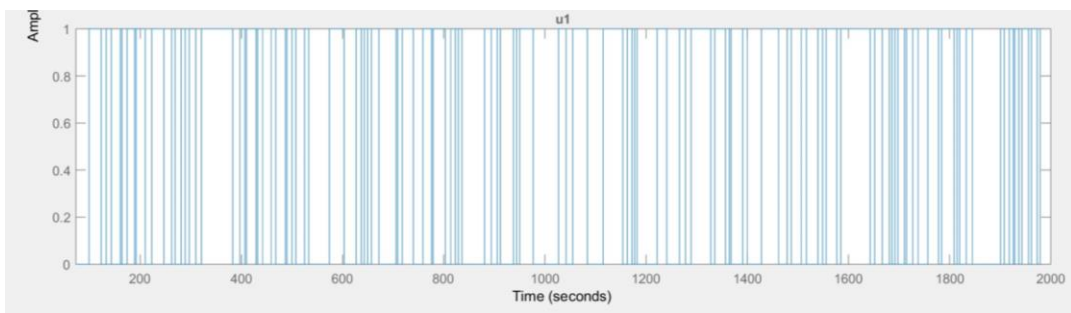


Figura 52: Señal PRBS generada para planta de Temperatura

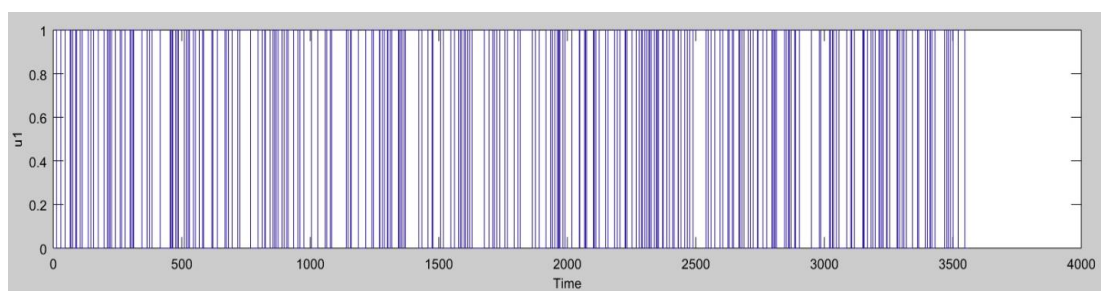


Figura 51: Señal PRBS generada para planta de Salinidad

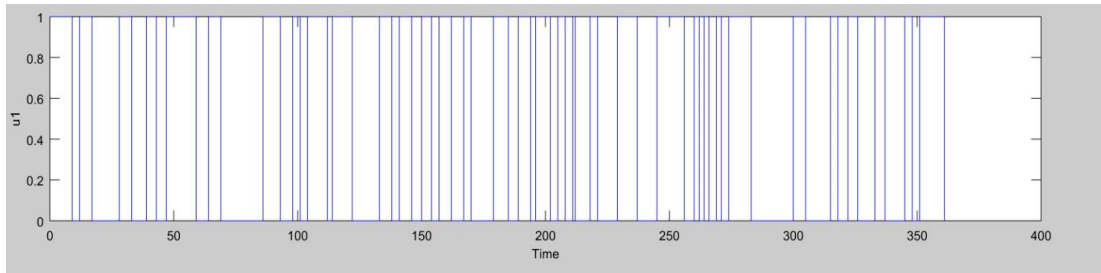


Figura 53: Señal PRBS generada para planta de Salinidad

Su equivalente en tabla de datos se muestra en la figura 54

C	D
Tiempo	ValorA...
1.6808	1
2.6439	1
3.6894	1
4.7041	1
5.7532	1
7.7889	0
8.8126	0
9.8533	0
10.8959	0
11.9170	0
12.9866	0
15.0204	0
16.0238	0
17.0702	0
18.0872	0
19.1249	0
20.1507	0

Figura 54: Tabla de valores de datos de señal PRBS

Dichas señales son enviadas a través de la comunicación inalámbrica hacia la planta siendo estas las entradas para Temperatura, salinidad y oxígeno.

3.5.2 Adquisición de Datos

Una vez enviadas las señales PRBS a la planta, los sensores instalados devolverán valores de lectura de cada variable, las cuales registran el comportamiento de los sistemas de acuerdo a cada señal aplicada. El tiempo de muestreo está fijado a 1

segundo y se almacenan como vectores en Matlab, para luego ser generado un archivo Excel (Figura 55)

B	C	D
Tempe...	Tiempo	ValorA...
Number ▾	Number ▾	Number ▾
Tempe...	Tiempo	Valor A...
26.6200	1.6808	1
26.6900	2.6439	1
26.6900	3.6894	1
26.6900	4.7041	1
26.6900	5.7532	1
26.6900	7.7889	0
26.6900	8.8126	0
26.6900	9.8533	0
26.6900	10.8959	0
26.6900	11.9170	0
26.6900	12.9866	0
26.6900	15.0204	0
26.6900	16.0238	0
26.6900	17.0702	0
26.6900	18.0872	0
26.6900	19.1249	0
26.7500	20.1507	0
26.7500	22.1947	0
26.6900	23.2341	0
26.7500	24.2569	0
26.6900	25.2955	0
26.7500	26.3203	0

Figura 55: Lectura de sensor de temperatura

3.5.3 Identificación de plantas de temperatura, salinidad y oxígeno disuelto.

Con los datos de entrada y salidas recolectados se procede a realizar la identificación de los modelos matemáticos de cada uno de los sistemas utilizando la herramienta IDENT, propia del software MATLAB. En la cual, ingresando los datos de entrada-salida, calcula y aproxima las funciones de transferencias, evaluando cual es la mejor aproximación en base a os porcentajes de validación.

La identificación del modelo de la planta de temperatura se realiza bajo las siguientes condiciones:

Salinidad: 30 PPT

Oxígeno disuelto: 7 mg/L

Volumen de agua: 100 L.

Obteniendo la siguiente función con 95% de estimación.

$$G_Temp = \frac{\text{From input "u1" to output "y1":}}{2.193e-05 s + 7.844e-07} \quad (30)$$
$$\frac{\text{-----}}{s^3 + 0.02801 s^2 + 0.0001681 s + 5.06e-09}$$

La identificación del modelo de la planta de salinidad se realiza bajo las siguientes condiciones:

Oxígeno Disuelto: 7 mg/L

Temperatura: 24°C

Volumen de agua inicial: 15 L.

Obteniendo la siguiente función con 91% de estimación.

$$G_Sal = \frac{\text{From input "u1" to output "y1":}}{-0.1218 s^2 + 0.04436 s + 0.0001015} \quad (31)$$
$$\frac{\text{-----}}{s^3 + 0.1246 s^2 + 0.0008079 s + 1.129e-07}$$

La identificación del modelo de la planta de oxígeno disuelto se realiza bajo las siguientes condiciones:

Salinidad: 30 PPT

Temperatura: 24°C

Volumen de agua: 100 L.

En base a las pruebas realizadas con la herramienta Ident, probando cada clase de controlador para aproximar la planta lo más acertada posible. No se llegó a una

estimación con un porcentaje correcto (por encima del 90%) debido a la naturaleza de la variable. Se puede observar en la figura 56 como la función de transferencia en el tiempo con el porcentaje más alto que se pudo obtener en la verificación (31,77%).

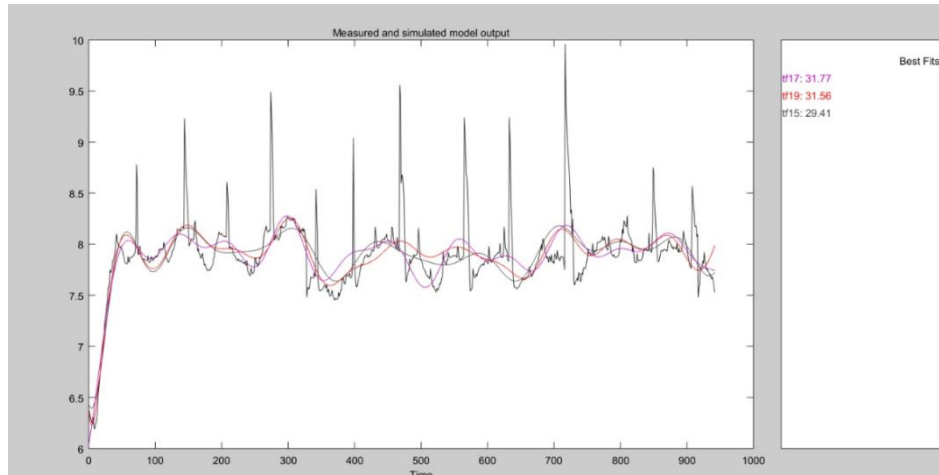


Figura 56: Funciones de transferencias estimadas para planta de Oxígeno

Dado que este porcentaje no es aceptable para una correcta simulación se hizo uso del modelado discreto para el sistema lineal, en específico, del modelo auto regresivo con variable exógena (ARX).

La identificación del modelo de la planta de oxígeno se realiza bajo las siguientes condiciones:

Temperatura: 24°C

Volumen de agua inicial: 100 L.

Obteniendo la siguiente función con 90.31% de estimación.

G_Ox_ARX =
 Discrete-time ARX model: $A(z)y(t) = B(z)u(t) + e(t)$

$$\begin{aligned}
 A(z) = & 1 - 0.9559 z^{-1} + 0.05205 z^{-2} + 0.06638 z^{-3} + 0.1507 z^{-4} - 0.1987 z^{-5} \\
 & + 0.0381 z^{-6} - 0.05805 z^{-7} + 0.06734 z^{-8} - 0.04944 z^{-9} + 0.0281 z^{-10} \\
 & - 0.0742 z^{-11} + 0.06009 z^{-12} - 0.03117 z^{-13} + 0.00637 z^{-14} + 0.02068 z^{-15} \\
 & - 0.03745 z^{-16} - 0.000209 z^{-17} - 0.01457 z^{-18} + 0.04711 z^{-19} - 0.02114 z^{-20} \\
 & - 0.03209 z^{-21} + 0.0172 z^{-22} + 0.01063 z^{-23} - 0.05454 z^{-24} + 0.06846 z^{-25} \\
 & - 0.03199 z^{-26} + 0.0236 z^{-27} - 0.0524 z^{-28} + 0.04594 z^{-29} - 0.03889 z^{-30} \\
 & + 0.006 z^{-31} + 0.01271 z^{-32} - 0.02963 z^{-33} + 0.03363 z^{-34} - 0.001774 z^{-35} \\
 & - 0.02178 z^{-36} - 0.09933 z^{-37} + 0.1155 z^{-38} - 0.02602 z^{-39} + 0.04405 z^{-40} \\
 & - 0.1136 z^{-41} + 0.06968 z^{-42} + 0.01435 z^{-43} - 0.01077 z^{-44} - 0.06466 z^{-45} \\
 & + 0.04417 z^{-46} - 0.01114 z^{-47} + 0.04275 z^{-48} - 0.05676 z^{-49} + 0.03041 z^{-50} \\
 & - 0.03154 z^{-51} + 0.01898 z^{-52} - 0.01102 z^{-53} + 0.02618 z^{-54} + 0.01541 z^{-55} \\
 & - 0.03449 z^{-56} + 0.04827 z^{-57} - 0.05987 z^{-58} + 0.08305 z^{-59} - 0.05607 z^{-60} \\
 & - 0.1609 z^{-61} + 0.138 z^{-62} + 0.008563 z^{-63} - 0.09807 z^{-64} + 0.01668 z^{-65} \\
 & + 0.03226 z^{-66} + 0.003126 z^{-67} - 0.02333 z^{-68} - 0.01103 z^{-69} + 0.0425 z^{-70} \\
 & + 0.008017 z^{-71} + 0.01079 z^{-72} - 0.04194 z^{-73} + 0.05495 z^{-74} + 0.008839 z^{-75} \\
 & + 0.002599 z^{-76} - 0.02841 z^{-77} + 0.01691 z^{-78} - 0.01325 z^{-79} + 0.00531 z^{-80}
 \end{aligned}
 \tag{32}$$

```

B(z) = -0.006008 z^-1 - 0.002407 z^-2 + 0.002708 z^-3 + 0.005105 z^-4 - 0.005595 z^-5
- 0.005075 z^-6 - 0.006516 z^-7 + 0.02243 z^-8 - 0.009035 z^-9 - 0.00755 z^-10
+ 0.023 z^-11 + 0.009051 z^-12 + 0.005068 z^-13 + 0.00176 z^-14 - 0.01783 z^-15
+ 0.02794 z^-16 + 0.001916 z^-17 + 0.0003421 z^-18 + 0.03108 z^-19 - 0.02894 z^-20
- 0.008857 z^-21 + 0.008964 z^-22 - 0.002256 z^-23 - 0.002849 z^-24 + 0.01109 z^-25
- 0.00143 z^-26 + 0.004287 z^-27 - 0.01522 z^-28 + 0.02016 z^-29 - 0.03982 z^-30
+ 0.04825 z^-31 - 0.01022 z^-32 + 0.002916 z^-33 - 0.003385 z^-34 - 0.00444 z^-35
+ 0.002729 z^-36 - 0.001469 z^-37 + 0.001366 z^-38 + 0.003064 z^-39 - 0.002478 z^-40
+ 0.0196 z^-41 - 0.03041 z^-42 + 0.0001418 z^-43 + 0.008954 z^-44 - 0.004421 z^-45
- 0.005455 z^-46 - 0.001325 z^-47 + 0.03083 z^-48 - 0.03061 z^-49 - 0.004427 z^-50
+ 0.002363 z^-51 + 0.001619 z^-52 + 0.005937 z^-53 + 0.008482 z^-54 - 0.03703 z^-55
+ 0.02298 z^-56 + 0.01997 z^-57 + 0.0001497 z^-58 - 0.01567 z^-59 + 0.003404 z^-60
- 0.01902 z^-61 + 0.01144 z^-62 + 0.00915 z^-63 - 0.01436 z^-64 + 0.002856 z^-65
- 0.005247 z^-66 + 0.01612 z^-67 + 0.005929 z^-68 - 0.01307 z^-69 - 0.008252 z^-70
+ 0.00589 z^-71 - 0.002056 z^-72 - 0.004809 z^-73 - 0.003569 z^-74 + 0.0006602 z^-75
- 0.009255 z^-76 + 0.007358 z^-77 + 0.003099 z^-78 + 0.01908 z^-79 - 0.03145 z^-80

```

Sample time: 1 seconds

Parameterization:

Polynomial orders: na=80 nb=80 nk=1

Number of free coefficients: 160

Use "polydata", "getpvec", "getcov" for parameters and their uncertainties.

Status:

Estimated using ARX on time domain data "Oxigeno disuelto".

Fit to estimation data: 90.31% (prediction focus)

FPE: 0.006097, MSE: 0.003283

Con las funciones de transferencia obtenidas para cada sistema se procede a realizar el proceso de validación con el fin de asegurar la similitud del modelo matemático calculado con la planta real.

Se toma una porción de los datos recolectados anteriormente y mediante la herramienta IDENT realiza la comparación entregando un porcentaje de similitud, el cual debe estar por encima del 70%. Para la planta de temperatura se obtuvo un porcentaje del 95% (figura 57).

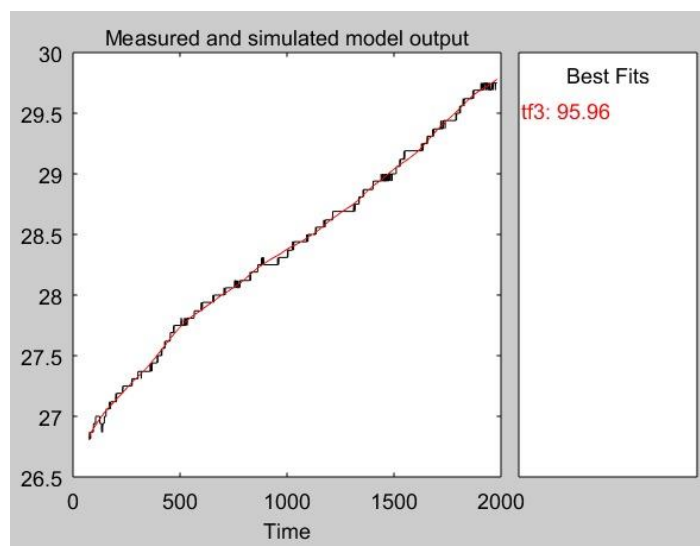


Figura 57: Validación de función de transferencia planta temperatura

En la planta salinidad se obtuvo un porcentaje de validación del 91,31% (figura 58).

