



UNIVERSIDAD NACIONAL DE CATAMARCA

Facultad de Tecnología y Ciencias Aplicadas

Maximio Victoria N°55 - Tel/Fax: 0383-4435112 Int. 171
San Fernando del Valle de Catamarca (CP 4700)
Argentina

**DISEÑO Y CONSTRUCCIÓN DE UNIDAD DIDÁCTICA
PARA SISTEMAS DE CONTROL CON ENFOQUE EN
PROCESOS DE NIVEL**

TRABAJO FINAL

PARA OBTENER EL TÍTULO DE GRADO DE

INGENIERO ELECTRÓNICO

AUTOR

JULIAN DANIEL BARRIONUEVO

DIRECTOR: ING. MARTÍN HERRERA



Esta obra está bajo una Licencia Creative Commons
Atribución-NoComercial-CompartirIgual 4.0 Internacional

*El amor jamás dejara de existir.
En cambio, las profecías acabarán, las lenguas dejarán de hablarse,
y el conocimiento llegará a su fin.*

1 Corintios 13:8

Dedicado :

A Dios primeramente, por ser mi guía y fortaleza

A mis padres, por supuesto: Rodriguez Adriana y Barrionuevo Daniel

A mis hermanos: Martin, Florencia y Sofia

A mis sobrinos: Lucas y Benjamin

A ellos, a quienes amo profundamente

Agradecimientos

A ustedes papá, mamá y hermanos, no tan solo por el apoyo económico sino por el apoyo moral y tiempo dedicado, por esos momentos en lo que no pude compartir y por esas visitas esporádicas a sus casas.

También para ellos, mis pequeños sobrinos a los cuales deje de hacerlos jugar por momentos y me comprendieron inocentemente.

A mis grandes amigos y compañeros de estudio, Sergio, Félix, Carmen, Josue y Fede. Por los aplausos y por los silencios.

A Erika, por su apoyo incondicional y compañía desde el momento de conocerla. Por todo lo compartido juntos.

A los profesores de mi querida Universidad Nacional de Catamarca por enseñarme no solo lo académico sino lo humano ante cada decisión.

Un infinito agradecimiento, a ti Señor Jesús por haberme alcanzado de las tinieblas y puesto en luz, por hacerme conocer la verdad que me hizo libre.

Resumen

Este documento presenta el diseño, desarrollo y construcción de una unidad didáctica de sistemas de control a fin de que estudiantes de la carrera de Ingeniería Electrónica puedan interpretar de manera conveniente los problemas de control, afianzar conocimientos, familiarizarse con los procesos de nivel de líquidos y realizar prácticas en el laboratorio de Electrónica.

Al tratarse de una unidad didáctica, presenta dos modos de funcionamiento general, libre y automático. El entorno libre acude al uso para posibles practicas en el laboratorio permitiendo el uso de cualquier Microcontrolador a las diferentes partes de la unidad. Mientras que el automático dispone de opciones para que el usuario decida y visualice el comportamiento del sistema elegido.

El desarrollo de esta unidad didáctica se basa en la utilización de la placa de desarrollo electrónico denominada comercialmente Arduino[®] Mega 2560, como placa de adquisición, procesamiento y almacenamiento de datos. Posee una moderna pantalla táctil que controla toda la unidad y permite la visualización del sistema. A su vez, se realizó una aplicación móvil para Android a través del entorno App Inventor[®], mediante la cual es posible la manipulación de la unidad de manera remota.

El enfoque realizado a lo largo de este trabajo fue el de abordar un proceso de diseño integrando componentes electrónicos provistos en el país de tecnología moderna. Para conseguir los objetivos, fue necesario adquirir habilidades en sistemas mecánicos y desarrollar habilidades en torno a la programación de lenguajes de software. Se resalta que tanto los módulos independientes para cada actuador como el PCB (*Printed Circuit Board*) montado sobre la placa Arduino[®] fueron realizados luego de abordar un proceso de diseño de placas PCB de alta generación e interactuar con proveedores y fabricantes nacionales.