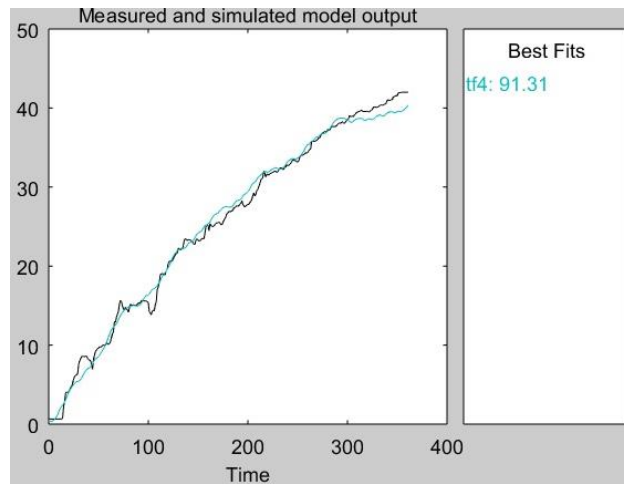


Figura 58: Validación de función de transferencia planta Salinidad

En la planta de exige se obtuvo un porcentaje de validación del 39.67% (figura 59).

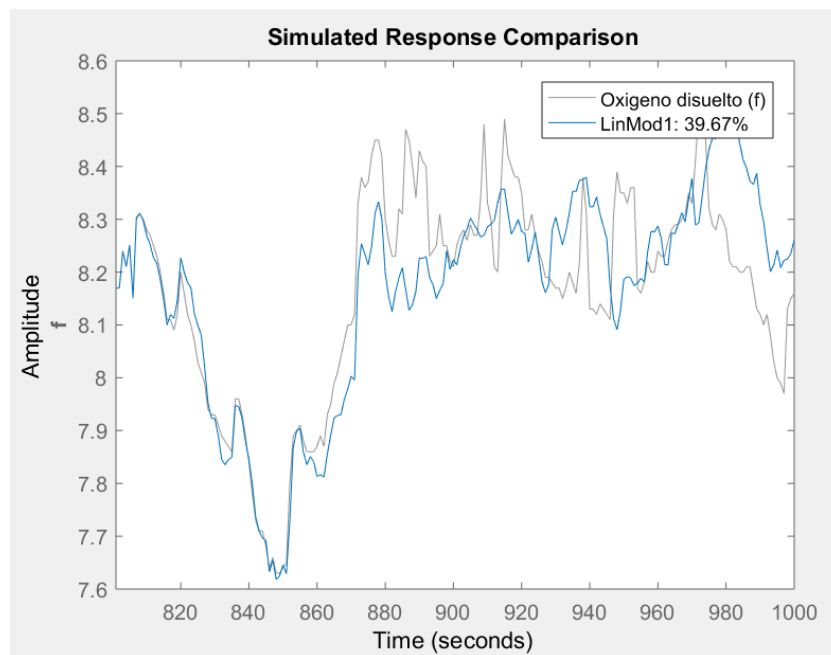


Figura 59: Validación de función de transferencia planta Oxígeno

3.5.4 Cálculo de parámetros para controladores PID.

Para calcular los valores de las ganancias proporcional, integrativa y derivativa de los controladores de las plantas se utilizan los métodos de Ziegler Nichols, se aplica una señal escalón en lazo abierto a cada una de ellas y se analiza la respuesta.

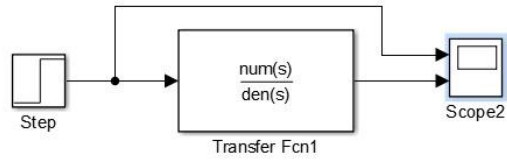


Figura 60: Planta de temperatura en lazo abierto con una entrada escalón

Como se muestra en la figura 61,62 y 63 las salidas de los sistemas presentan una forma de rampa al aplicar la señal escalón, haciendo imposible aplicar el primer método de Ziegler Nichols.

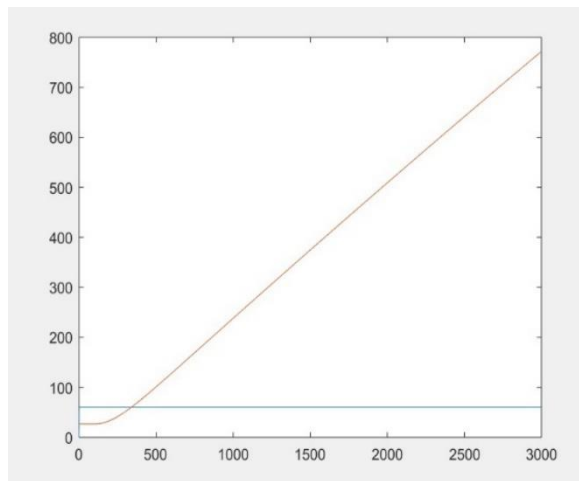


Figura 61: Respuesta de planta de temperatura con una entrada escalón en lazo abierto

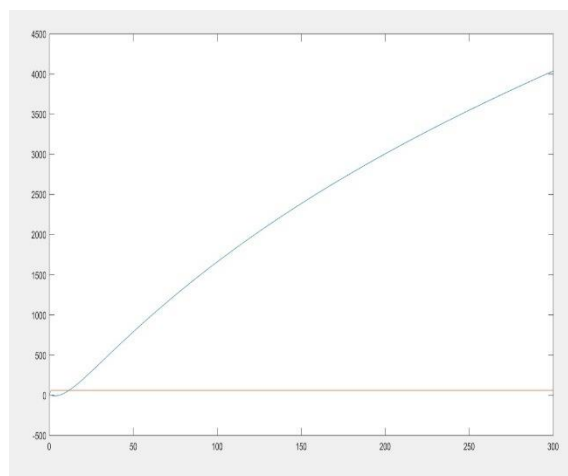


Figura 62: Respuesta de la planta de salinidad con una entrada escalón en lazo abierto

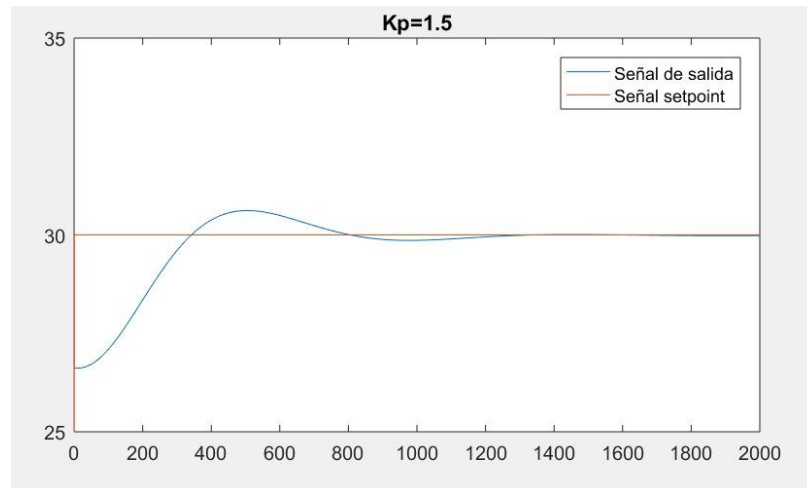


Figura 63: Respuesta de planta oxígeno disuelto ante una entrada escalón en lazo abierto

Por lo cual se aplica el segundo método de Ziegler Nichols, aplicando la señal escalón, está vez en lazo cerrado y variando la constante proporcional K_p hasta que el sistema se vuelva estable en el menor tiempo posible u oscile uniformemente.

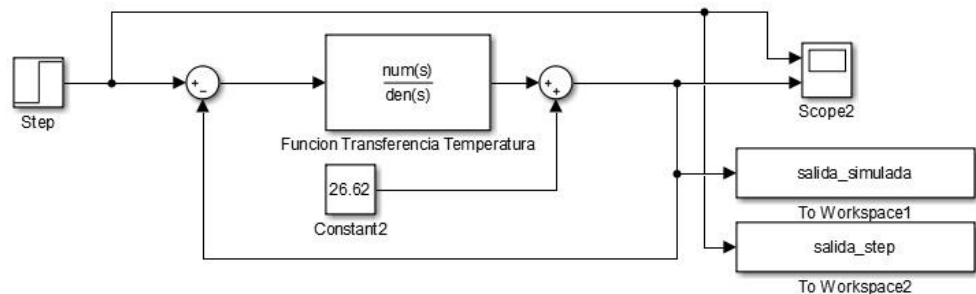


Figura 64: Planta de temperatura en lazo cerrado aplicada una entrada escalón

Como se puede apreciar en la figura 65 la constante K_p que estabiliza el sistema de temperatura posee un valor de 1,5.

Para la planta de salinidad el sistema se vuelve oscilatorio con $K_p = 1$, siendo esta la ganancia crítica (figura 66), se calcula el tiempo crítico a partir de la gráfica teniendo como resultado un $T_{cr}=30$ seg.

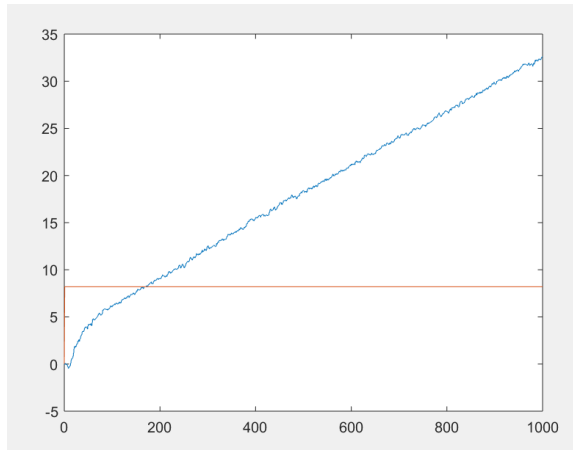


Figura 65: Respuesta de la planta de temperatura con $K_p=1,5$

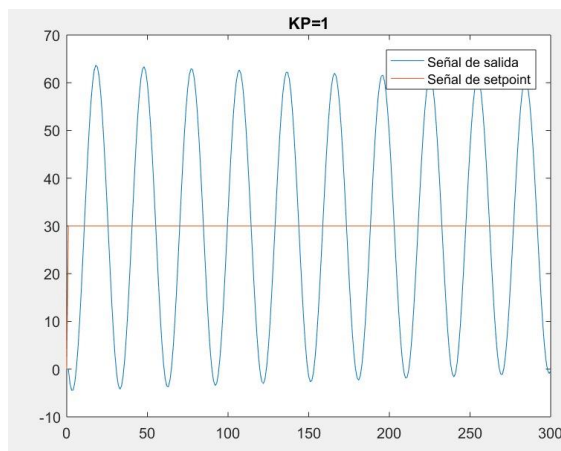


Figura 66: Respuesta de la planta de Salinidad con $K_p=1$

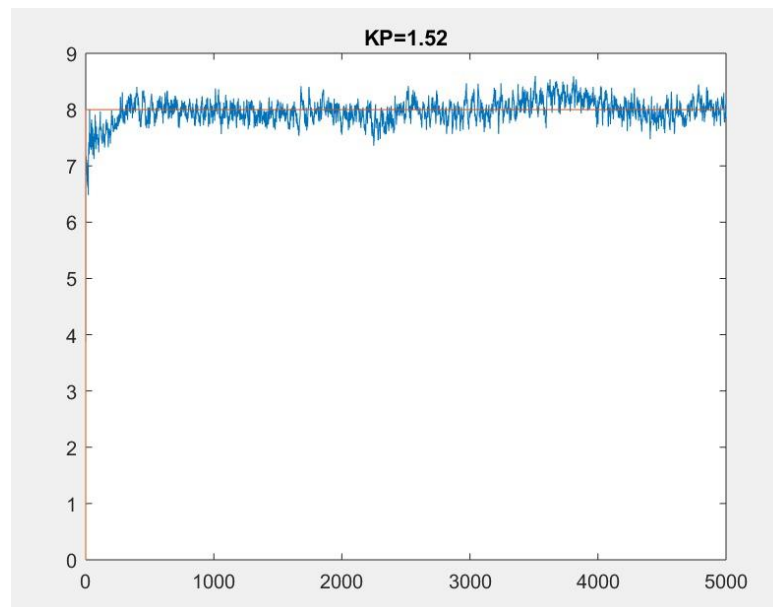


Figura 67: Respuesta de la planta de Oxigeno con $K_p=1.52$

3.5.5 Diseño de controladores clásicos

Para la planta de temperatura la ganancia de k_p calculada estabiliza el sistema, por lo cual solo se debe variar el valor de las otras ganancias K_i y K_d para mejorar la respuesta. Se utiliza la herramienta de simulink *PID Tunning*, obteniendo los siguientes resultados para los controladores P, PI, PD y PID (tabla 7).

Tipo	Tiempo de subida T_r (seg)	Tiempo de asentamiento T_s (seg)	Tiempo pico T_p (seg)	Sobrepaso máximo M_p (%)	Error en estado estable Ess (%)
P	521.8	808.6	1.121×10^3	1.38%	0.13%
PD	171.8	279.3	1.012×10^3	0%	0.16%
PI	204.3	1.063×10^3	471.4	22.45%	0.06%
PID	170.6	1.801×10^3	468.18	10.2%	0.03%

Tabla 7: Comparativa entre controladores clásicos. Planta Temperatura

En base a los resultados mostrados en la tabla 7, el controlador clásico que posee una mejor respuesta es el PD. Se realiza la gráfica comparativa para verificar esta respuesta.

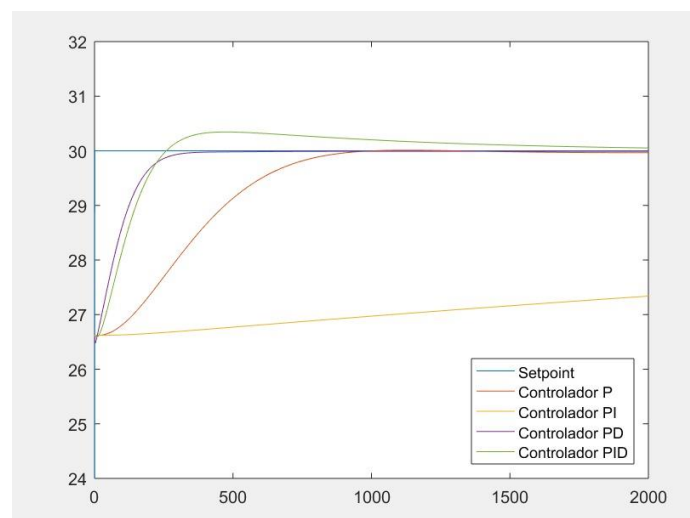


Figura 68: Respuesta de controladores Clásicos con setpoint de 30°C. Planta de Temperatura

Los valores de K_p y K_d calculados para el controlador Proporcional Derivativo, son los siguientes (Figura 69):

Controller Parameters	
	Tuned
P	2.3222
I	n/a
D	292.0551

Figura 69: Valores de K_p y K_d para controlador PD. Planta Temperatura

Para la planta de salinidad se realiza el mismo procedimiento anteriormente aplicado, adecuando los valores de las ganancias en cada controlador para obtener la mejor respuesta.

En la siguiente tabla se muestran los valores de los parámetros obtenidos de cada controlador.

Tipo	Tiempo de subida Tr (seg)	Tiempo de asentamiento to Ts (seg)	Tiempo pico Tp (seg)	Sobrepaso máximo Mp (%)	Error en estado estable Ess (%)
P	7.30	139.03	24.067	52.82%	1,86%
PD	5.751	44.54	17.31	22.99%	1.06%
PI	24.68	378.90	55.58	7.86%	1,33%
PID	5.062	103.47	51.41	3.78%	1.9%

Tabla 8: Comparativa entre controladores clásicos. Planta Salinidad

En base a los resultados mostrados en la tabla 8, el controlador clásico que posee una mejor respuesta es el PID. Se realiza la gráfica comparativa para verificar las respuestas.

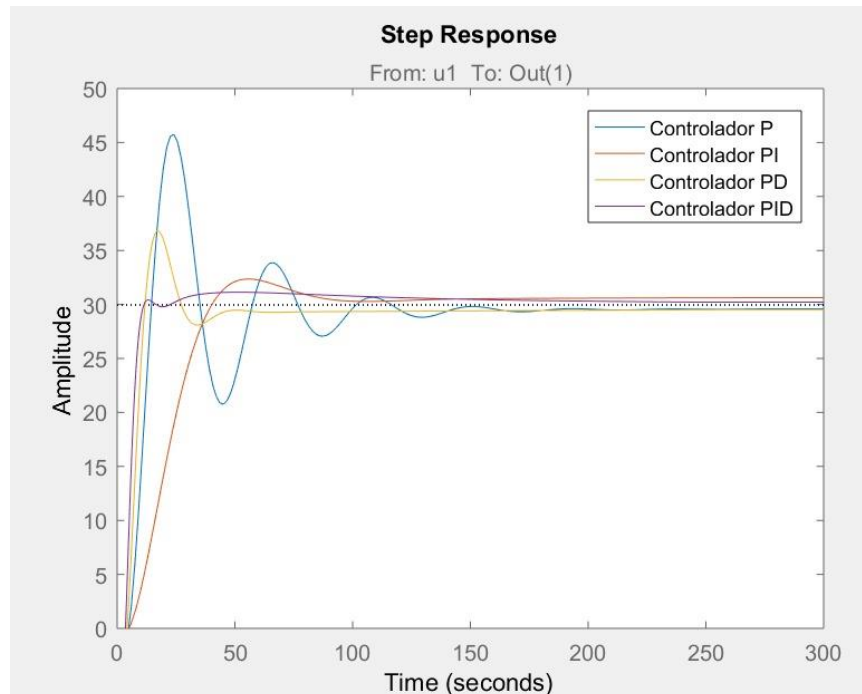


Figura 70: Respuesta de controladores Clásicos con setpoint de 30 PPT. Planta de Salinidad

Los valores de K_p , K_i y K_d calculados para el controlador PID, son los siguientes:

Controller Parameters	
	Tuned
P	0.35971
I	0.0045416
D	2.9855

Figura 71: Valores de K_p , K_i y K_d para controlador PID. Planta Salinidad

Los diferentes controladores aplicados a la planta de oxígeno devuelven los siguientes resultados mostrados en la tabla siguiente tabla (tabla 9):

Tipo	Tiempo de subida T_r (seg)	Tiempo de asentamiento o T_s (seg)	Tiempo pico T_p (seg)	Sobrepaso máximo M_p	Error en estado estable E_{ss}
P	461	895	2549	6%	1,86%
PD	15	595	1639	12%	1.06%
PI	369	2078	992	15%	1,33%
PID	80	1038	448	9%	1.9%

Tabla 9: Comparativa entre controladores clásicos. Planta Oxígeno

En base a los resultados mostrados en la tabla 9, el controlador clásico que posee una mejor respuesta es el PID. Se realiza la gráfica comparativa para verificar las respuestas

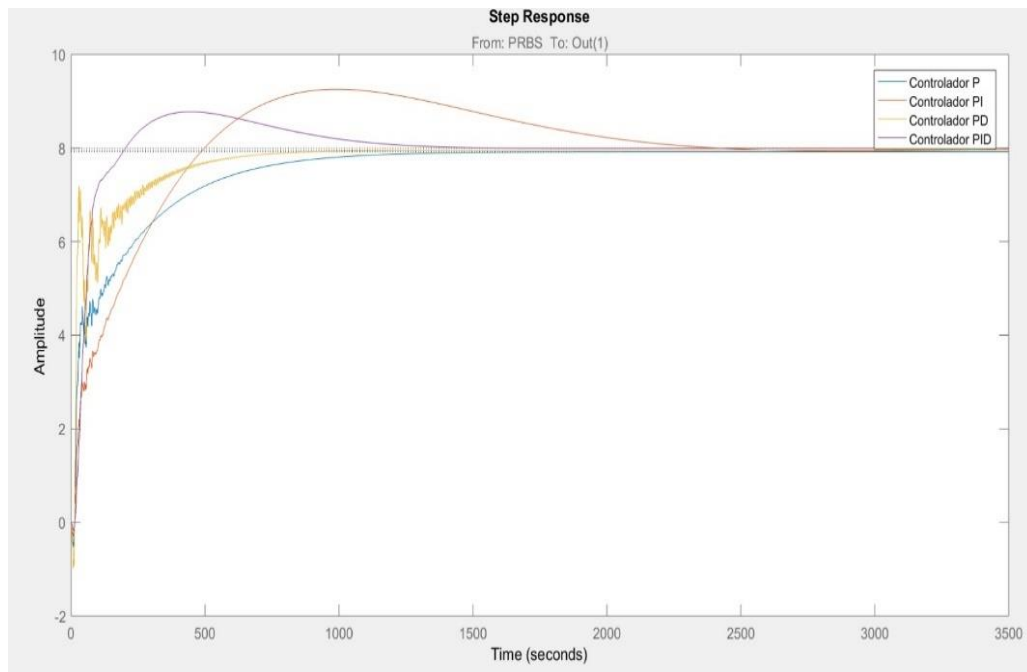


Figura 72: Respuesta de controladores Clásicos con setpoint de 8 PPM. Planta de Oxígeno

3.5.6 Diseño de controladores difusos

Para el diseño de los controladores difusos se utiliza el software Labview con su herramienta Fuzzy System Designer, creando funciones de membresía y reglas específicas para cada planta. En la siguiente figura (figura 73) se muestra el VI utilizado para ejecutar los controladores Fuzzy antes mencionados.

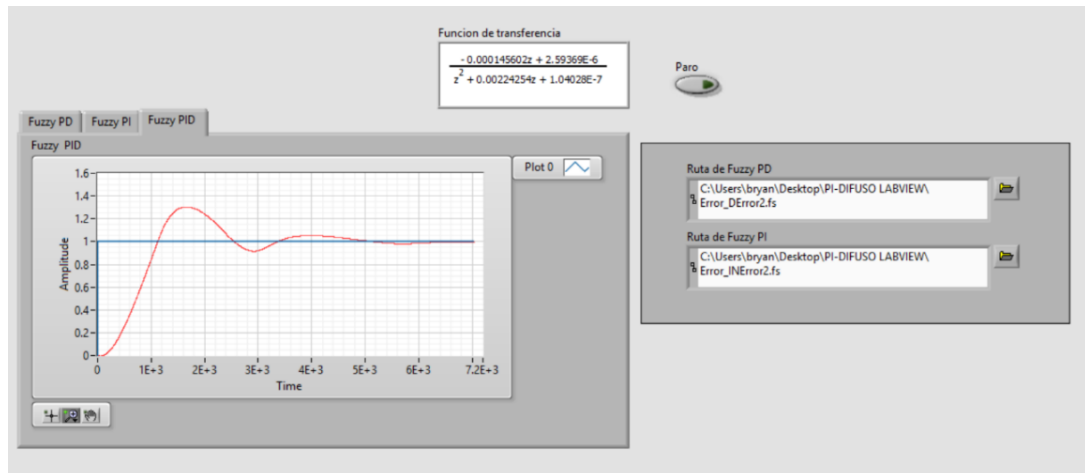


Figura 73:VI de respuestas en lógica difusa

3.5.6.1 Funciones de membresía

Las variables presentes en todos los controladores difusos y con las que se realizan las funciones de membresía de las tres plantas son: Setpoint y Error.

3.5.6.1.1 Variable Error

El error describe la diferencia entre el setpoint fijado por el usuario y la lectura en tiempo real de la variable analizada (temperatura, salinidad u oxígeno disuelto). Esta variable puede tomar valores tanto positivos como negativos, dependiendo del valor en que se encuentre.

Si el valor de la variable se encuentra por debajo del setpoint, el error toma valores positivos. Cuando la variable ha sobrepasado el setpoint fijado, el error toma valores negativos.

3.5.6.1.2 Variable Setpoint

La variable Setpoint toma como valores los rangos de operación permitidos por cada planta y los agrupa en niveles altos, medios y bajos. La acción del controlador de las plantas dependerá en qué nivel se encuentre fijado el Setpoint y cuanto sea el error actual. Los rangos de operación de la planta se muestran a continuación (tabla 10).

Planta	Rango de Operación
Temperatura	24°C- 30,5°C
Salinidad	0 PPT- 30PPT
Oxígeno Disuelto	7PPM- 9,88 PPM

Tabla 10: Rangos de operación de plantas de temperatura, oxígeno y salinidad

3.5.6.2 Diseño de Controlador Difuso para planta de Temperatura

Se inicia con el dimensionamiento del controlador, identificando entradas, salidas y sus variables lingüísticas para cada una de ellas, en este caso, la entrada "Setpoint" posee 5 variables lingüísticas que se definen de la siguiente manera (tabla 11).

B	BAJO
MB	MEDIO BAJO
M	MEDIO
MA	MEDIO ALTO
A	ALTO

Tabla 11: Variables lingüísticas, entrada Setpoint.

La entrada "Error" posee 7 variables lingüísticas, definidas de la siguiente manera (tabla 12).

NA	NEGATIVO ALTO
NM	NEGATIVO MEDIO
NB	NEGATIVO BAJO
Z	CERO
PB	POSITIVO BAJO
PM	POSITIVO MEDIO
PA	POSITIVO ALTO

Tabla 12: Variables lingüísticas, entrada Error

A continuación, se muestran las funciones de membresía tanto de entrada como de salida (figuras 74,75,76) que se diseñaron para cada variable de la planta de temperatura.

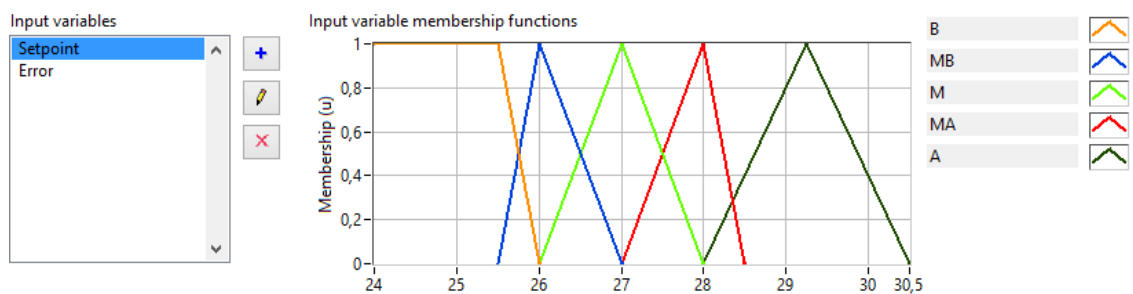


Figura 74: Función de membresía entrada Setpoint. Planta de Temperatura

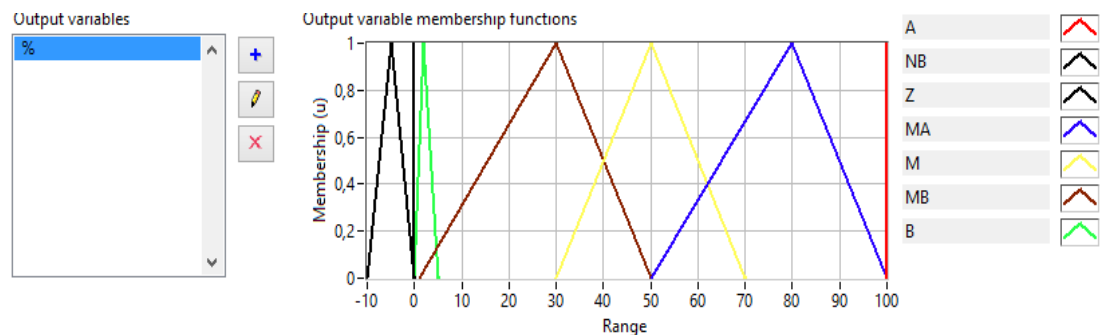


Figura 75: Función de membresía salida %. Planta de Temperatura

En base a las funciones de membresía se realizan las reglas del controlador (tabla13)

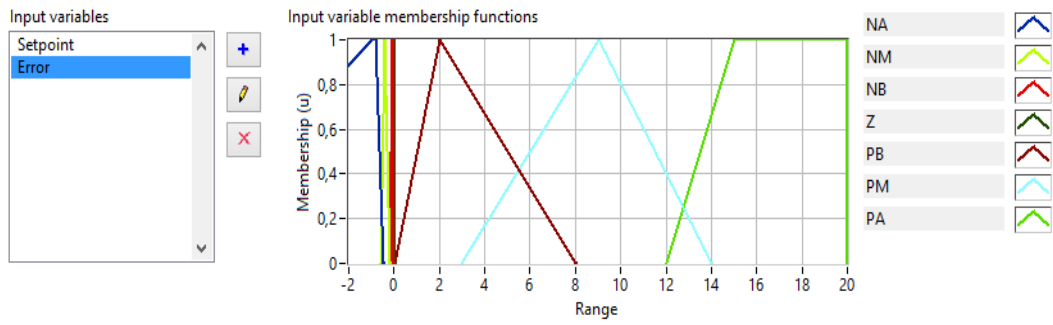


Figura 76: Función de membresía entrada Error. Planta de Temperatura

		Error							
			NA	NM	NB	Z	PB	PM	PA
Setpoint	B	NB	Z	Z	Z	MB	MA	A	
	MB	NB	Z	Z	Z	MB	M	A	
	M	NB	Z	Z	Z	MB	M	A	
	MA	NB	Z	Z	Z	B	MB	A	
	A	NB	Z	Z	Z	B	MA	A	

Tabla 13: Reglas controlador Difuso Temperatura

Para probar el controlador con la función de transferencia de la planta se hace uso del software Labview, usando la herramienta Control & Simulation Loop (figura 77), obteniendo la respuesta simulada del sistema (figura 78).

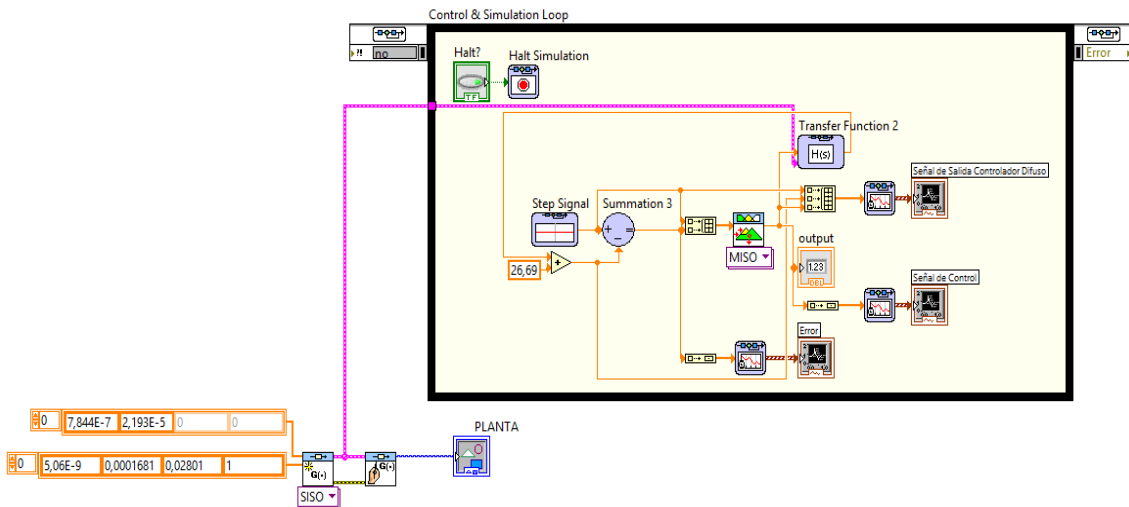


Figura 77: VI de simulación de control Difuso

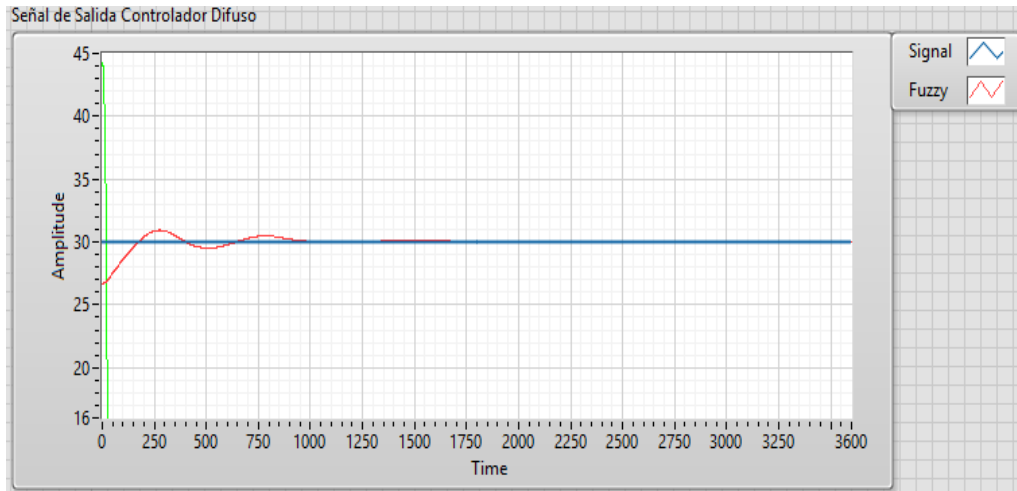


Figura 78: Respuesta de simulación de controlador difuso. Planta Temperatura

El punto de consigna fue fijado a 30°C teniendo un offset de 26,69°C simulando las condiciones normales en que se encontraba la planta. El controlador difuso devuelve los siguientes resultados mostrados en la tabla siguiente tabla (tabla 14).

Tipo	Tiempo de subida <i>Tr (seg)</i>	Tiempo de asentamiento <i>Ts (seg)</i>	Tiempo pico <i>Tp (seg)</i>	Sobrepaso máximo <i>Mp (%)</i>	Error estado estable <i>Ess (%)</i>
Lógica difusa	127,63	171,1	275	20%	0,06%

Tabla 14: Características de control Fuzzy. Planta Temperatura

3.5.6.3 Diseño de Controlador difuso para planta de Oxígeno

Disuelto

A continuación, se muestran las funciones de membresía (figuras 79,80,81) tanto de entrada como de salida que se diseñaron para cada variable de la planta de Oxígeno disuelto.

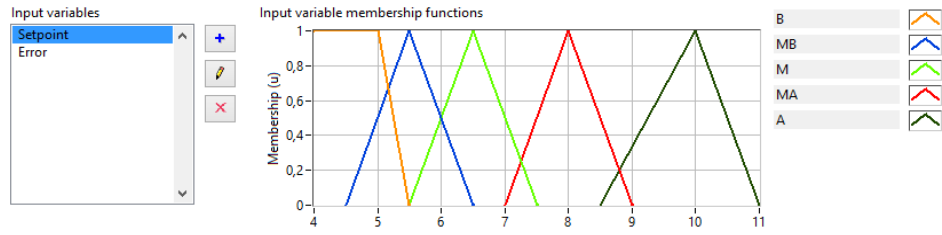


Figura 79: Función de membresía entrada Setpoint. Planta de Oxígeno

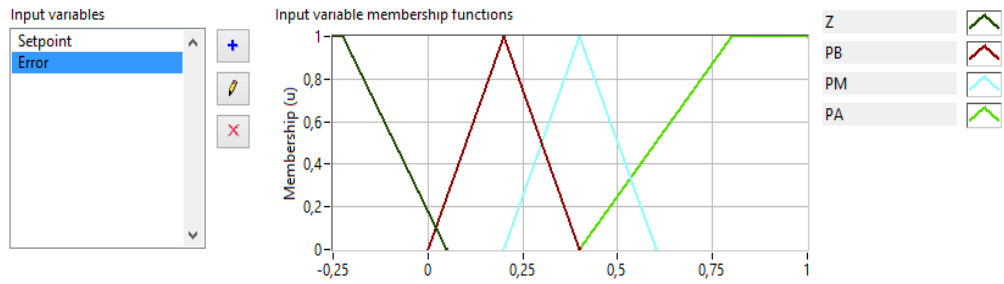


Figura 80: Función de membresía entrada Error. Planta de Oxígeno

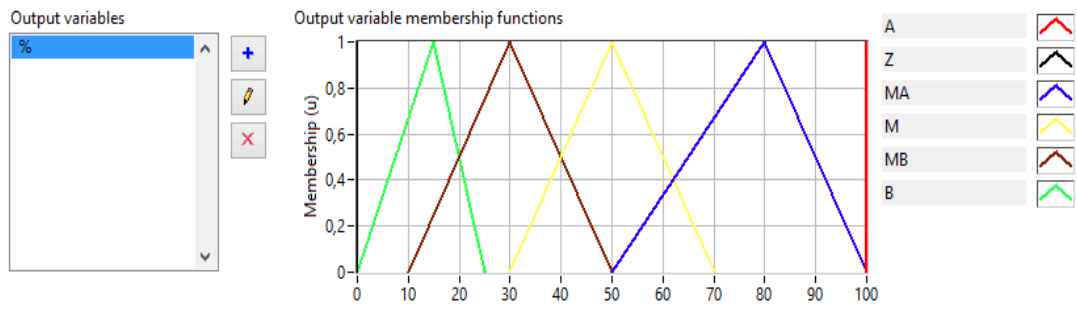


Figura 81: Función de membresía salida % . Planta de Oxígeno

En base a las funciones de membresía se realizan las reglas del controlador (tabla 15).

	Error				
		Z	PB	PM	PA
Setpoint	B	Z	MB	MA	A
	MB	Z	MB	B	A
	M	Z	MB	M	A
	MA	Z	B	MB	A
	A	Z	B	M	A

Tabla 15: Reglas controlador Difuso Oxígeno

Al poseer una función de transferencia de la forma ARX, en la planta de oxígeno se procede a probar el controlador directamente en la planta. El punto de consigna fue fijado a 8 mg/L teniendo un offset de 7 mg/L, que es el oxígeno ya contenido dentro de la piscina, obteniendo el siguiente gráfico (figura 82).

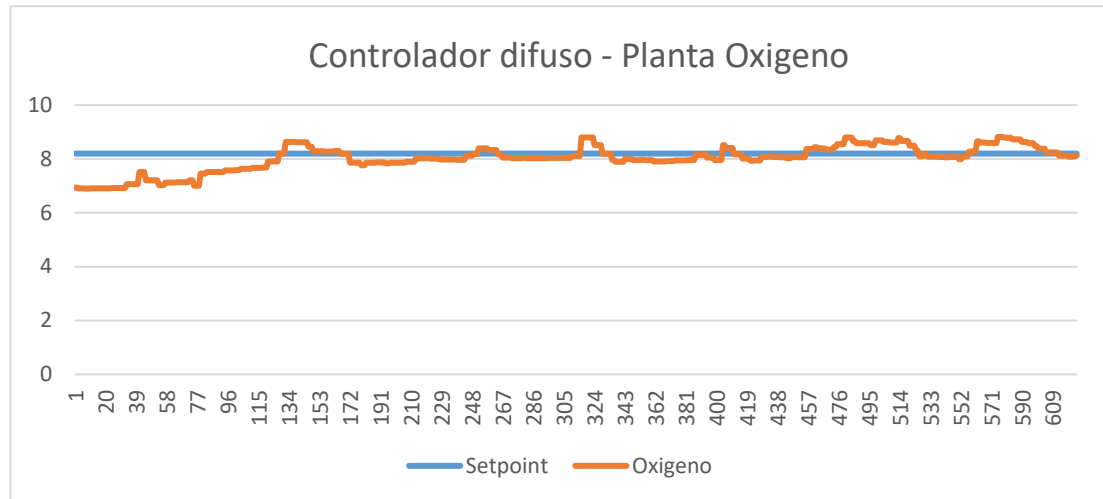


Figura 82: Respuesta del controlador Fuzzy en planta de Oxígeno

El controlador difuso devuelve los siguientes resultados mostrados en la tabla 16.

Tipo	Tiempo de subida T_r	Tiempo de asentamiento T_s	Tiempo pico T_p	Sobrepaso máximo M_p	Error en estado estable E_{ss}
Lógica difusa	126	179	135	30%	-

Tabla 16: Características de control Fuzzy. Planta Oxígeno

3.5.6.4 Diseño de Controlador difuso para planta de Salinidad

Se identifican las variables lingüísticas que conformar las entradas y salidas del controlador teniendo las siguientes (tabla 17 y 18).

B	BAJO
MB	MEDIO BAJO
M	MEDIO
MA	MEDIO ALTO
A	ALTO

Tabla 17: Variables lingüísticas, entrada Setpoint

NA	NEGATIVO ALTO
NM	NEGATIVO MEDIO
NB	NEGATIVO BAJO
Z	CERO
PB	POSITIVO BAJO
PM	POSITIVO MEDIO
PM	POSITIVO MEDIO M
PMA	POSITIVO MEDIO ALTO
PA	POSITIVO ALTO

Tabla 18: Variables lingüísticas, entrada Error

Las funciones de membresía de entradas y salida se muestran a continuación (figura 83,84,85)

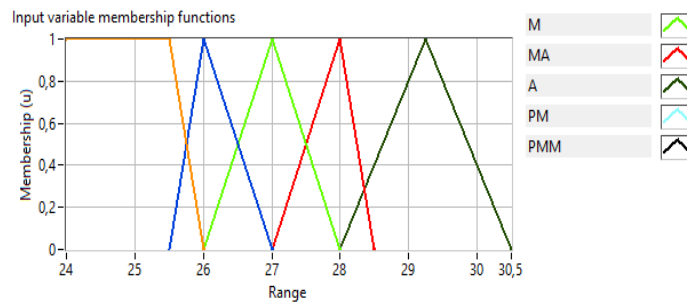


Figura 83: Función de membresía entrada Setpoint. Planta de Salinidad

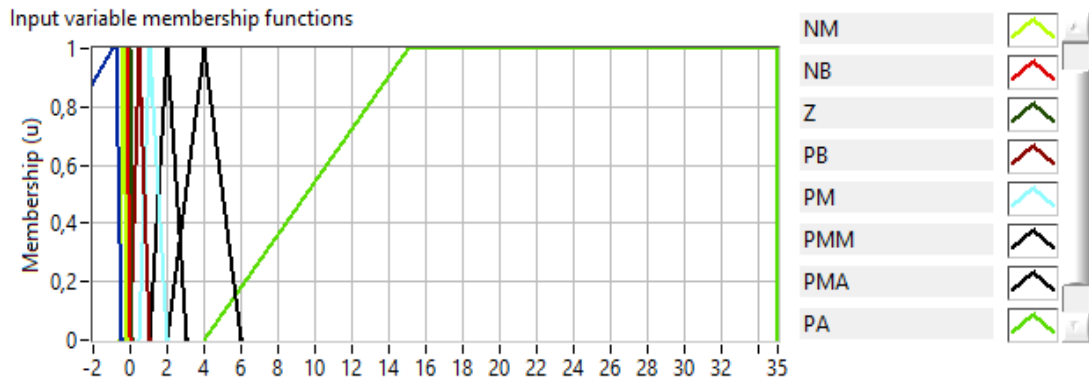


Figura 84: Función de membresía entrada Error. Planta de Salinidad

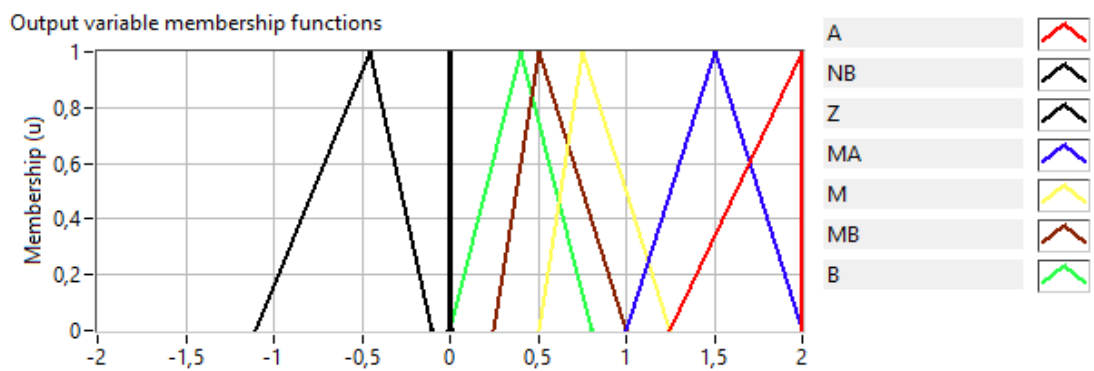


Figura 85: Función de membresía salida %. Planta de Salinidad

Las reglas aplicadas a dicho controlador son las siguientes (tabla 19)

	Error									
		NA	NM	NB	Z	PB	PM	PMM	PMA	PA
<u>Setpoint</u>	B	NB	Z	Z	Z	MB	M	A	B	A
	MB	NB	NB	Z	Z	MB	M	A	B	A
	M	NB	NB	Z	Z	MB	M	A	B	A
	MA	NB	NB	Z	Z	B	MB	A	B	A
	A	NB	NB	Z	Z	B	B	B	B	B

Tabla 19: Conjunto de reglas para controlador difuso. Planta de Salinidad

Para probar el controlador con la función de transferencia de la planta se hace uso del software Labview, usando la herramienta Control & Simulation Loop, obteniendo la respuesta simulada del sistema.

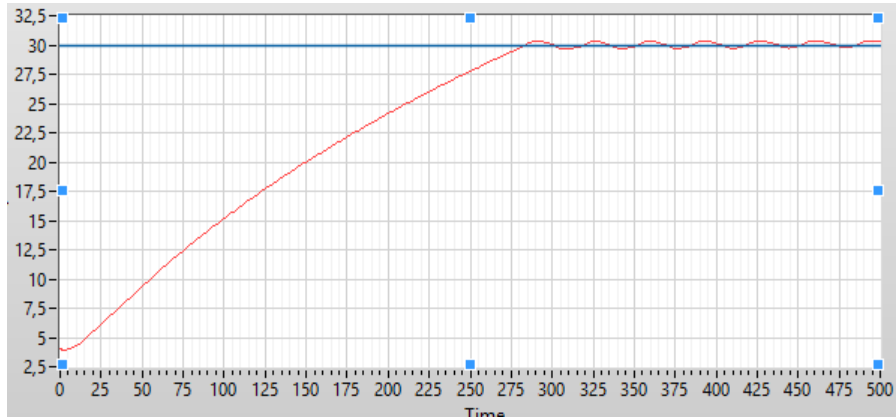


Figura 86: Respuesta de simulación de controlador difuso. Planta Salinidad

El punto de consigna fue fijado a 30° PPT teniendo un offset de 4PPT simulando las condiciones normales de la planta en que se encontraba la planta.

El controlador difuso devuelve los siguientes resultados mostrados en la tabla siguiente tabla (tabla 19)

Tipo	Tiempo de subida T_r	Tiempo de asentamiento T_s	Tiempo pico T_p	Sobrepaso máximo M_p	Error en estado estable E_{ss}
Lógica difusa	215,92 seg	-	291,1 seg	40%	Oscilatorio (8%)

Tabla 20: Características de control Fuzzy. Planta Salinidad

3.6 Software de monitoreo y control

3.6.1 Comunicación del puerto serial XBEE USB

La comunicación de radio por medio del protocolo ZigBee ha dado paso a una revolución de transmisión de datos debido a su bajo coste y gran fiabilidad al momento de recepción dado a su funcionamiento de red punto a punto.

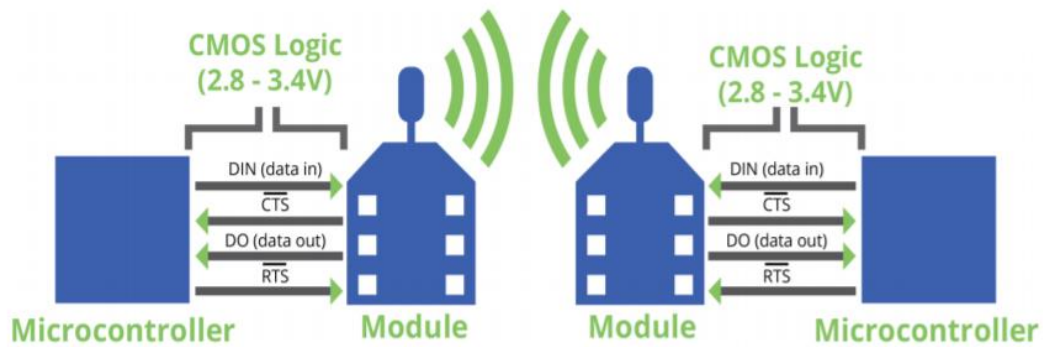


Figura 87: Flujo de datos UART del protocolo ZipBEE (DIGI, 2018)

3.6.1.2 Interconexión

Los módulos XBEE contienen un pequeño buffer al recibir caracteres de forma serial y en radiofrecuencia, el buffer de recepción contiene todos los datos almacenados hasta que estos puedan ser procesados.

De otra manera, el buffer de transmisión colecta los datos que recoge por radiofrecuencia y que será en posterior transmitido de manera UART.

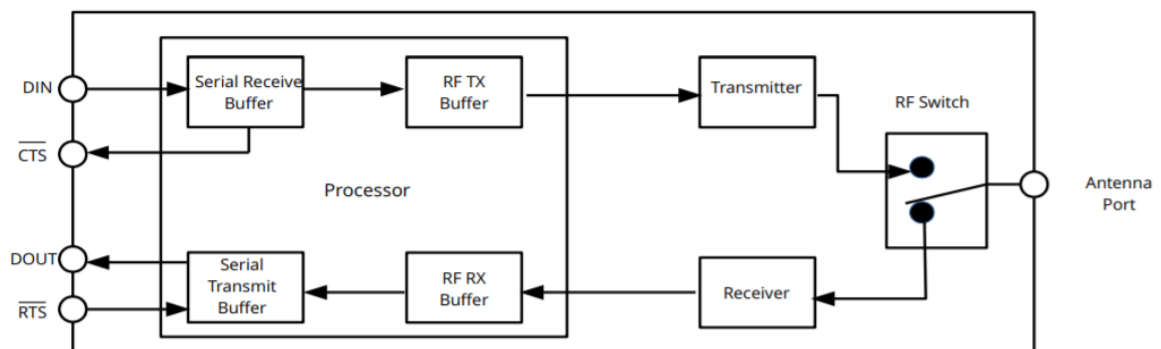


Figura 88: Esquema de transmisión XBEE. (DIGI, 2018)

3.6.1.3 Comunicación entre el PC y el módulo XBee

La comunicación entre el dispositivo XBee y el computador se hace por medio de un adaptador Xbee, que funciona como una interfaz para la recolección de datos por medio de UART.



Figura 89: Adaptador UART XBee

Se emplea la topología Par, la red más simple para la conexión de los módulos XBee se basa en dos nodos, uno de ellos debe ser el coordinador (Conectado al PC) y el siguiente es configurado como dispositivo final (Conectado al Arduino MEGA).

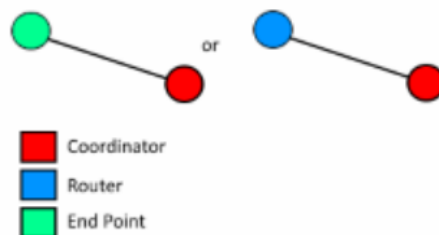


Figura 90: Topología PAR (Belback, 2017)

El flujo de datos se hace a partir de la recepción de datos de los sensores por medio de radiofrecuencia hacia el adaptador Xbee en topología par, una vez recolectado en

el buffer se transmite a Labview haciendo uso del toolkit VISA donde se puede realizar el control SCADA. De igual manera para el nivel de retroalimentación se realiza envío de datos por parte de Labview para la activación de relevadores haciendo uso del firmware que contiene el microcontrolador.

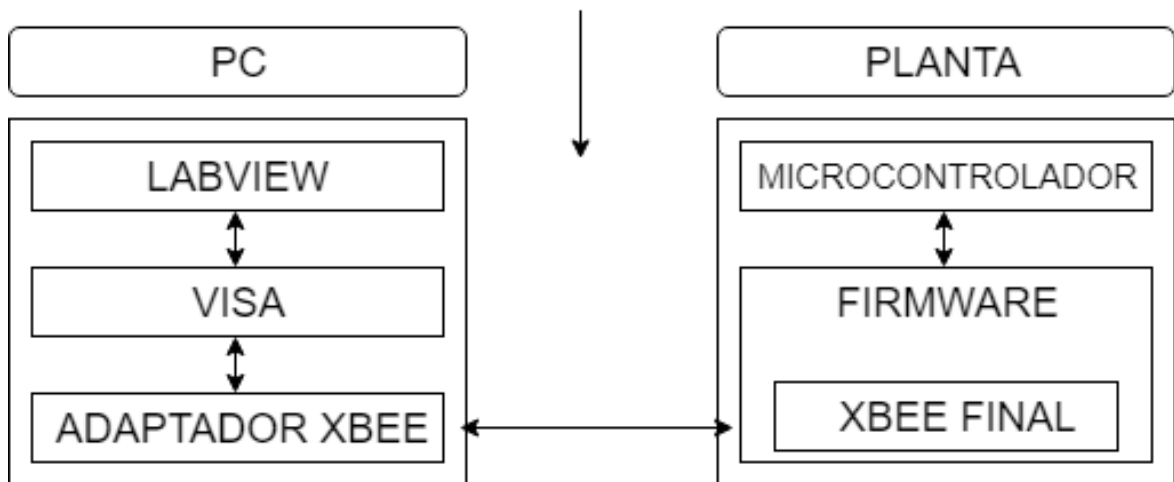


Figura 91: Etapas de comunicación entre PC y la planta.

3.6.1.4 Descripción del VI para el monitoreo y control de la piscina

El diseño del sistema de control y monitoreo se realizó en Labview (figura 91). Muestra en tiempo real los valores de las variables (Salinidad, Temperatura y oxígeno) enviados inalámbricamente desde la planta por el módulo Xbee hacia el computador.

Así como realizar el proceso de control automático de la planta adecuando el agua a la salinidad requerida para luego controlar la temperatura y nivel de oxígeno.

Sin embargo, los controladores del oxígeno disuelto y la salinidad son activados por relevadores electromecánicos y no son controlados por sistema proporcional alguno dado que no se contemplan los actuadores necesarios para este fin. Adicional a esto el sistema tiene la capacidad de manejarse de forma manual para cada actuador utilizado (bombas de agua, electroválvulas, resistencias, bombas de oxígeno).

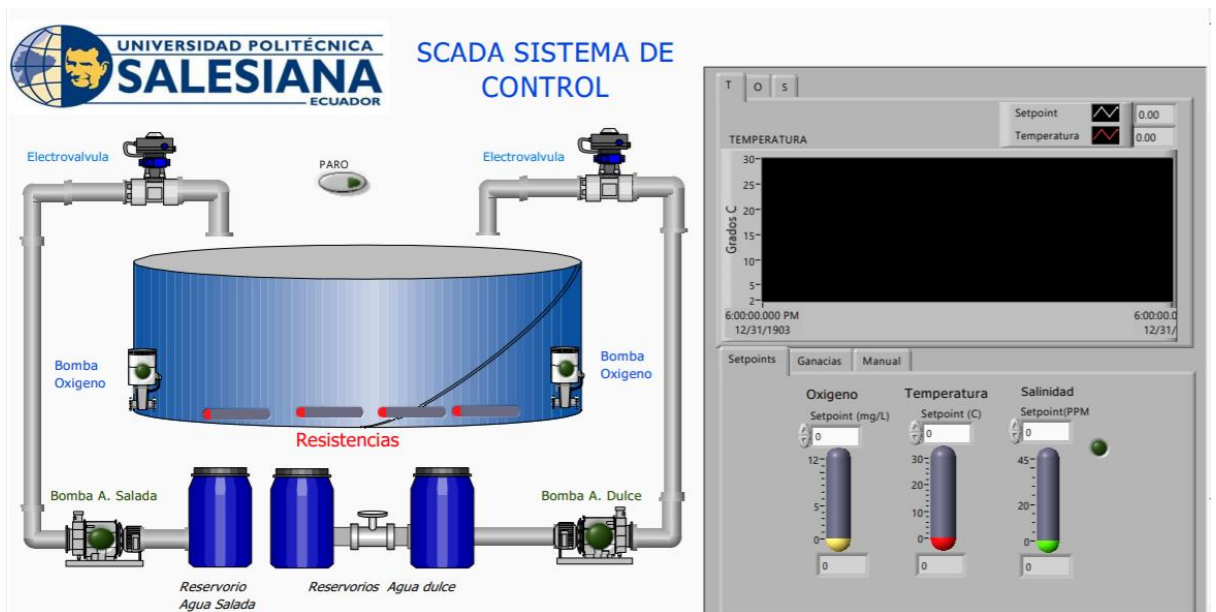


Figura 92: Panel frontal del sistema de control y monitoreo.

A continuación, se explica las distintas partes que componen el VI de monitoreo y control de la piscina.

3.6.1.4.1 Visualización de actuadores

Se presenta de manera dinámica las activaciones de los diferentes actuadores que componen la planta (figura 93).

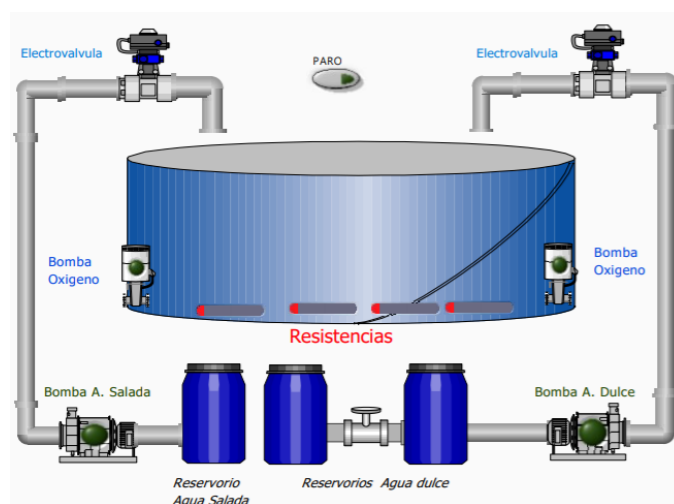


Figura 93: Visualización de actuadores en panel frontal.

3.6.1.4.2 Graficas de valor de sensores y setpoint

En estas graficas (figura 94) se muestran los valores que recogen los sensores de salinidad, temperatura y oxígeno disuelto y los presenta junto al valor del setpoint requerido.

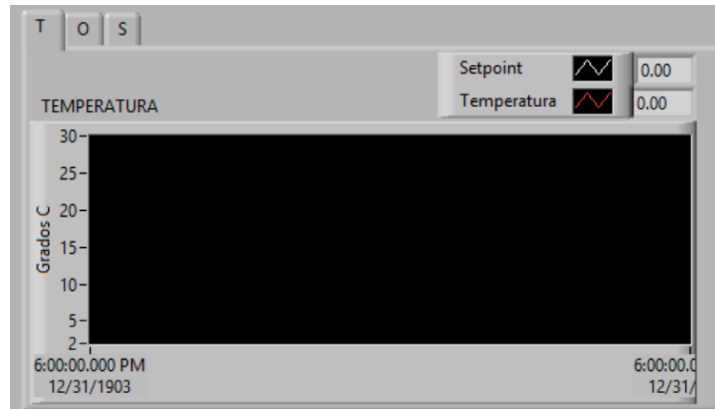


Figura 94: Grafica de valor de temperatura

3.6.1.4.3 Setpoints

En este contenedor se configuran los valores de setpoints a los cuales se desea que llegar para estabilizar la piscina (figura 95).

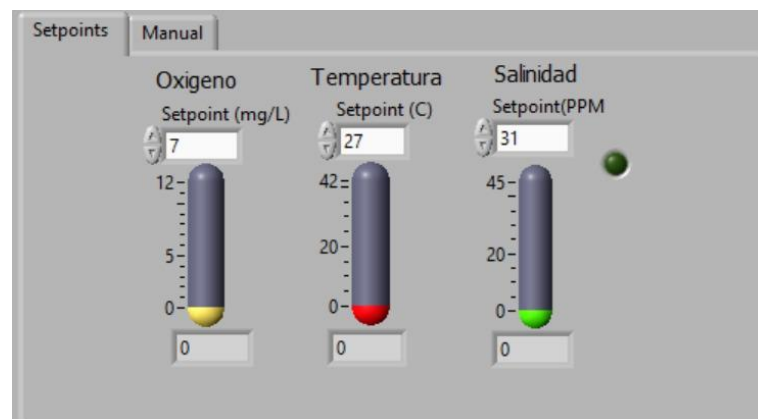


Figura 95: Valores de Setpoint de las variables de la planta

3.6.1.4.4 Controles manuales

Tal como se indica en la figura 96, se puede realizar la activación manual de los diferentes actuadores presentes en la piscina, de esta manera la activación

automática no se realizará por el lapso de tiempo que interruptor “Activar” se encuentre encendido.

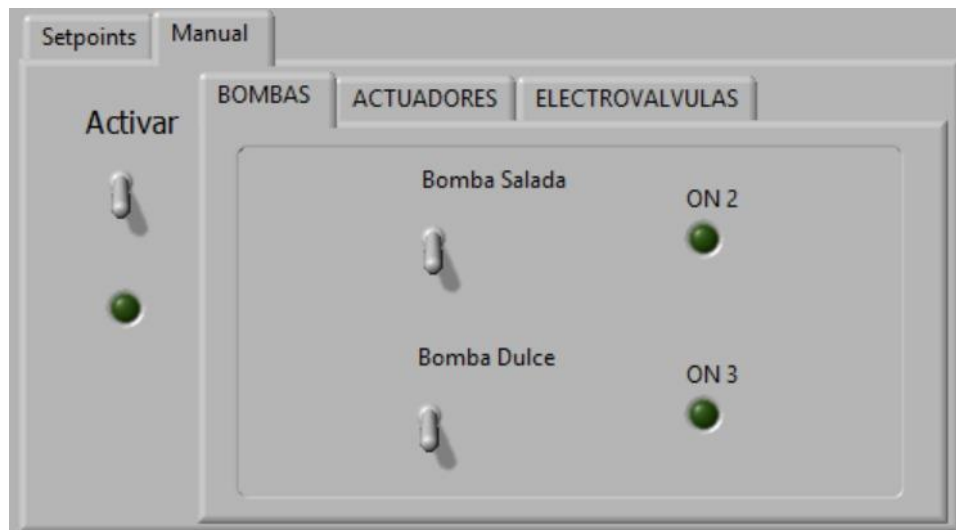


Figura 96: Activación manual de los actuadores de la planta

4. RESULTADOS

4.1. Análisis de resultados

El objetivo de este capítulo es presentar un análisis sobre el control de la planta de temperatura, sin embargo, se han realizado los procesos de salinidad y oxígeno de igual manera (sin englobar actuadores proporcionales) con propósitos académicos.

4.1.1 Planta Temperatura

En esta planta se prioriza el tiempo de asentamiento y un error de estado estable bajo, el valor final no debe superar el 5%, al ser la planta de temperatura un sistema de respuesta lenta y con un retardo (tiempo muerto) por naturaleza de esta.

El controlador Fuzzy ha sido elegido para ser implementado en la planta real, siendo este programado en el VI principal (figura 97).

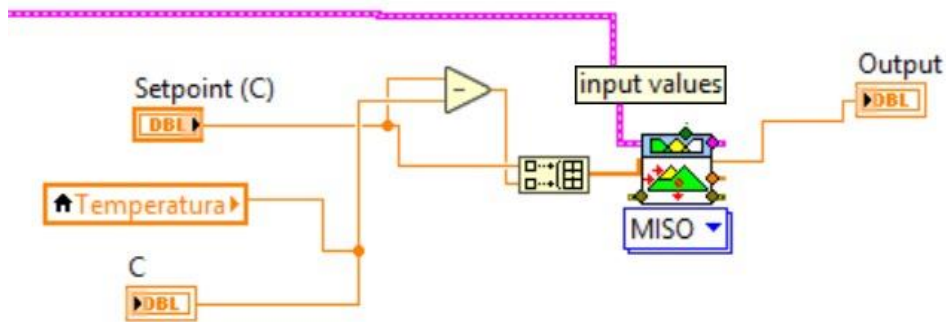


Figura 97: Controlador Fuzzy programado en VI principal de planta

Se obtiene la curva de temperatura (C) versus tiempo (sg) con un setpoint de 26.6 C como se puede apreciar en la figura 98, donde se puede comprobar como el sistema realiza la activación adecuada de las resistencias caloríficas para el aumento de temperatura dentro del prototipo.

Esto lo realiza en un tiempo de 00:27min:12seg, esto es debido al volumen de agua en el que se encuentra contenido.

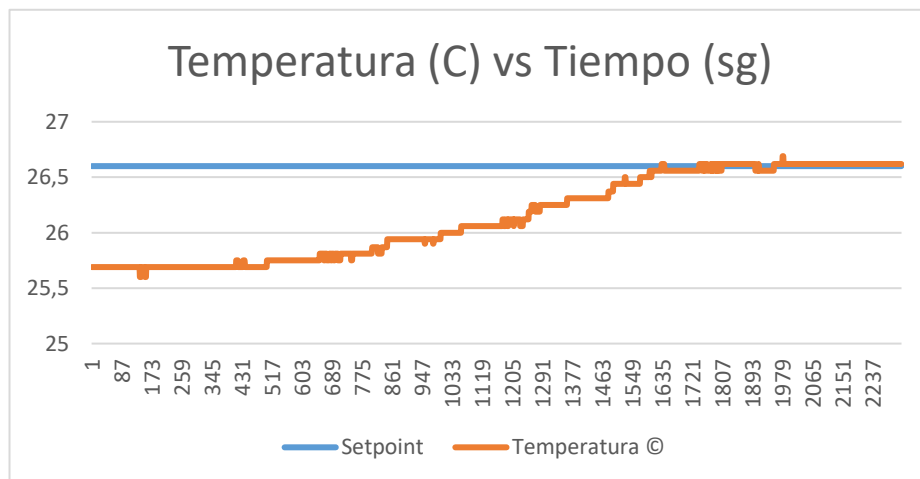


Figura 98: Curva temperatura vs tiempo. Valores reales de la planta de Temperatura

Se puede visualizar en la figura 99 la salida del controlador Fuzzy la cual entrega un porcentaje de 0 a 100% este es quizás el control más sencillo de implementación

debido a que la temperatura no presenta variaciones de grandes magnitudes en un tiempo corto.

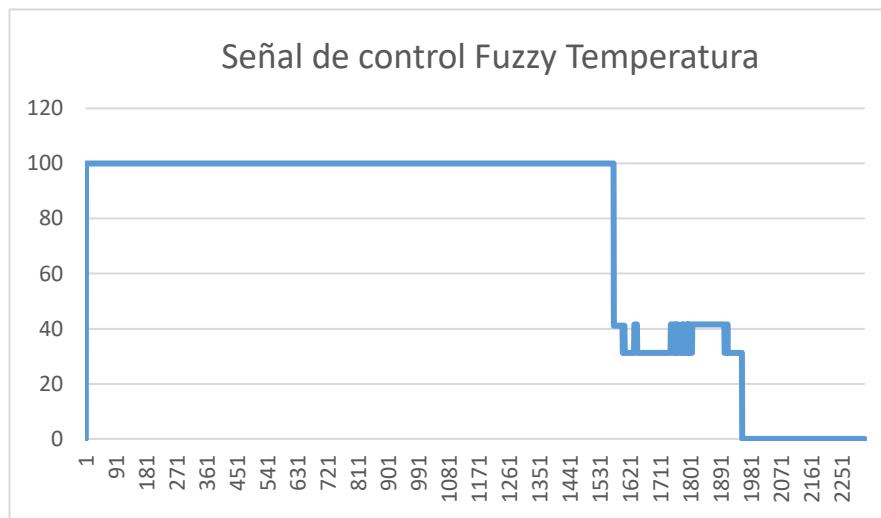


Figura 99: Señal de control dada por el controlador Fuzzy en planta Temperatura

Las activaciones realizadas por parte del SSR que maneja las resistencias caloríficas se puede apreciar en la figura 100

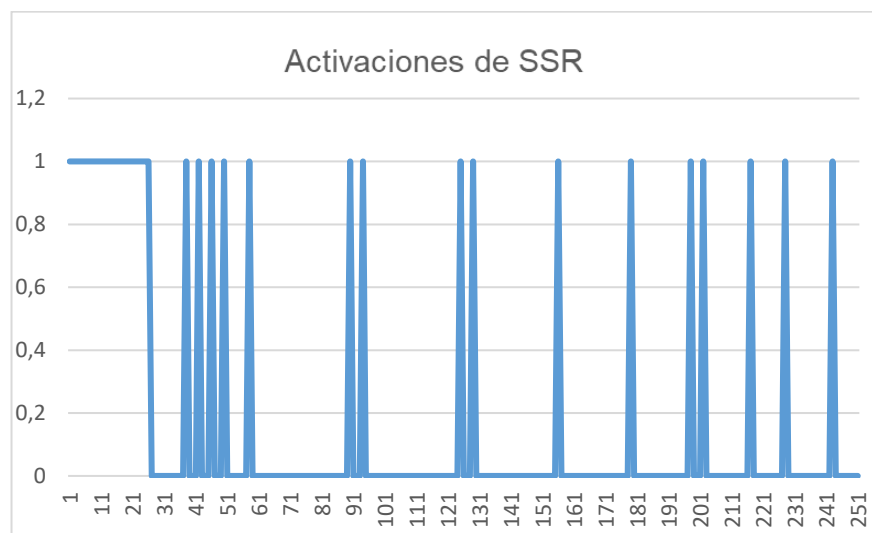


Figura 100 Activación de resistencias caloríficas

Se realiza la prueba de perturbación del sistema para analizar la rapidez de su estabilización y puede ser apreciada en la imagen 101. En la misma se comprueba

que luego de ingresar un volumen de agua de 10 lts con una temperatura de 15 C esta baja considerablemente lo que a su vez es rápidamente corregido por accionamiento de las resistencias caloríficas y en la que alcanza a llegar al setpoint de 27.8 C en alrededor de 00:08 min:00 seg la cual su activación se puede apreciar en la figura 102.

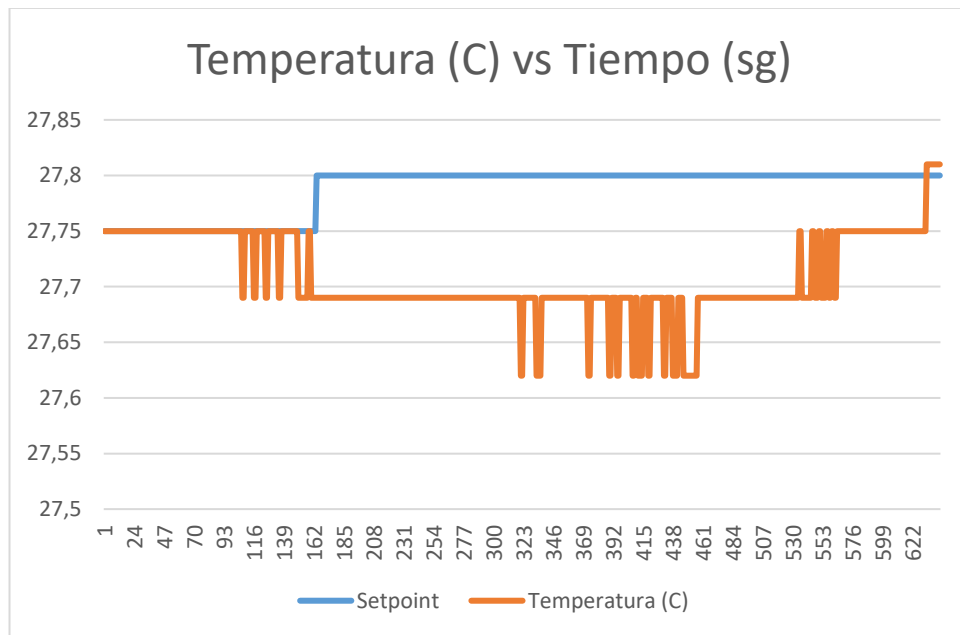


Figura 101 Curva temperatura vs tiempo en perturbación

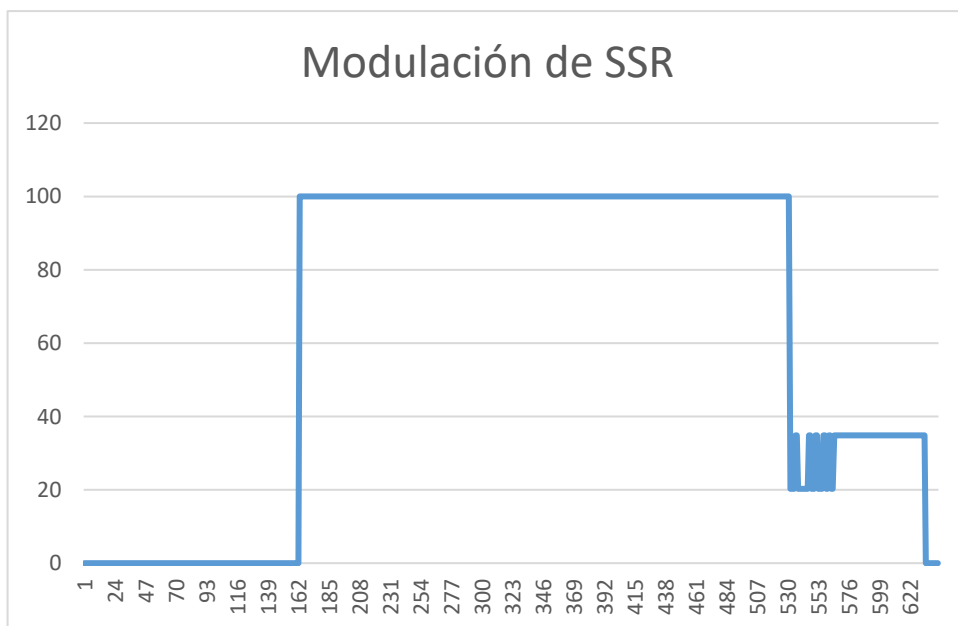


Figura 102: Respuesta de la señal de control ante perturbación en planta de Temperatura.

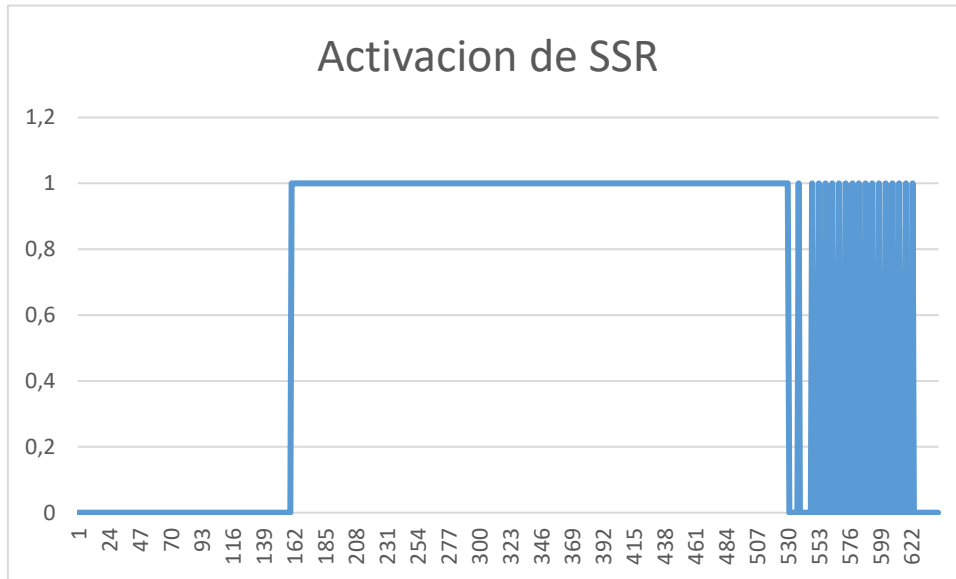


Figura 103: Activación de resistencias caloríficas en perturbación

4.2.2 Planta Salinidad

El proceso del control de salinidad se realiza una sola vez en la planta, fijando el setpoint de operación requerido obteniendo la curva de la salinidad (ppt) versus tiempo (sg) con un setpoint de 30 ppt como se puede apreciar en la figura 104, donde se puede comprobar como el sistema realiza la suministración adecuada de agua salada en base al requerimiento sin sobrepasar el rango de error. Esto lo realiza en un tiempo de 00:04min :01 seg.

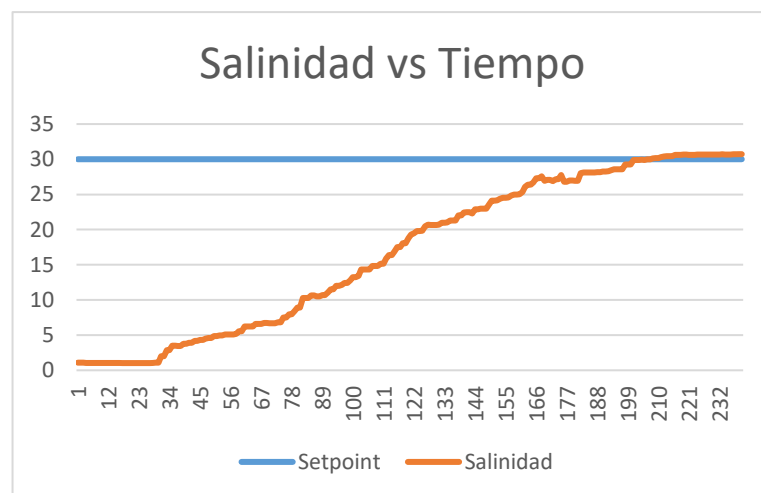


Figura 104: Curva Salinidad (ppt) vs Tiempo (sg)

Las activaciones realizadas por parte del relay de fuerza que maneja la bomba de agua salada se puede apreciar en la figura 105, se puede visualizar también como, conforme se acerca al setpoint, el sistema realiza un cierto periodo de pausa para poder abrir paso a la mezcla de solidos disueltos por parte de las bombas de aireación.

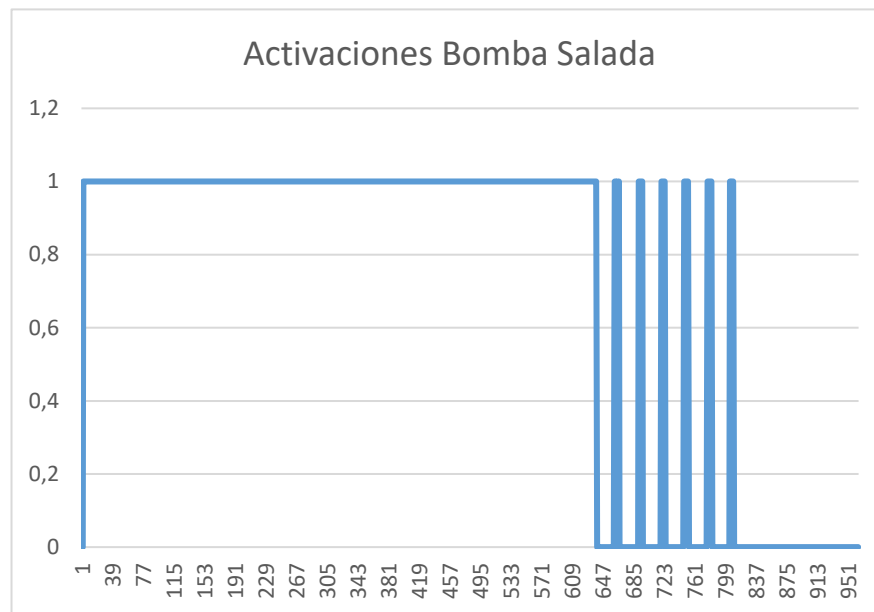


Figura 105: Activaciones de bomba de agua salada

Se realiza la prueba de perturbación del sistema para analizar la rapidez de su estabilización y puede ser apreciada en la imagen 10, en la misma se comprueba que existe un impulso de agua salina dentro de la piscina prototipo, y esta a su vez es rápidamente corregido por accionamiento de la bomba de agua dulce la cual su activación se puede apreciar en la figura 106.

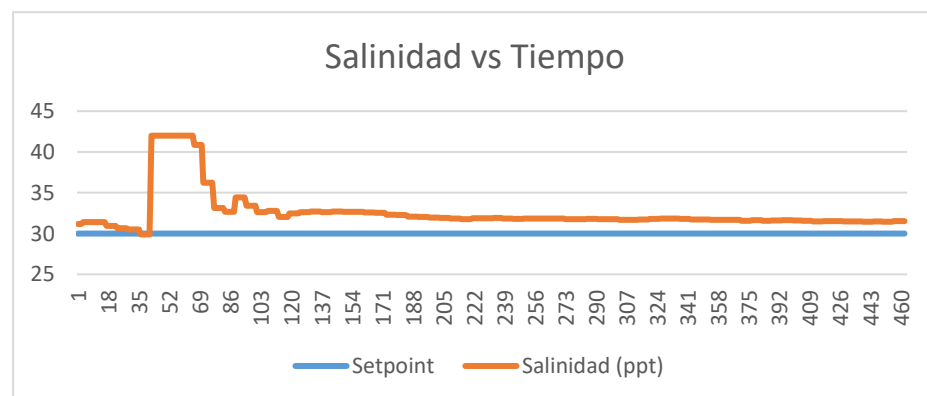


Figura 106 Curva salinidad vs tiempo en perturbación

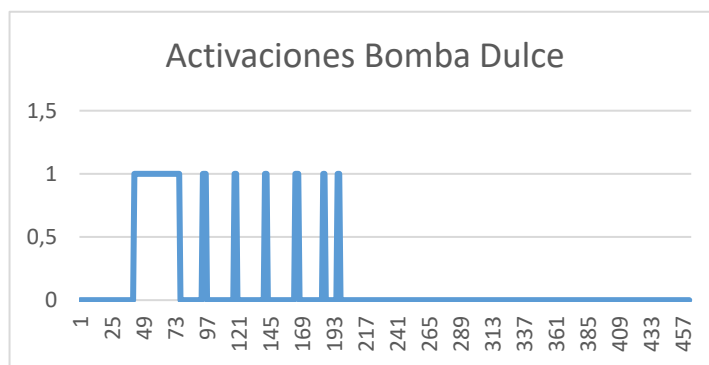


Figura 107 Activaciones de bomba dulce ante perturbación en planta de Salinidad

4.2.3 Análisis de resultado del oxígeno disuelto

Para el control del oxígeno disuelto se implementa el controlador Fuzzy anteriormente diseñado, ya que, al poseer una función de transferencia dinámicamente compleja, el controlador Fuzzy brinda una gran ventaja al no valerse de la función para poder ser diseñado e implementado.

Se obtiene la curva del oxígeno disuelto (ppm) versus tiempo (sg) con un setpoint de 8 ppm como se puede apreciar en la figura 108, donde se puede comprobar como el sistema realiza la activación adecuada de la bomba de aire para la reaireación del sistema. Se aprecia también las variaciones que existen en la planta y por lo que el sistema está constantemente regulándose.

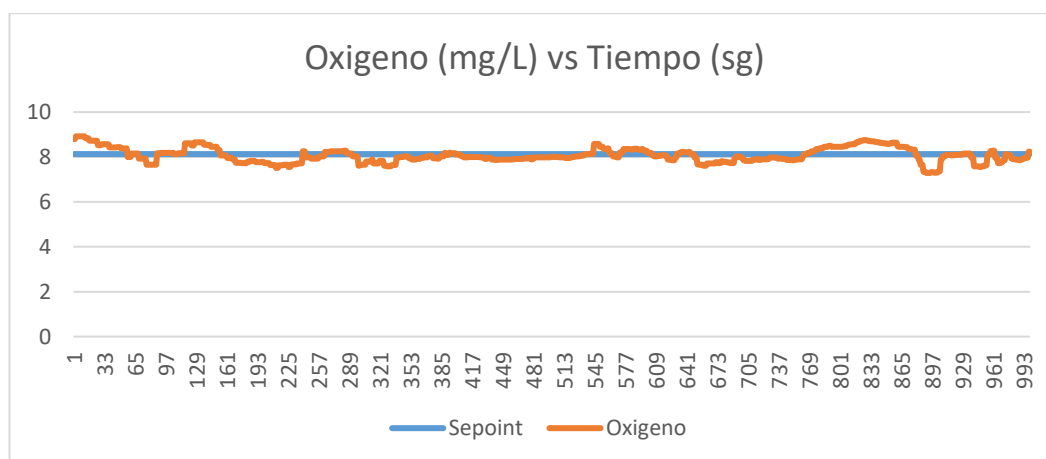


Figura 108 Curva oxigeno vs tiempo

Las activaciones realizadas por parte del relay de control que maneja la bomba de aire se puede apreciar en la figura 109, se puede visualizar también en la figura 110 las modulaciones en carácter porcentual de la salida además de cómo, una vez establecido el setpoint y en conformidad sobre la demanda de oxígeno este decremента, la misma empieza a realizar el tren de pulsos para mantenerse en el rango deseado.

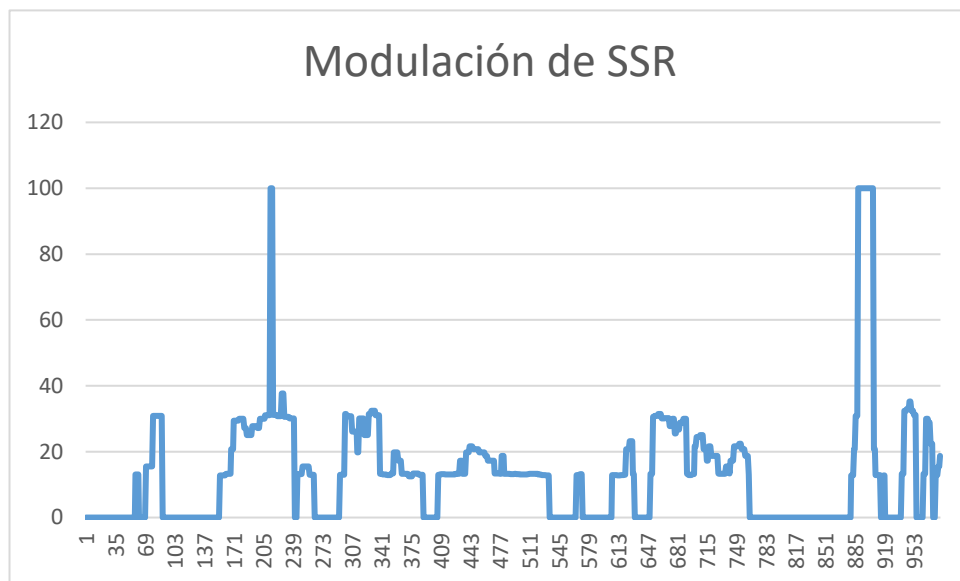


Figura 109 Respuesta de la señal de control en planta de oxígeno

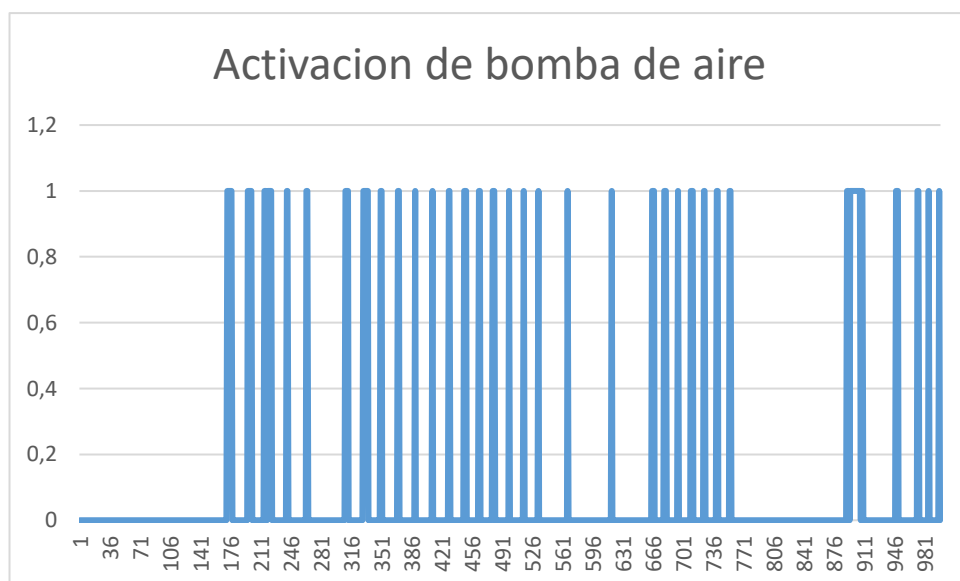


Figura 110 Activaciones de bomba de aire

Se realiza la prueba de perturbación del sistema para analizar la rapidez de su estabilización y puede ser apreciada en la imagen 111, en la misma se comprueba que luego de forzar al sistema a la desactivación manual de la bomba de aireación esta baja considerablemente este a su vez es rápidamente corregido por accionamiento de la bomba de aireación con lo que consigue llegar al setpoint de 8 ppm en alrededor de 00:00min:30 seg la cual su activación se puede apreciar en la figura 112.

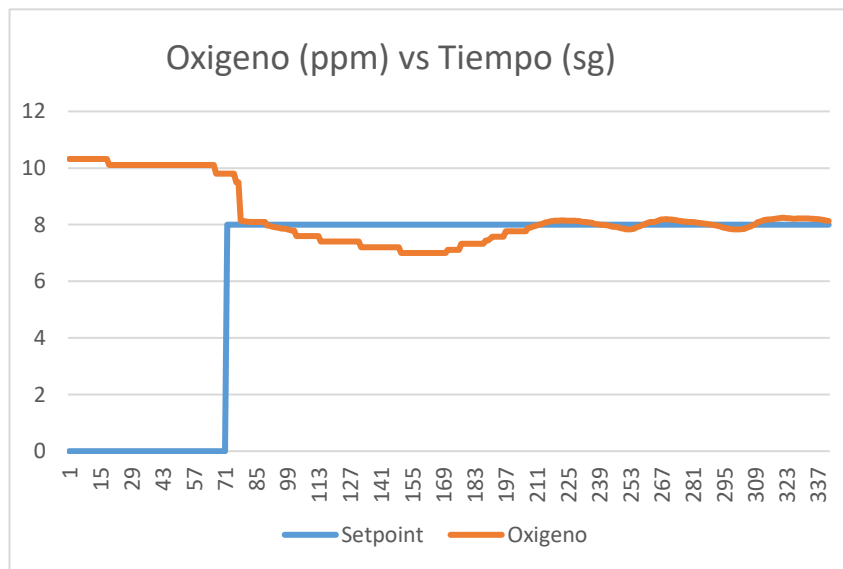


Figura 111: Curva oxigeno vs tiempo en perturbación

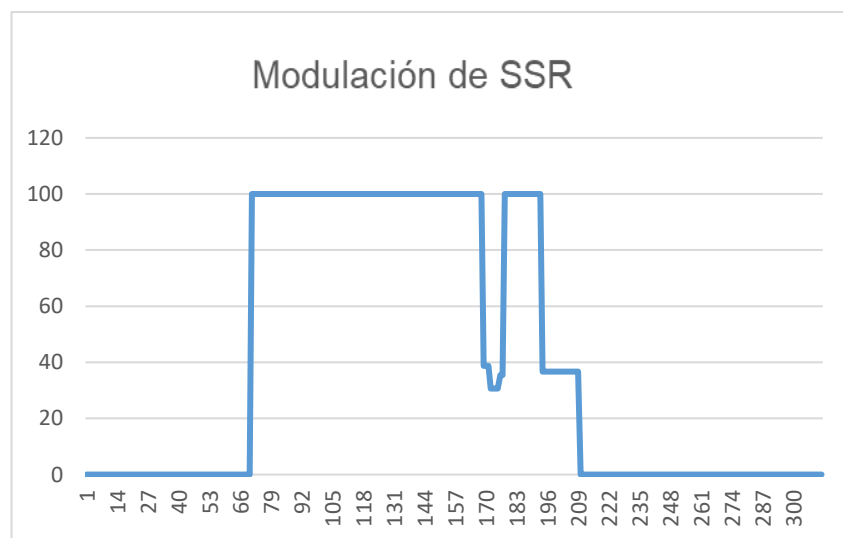


Figura 112: Respuesta de la señal de control ante perturbación en planta de Oxígeno

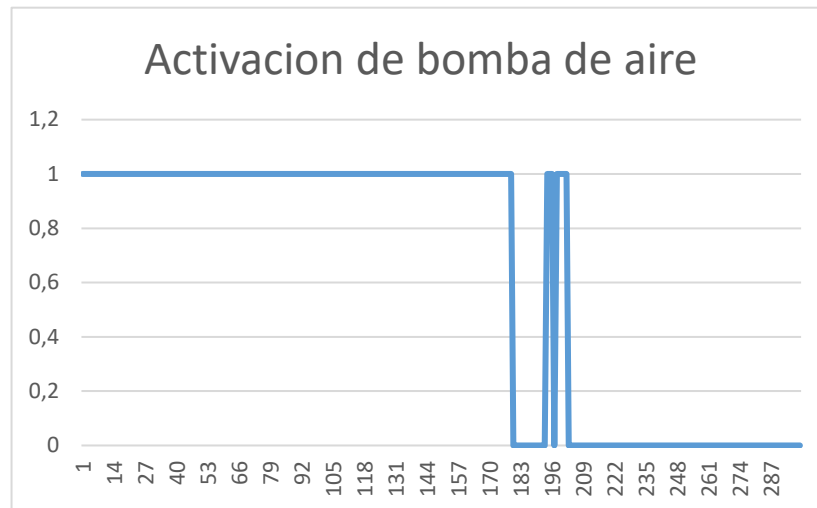


Figura 113: Activación de bomba de aire en perturbación

Conclusiones

- En base a los resultados se puede concluir que para la planta de temperatura en los controladores clásicos el control PD se halla como el controlador óptimo dentro de este grupo debido a su nulo sobrepaso máximo, sin embargo, el controlador de lógica difusa prevalece debido a su bajo tiempo de asentamiento y un error en estado estable menor del 1%, como se muestra en el capítulo 4.
- En base a los resultados se puede concluir que para la planta de salinidad en los controladores clásicos el control PID se halla como el controlador óptimo dentro de este grupo debido a su bajo sobrepaso máximo y también bajo error en estado estable, siendo mejor que incluso el controlador de lógica difusa.
- En base a los resultados se puede concluir que para la planta de oxígeno en los controladores clásicos el control PID se halla como el controlador óptimo dentro de este grupo debido a su alto tiempo de subida, bajo error en estado estable y un menor tiempo pico, sin embargo, el controlador de lógica difusa prevalece debido a su bajo tiempo de asentamiento y rápidas oscilaciones que mejoran la estabilidad del oxígeno.

- Se diseña la estructura de tal manera que pueda soportar un volumen máximo de 200 litros, además las adecuaciones para la adquisición tanto de agua salina como de salada, se realiza un tablero de control encapsulado con la finalidad de evitar contacto directo con los fluidos.
- Se desarrolla los modelos matemáticos simulados en base a previas experimentaciones y se analizan los mejores casos de acción para el efecto de estos, además de generar una señal de pulsos variantes para una mejor obtención de función de transferencia.
- El controlador Difuso posee una mayor efectividad que el controlador PID para el caso de la planta de Temperatura y Oxígeno, teniendo un menor tiempo de subida y tiempo de asentamiento acelerando el proceso de calentamiento de gran cantidad de agua (100 litros) sin afectar el error en estado estable.
- El uso de un relevador de estado estable SSR ofrece una gran ventaja para el control de elementos resistivos, como lo son los actuadores de la planta de Temperatura, brindando una modulación proporcional mediante la variación del Duty Cycle.
- Se evalúa la capacidad de comunicación por parte de los adaptadores XBEE que permitieron una distancia remota máxima de 15 metros con obstáculos a la redonda que fueron de gran ayuda en el desarrollo del proyecto, debido a que cualquier elemento computacional cercano que no cumpla con exigencias industriales puede ser fácilmente malogrado por parte del caudal y la cantidad de sal que puede contener.
- Los actuadores, bombas de agua, bombas de aire, y resistencias caloríficas, aunque considerablemente económicas cumplen con el cometido del proyecto, desarrollando un sistema estable ante perturbaciones esto debido al sistema eléctrico de activación que se implementa.

Recomendaciones

- Se debe evitar la conexión directa los sensores de calidad de agua Atlas Scientific, esto debido al gran espectro de corrientes parasitas que puedan estar contenido el circuito, en su defecto contactar con el fabricante y utilizar un aislador específico para el mismo.
- Se recomienda analizar la cantidad de Cloruro de sodio que se ingresa en el reservorio, esto debido a que el mismo puede llegar a saturar el sensor de conductividad y desencadena en un gasto para el usuario, de igual manera se recomienda revisar a detalle los rangos máximos de medición para la conductividad o salinidad.
- Se recomienda realizar una limpieza total de los reservorios una vez finalizado el proceso, esto con el fin de mantener lejos alguna impureza que pueda acarrear en suciedad para la piscina, además de la limpieza regular de la piscina debido a las sales que puedan encontrarse.
- No se debe manipular en ninguna circunstancia el tablero de control o fuerza, debido a que el panel de control maneja corrientes bajas y la estática que el tacto puede mantener podría perjudicar la comunicación y desencadenar en una incorrecta activación/desactivación de los actuadores. Mientras que en el tablero de fuerza maneja corrientes que pueden ser peligrosas para el ser humano.
- Para una mejor integración de las sales se recomienda agitar manualmente el reservorio de agua salada antes de iniciar el proceso, esto con el fin de que los sólidos disueltos que se encuentran dentro no permanezcan en la parte baja del mismo dando lugar a una mayor concentración de sales que se degraden en conformidad con el tiempo.

- Siempre que sea posible realizar una limpieza del sensor de salinidad para mantener una magnitud de lectura igual a cero, debido a que la sal se puede cristalizar y a lo largo del mismo y se puede obtener valores erróneos.

Bibliografía

- Åkesson, M. (1998). Control of Dissolved Oxygen in stirred bioreactors. *Lund Institute of Technology*.
- Association, A. P. (1999). *Standard Methods for the Examination of Water and Wastewater*.
- Atlas Scientific. (24 de 7 de 2018). *Atlas Scientific*. Obtenido de <https://www.atlas-scientific.com/>
- Bain, M. B. (1999). *Aquatic habitat assessment*. Bethesda, Maryland: American Fisheries Society.
- Benito, T., & Isabel, V. (8 de Septiembre de 2010). *Departamento de Ingeniería Telemática*. Obtenido de Universidad Carlos III de Madrid: <http://www.it.uc3m.es/jvillena/irc/practicas/08-09/10.pdf>
- Bett, C. (2009). Combined effect of body weight, temperature and salinity on shrimp *Litopenaeus vannamei* oxygen consumption rate. *Brazilian Journal of Oceanography*.
- Boltzmann, L. (1986). *Escritos de mecánica y termodinámica*. .
- Boyd, C. E. (1992). Consideraciones sobre la calidad de agua y suelo en cultivos de camarón. *Department of Fisheries and Allied Aquacultures*.
- Boyd, C. E. (2001). Consideraciones sobre la calidad del agua y del suelo en cultivos de camarón.
- Burmeister, L. C. (1993). *Convective Heat Transfer* .
- Charlton, W. (1968). Linear System Identification using Pseudo Random.
- Clasificaciones, E. d. (2016). *tiposde*. Obtenido de tiposde: <http://www.tiposde.org/salud/182-tipos-de-hormonas/>
- Cortez, K. A. (2006). *Procedimientos técnicos para producción de nauplios de la Empresa TEXCUMAR S.A, Ecuador*. Honduras.
- CRUZ, A. J. (1999). Estimation of the volumetric oxygen transfer coefficient (KLa) from the gas balance and using a neural network technique. *Brazilian Journal of Chemical Engineering*.
- DFRobot. (2008). *Waterproff DS18B20*. Obtenido de <https://www.dfrobot.com/product-689.html>
- DFRobot. (24 de 07 de 2018). Obtenido de Gravity:Terminal Sensor Adapter V2.0: www.dfrobot.com
- DIGI. (2018). *DIGI.com*. Obtenido de <https://www.digi.com/resources/documentation/digidocs/pdfs/90000976.pdf>

- DMedicina. (10 de 2016). *DMedicina*. Obtenido de DMedicina: <http://www.dmedicina.com/>
- Gerrig, R. J. (s.f.). *Psicología y vida*. PEARSON Educacion.
- Gomez, C. S. (2010). *Apunte de procesos quimicos* .
- Goyenola, G. (2007). Guía para la utilización de las Valijas Viajeras . *Red de Monitoreo Ambiental Participativo de Sistemas Acuáticos*.
- IctioTerm. (2005). Obtenido de <http://www.ictioterm.es>
- Jenkins, H. (2016). *PID controllers*. Obtenido de webtasks: <https://webtasks.mercer.edu/FacultyWebDirectory/>
- LEWIS, E. (s.f.). The practical salinity scale 1978 and its antecedents . *IEEEJ. Oceanic*.
- Lima, B. (22 de Mayo de 2016). *Lógica Difusa: Aspectos Básicos*. Obtenido de Universidad Politécnica Salesiana: <http://virtual.ups.edu.ec/presencial48/>
- M.Hill. (1970). *The Sea*.
- Mathworks. (2018). *Matlab*. Obtenido de <https://la.mathworks.com/>
- Ogata, K. (2010). *Ingeniería de Control Moderna*. Madrid: PEARSON EDUCACIÓN S.A.
- Pèrez, C. (2015). *Naturpsico*. Obtenido de Naturpsico: <http://www.naturpsico.net/las-hormonas-en-el-comportamiento/>
- Picuino. (Marzo de 2017). *Controlador PID*. Obtenido de <http://www.picuino.com/es/arduprog/control-pid.html>
- Ponce, P. (2010). *Inteligencia Artificial Con Aplicaciones En La Ingenieria*. México: Alfaomega.
- Regalado, A. (2008). Cómo hacer un modelo matemático. *UTM, Temas de ciencia y tecnología*, 11.
- Sanjuán, J. (Agosto de 2007). *Universidad Tecnológica de la Mixteca*. Obtenido de <http://www.utm.mx/>
- Smith-Corripio. (1997). *Principles and Practice of*. New York: Jhon Wiley & Sons.
- Tourlousse, D. (2007). Design of an Experimental Unit for the Determination of Oxygen Gas-Liquid Volumetric Mass Transfer Coefficients using the Dynamic Re-oxygenation Method. *Michigan State University*.
- UNAM. (14 de Noviembre de 2006). *Reglas de sintonia de controladores PID*.
- UNESCO. (1988).
- Villajulca, J. (1 de Abril de 2011). *Instrumentación y Control*. Obtenido de <http://www.instrumentacionycontrol.net/cursos-libres/automatizacion/curso-control-pid-practico/item/364-el-control-proporcional-definiciones-pr%C3%A1cticas-y-precisas.html>



ANEXOS

ANEXO 1: DATASHEET SENSOR DE SALINIDAD C0 ATLAS SCIENTIFIC K1.0

Conductivity Probe K 1.0

Graphite

Reads	Conductivity
Range	5 – 200,000 $\mu\text{S}/\text{cm}$
Accuracy	+/- 2%
Response time	90% in 1s
Max pressure	3,447 kPa (500 PSI)
Max depth	343 meters (1,125 ft)
Temperature range °C	1 – 110 °C
Cable length	1 meter
Internal temperature sensor	No
Time before recalibration	~10 years
Life expectancy	~10 years
Maintenance	N/A



Revised 2/12/19

Dissolved Oxygen Probe

Reads	Dissolved Oxygen
Range	0 – 100 mg/L
Accuracy	+/- 0.05 mg/L
Response time	~0.3 mg/L/per sec
Max pressure	3,447 kPa (500 PSI)
Max depth	343 meters (1,125 ft)
Temperature range °C	1 – 50 °C
Cable length	1 meter
Internal temperature sensor	No
Time before recalibration	~1 Year
Life expectancy	5 Years +
Maintenance	~18 Months



ANEXO 3: FIRMWARE DE MICROCONTROLADOR ARDUINO

```
1. #include <OneWire.h>
2. String inputstring = "";
3. String sensorstring = "";
4. String sensorstring2 = "";
5. boolean input_string_complete = false;
6. boolean sensor_string_complete = false;
7. boolean sensor_string_complete2 = false;
8. String inputs;
9. #define relay1 2
10. #define relay2 3
11. #define relay3 4
12. #define relay4 5
13. #define relay5 6
14. #define relay6 7
15. #define relay7 8
16. #define nivelBajo 22
17.
18. int DS18S20_Pin = 12;
19. OneWire ds(DS18S20_Pin);
20. int sensorValor = 0;
21. int temperatura = 0;
22. int ByteRecibido = 0;
23. float f1 = 0;
24. float t1 = 0;
25. float temp=0;
26.
27.
28. void setup()
29. {
30. Serial.begin(9600);
31. Serial3.begin(9600);
32. Serial2.begin(9600);
33. pinMode(relay1, OUTPUT);
34. pinMode(relay2, OUTPUT);
35. pinMode(relay3, OUTPUT);
36. pinMode(relay4, OUTPUT);
37. pinMode(relay5, OUTPUT);
38. pinMode(relay6, OUTPUT);
39. pinMode(relay7, OUTPUT);
40. pinMode(11,OUTPUT);
41. pinMode(34,OUTPUT);
42. digitalWrite(relay1, HIGH);
43. digitalWrite(relay2, HIGH);
44. digitalWrite(relay3, HIGH);
45. digitalWrite(relay4, HIGH);
46. digitalWrite(relay5, HIGH);
47. digitalWrite(relay6, HIGH);
48. digitalWrite(relay7, HIGH);
49. digitalWrite(11, HIGH);
50. digitalWrite(34, HIGH);
51.
52. inputstring.reserve(10);
```

```

53. sensorstring.reserve(30);
54. sensorstring2.reserve(30);
55.
56. pinMode(nivelBajo, INPUT);
57. delay(10);
58. }
59.
60. void serialEvent() {
61.   inputstring = Serial.readStringUntil(13);
62.   input_string_complete = true;
63. }
64.
65. void serialEvent3() {
66.   sensorstring = Serial3.readStringUntil(13);
67.   sensor_string_complete = true;
68. }
69.
70. void serialEvent2() {
71.   sensorstring2 = Serial2.readStringUntil(13);
72.   sensor_string_complete2 = true;
73. }
74.
75. \
76. void loop()
77. {
78.
79.   if (input_string_complete == true) {
80.     CheckRelay(inputstring);
81.     input_string_complete = false;
82.   } else {
83.     if (sensor_string_complete == true ) {
84.       if (isdigit(sensorstring[0]) == false) {
85.         //Serial.println(sensorstring);
86.       }
87.       else
88.       {
89.         print_EC_data();
90.       }
91.       sensorstring = "";
92.       sensor_string_complete = false;
93.
94.       if (sensor_string_complete2 == true){
95.         print_DO_data();
96.       } else {
97.         sensorstring2 = "";
98.         sensor_string_complete2 = false;
99.
100.        }
101.        temp = getTemp();
102.        Serial.print(temp);
103.        int boya=digitalRead(nivelBajo);
104.        if (boya==HIGH){
105.          Serial.println(",3.14159");
106.        } else {
107.          Serial.println(",0.314159");

```

```

108.     }
109.
110.     delay(1000);
111.     boya=0;
112.     }
113. }
114.
115.
116. float getTemp(){
117.     byte data[12];
118.     byte addr[8];
119.
120.     if ( !ds.search(addr) ) {
121.         ds.reset_search();
122.         return -1000;
123.     }
124.
125.     if ( OneWire::crc8( addr, 7) != addr[7] ) {
126.         Serial.println("CRC is not valid!");
127.         return -1000;
128.     }
129.
130.     if ( addr[0] != 0x10 && addr[0] != 0x28 ) {
131.         Serial.print("Device is not recognized");
132.         return -1000;
133.     }
134.
135.     ds.reset();
136.     ds.select(addr);
137.     ds.write(0x44,1);
138.
139.     byte present = ds.reset();
140.     ds.select(addr);
141.     ds.write(0xBE);
142.
143.
144.     for (int i = 0; i < 9; i++) {
145.         data[i] = ds.read();
146.     }
147.
148.     ds.reset_search();
149.
150.     byte MSB = data[1];
151.     byte LSB = data[0];
152.
153.     float tempRead = ((MSB << 8) | LSB);
154.     float TemperatureSum = tempRead / 16;
155.
156.     return TemperatureSum;
157. }
158. }
159.
160.
161.
162. void CheckRelay(String inputs){

```

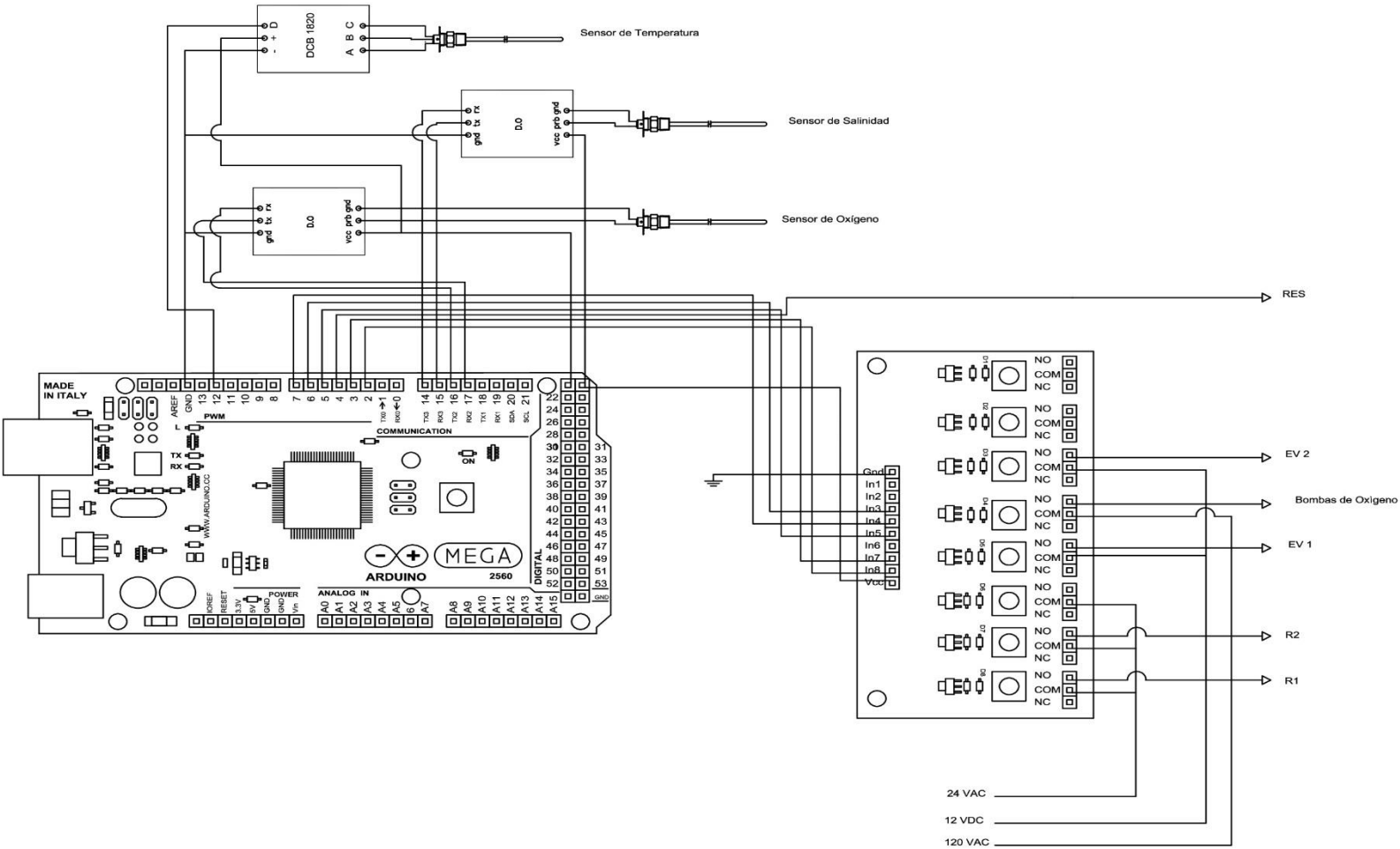
```
163.
164.   delay(10);
165.   if (inputs.length() >0)
166.   {
167.     if(inputs == "A0")
168.     {
169.       digitalWrite(relay1, HIGH);
170.     }
171.     else if(inputs == "A1")
172.     {
173.       digitalWrite(relay1, LOW);
174.     }
175.     else if(inputs == "B0")
176.     {
177.       digitalWrite(relay2, HIGH);
178.     }
179.     else if(inputs == "B1")
180.     {
181.       digitalWrite(relay2, LOW);
182.     }
183.     else if(inputs == "C0")
184.     {
185.       digitalWrite(relay3, HIGH);
186.     }
187.     else if(inputs == "C1")
188.     {
189.       digitalWrite(relay3, LOW);
190.     }
191.     else if(inputs == "D0")
192.     {
193.       digitalWrite(relay4, HIGH);
194.     }
195.     else if(inputs == "D1")
196.     {
197.       digitalWrite(relay4, LOW);
198.     }
199.     else if(inputs == "E0")
200.     {
201.       digitalWrite(relay5, HIGH);
202.     }
203.     else if(inputs == "E1")
204.     {
205.       digitalWrite(relay5, LOW);
206.     }
207.     else if(inputs == "F0")
208.     {
209.       digitalWrite(relay6, HIGH);
210.     }
211.     else if(inputs == "F1")
212.     {
213.       digitalWrite(relay6, LOW);
214.     }
215.     else if(inputs == "G0")
216.     {
217.       digitalWrite(relay7, HIGH);
```

```

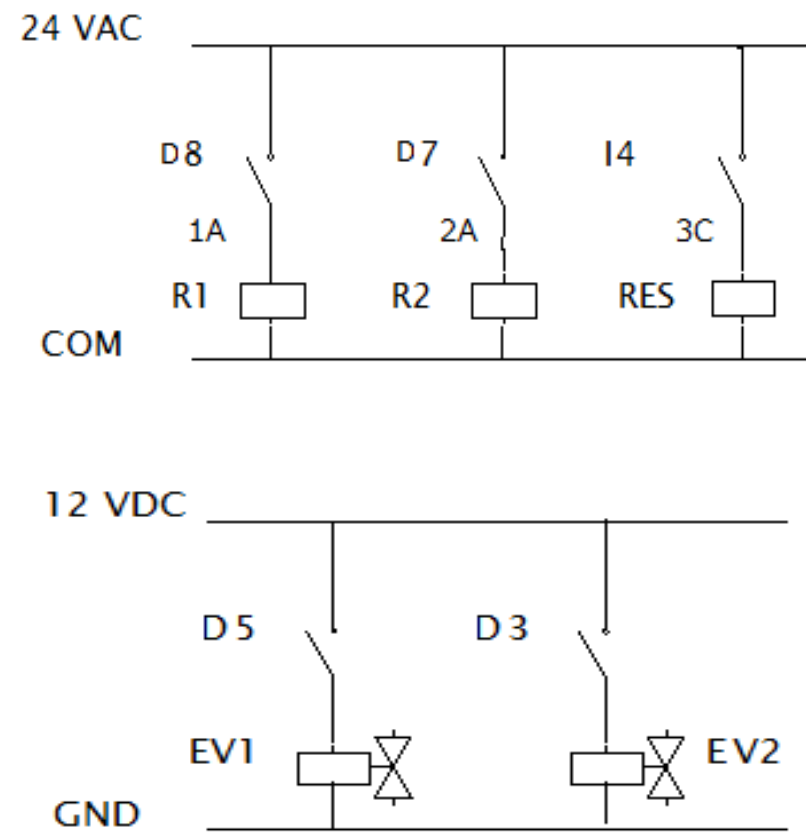
218.     }
219.     else if(inputs == "G1")
220.     {
221.         digitalWrite(relay7, LOW);
222.     }
223.     else{
224.         inputs="";
225.     }
226. }
227. }
228.
229. void print_EC_data(void) {
230.     char sensorstring_array[30];
231.     char *EC;
232.     char *TDS;
233.     char *SAL;
234.     char *GRAV;
235.     float f_ec;
236.     sensorstring.toCharArray(sensorstring_array, 30);
237.     EC = strtok(sensorstring_array, ",");
238.     TDS = strtok(NULL, ",");
239.     SAL = strtok(NULL, ",");
240.     GRAV = strtok(NULL, ",");
241.     Serial.print(EC);
242.     Serial.print(",");
243. }
244.
245. void print_DO_data(void) {
246.     char sensorstring_array2[30];
247.     float DO;
248.     float f_do;
249.     DO = sensorstring2.toFloat();
250.     Serial.print(DO);
251.     Serial.print(",");
252. }

```

ANEXO 4: DIAGRAMA CIRCUITO DE CONTROL DE PLANTA



ANEXO 5: DIAGRAMAS CIRCUITO DE RELES Y ELECTROVÁLVULAS



ANEXO 6: DIAGRAMAS CIRCUITO DE FUERZA DE LA PLANTA