Índice general

	Página
Agradecimientos	III
Resumen	IV
Índice de figuras	IX
Índice de tablas	XIII
1. Introducción	1
1.1. Contexto	2
1.1.1. Definición del problema	2
1.2. Objetivos del proyecto	2
1.2.1. Objetivo General	2
1.2.2. Objetivos Específicos	3
1.3. Motivación	3
2. Metodología de Trabajo y Antecedentes Técnicos	5
2.1. Revisiones Preliminares	5
2.1.1. Diseño de la Estructura y Hardware	5
2.1.2. Diseño e Implementación del Software	6
2.1.3. Verificación de Desempeño de la Unidad	6
2.2. Antecedentes	7
2.2.1. Situación actual de Enseñanza-Aprendizaje en el Mundo	7
2.2.2. Antecedentes Técnicos	8
2.2.3. Componentes de una Planta Piloto	9
2.3. Ideas de Cierre	10

3. Marco Teórico	12
3.1. Sistemas de Control	12
3.1.1. Breve reseña histórica de la evolución de las Teorías de Control	12
3.1.2. Términos básicos en los sistemas de control	15
3.1.3. Comparación entre sistemas de control realimentado y de lazo abierto	16
3.1.4. Tipos de sistemas de control realimentados	18
3.1.5. Sistemas lineales vs. No lineales	19
3.1.6. Sistemas invariantes vs. variantes en el tiempo	20
3.1.7. Tipos de Señales	21
3.1.8. Sistemas de control en tiempo continuo	23
3.1.9. Sistemas de control en tiempo discreto	24
3.1.10. Criterios para diseñar un sistema de control	24
3.2. Modelado Matemático de sistemas de control	25
3.2.1. Función de transferencia	26
3.2.2. Diagramas de bloques	27
3.2.3. Función de transferencia en lazo abierto	27
3.2.4. Función de transferencia en lazo cerrado	28
3.3. Tipos de controladores	29
3.3.1. Controladores On-Off	29
3.3.2. Controlador de acción Proporcional	30
3.3.3. Controlador de acción Integral	31
3.3.4. Controlador de acción Derivativo	31
3.3.5. Controlador Proporcional-Integral-Derivativo	32
3.4. Sistemas de nivel de líquido	32
3.4.1. Resistencia y capacitancia.	33
3.4.2. Análisis de sistemas de nivel de liquido	36
3.4.3. Análisis de sistemas de nivel en 2 tanques	38
3.5. Relé de estado sólido	39
3.5.1. Acoplamiento	39
3.5.2. Funcionamiento	40
3.5.3. Ventajas sobre relés electromecánicos	40
3.6. Control de velocidad de un motor AC	40
3.6.1. Triac	41
3.6.2. Modo de funcionamiento	41
3.7. Plataformas de desarrollo	43

3.7.1.	Arduino [®] Mega 2560	43
3.7.2.	Matlab [®]	43
3.7.3.	Pantalla Táctil Nextion 7" HMI	45
3.7.4.	Overleaf [®]	45
4.	Memoria Descriptiva	47
4.1.	Diseño y Construcción - Hardware	47
4.1.1.	Diseño de la unidad didáctica	47
4.1.2.	Modelo y simulación de dimensiones de los tanques	48
4.1.3.	Modelado estático del sistema de nivel	48
4.1.4.	Modelado dinámico del sistema de nivel	51
4.1.5.	Simulación de las funciones de transferencia	57
4.2.	Sensor de Presión	59
4.2.1.	Característica del sensor	60
4.2.2.	Acondicionamiento de la señal	61
4.2.3.	Acondicionamiento de señal para conversores de 0 a 10 V	64
4.2.4.	Función de Transferencia del Sensor	64
4.3.	Sensor de ultrasonido HC-SR04	67
4.3.1.	Características del sensor	67
4.3.2.	Función de transferencia	68
4.4.	Electroválvula	69
4.4.1.	Diseño y Construcción	69
4.4.2.	Características del motor	71
4.4.3.	Etapas de potencia	72
4.4.4.	Diagrama de bloques	73
4.4.5.	Función de transferencia	74
4.5.	Electrobomba Periférica	80
4.5.1.	Características del motor	80
4.5.2.	Etapas de potencia y acondicionamiento de señal	81
4.6.	Fuente de Alimentación	81
4.6.1.	Características Fuente:	81
4.6.2.	Diseño PCB	83
4.6.3.	Planos de Masa	83
4.7.	Sistema de Control de Nivel	84
4.7.1.	Diagrama de bloques	84
4.7.2.	Diseño de control	85

4.7.3.	Diseño del controlador PID analógico	87
4.7.4.	Digitalización del Sistema	94
4.7.5.	Discretización del controlador PID	96
4.7.6.	Ecuaciones en Diferencias	98
4.8.	Consideraciones de Diseño - Software	101
4.8.1.	Programación del Microcontrolador	101
4.8.2.	Pantalla Nextion	110
4.8.3.	App Inventor- Aplicación para Android	112
4.9.	Funcionamiento de la Unidad	114
4.9.1.	Modo Automático	114
4.9.1.1.	Control 1º Orden	115
4.9.1.2.	Control 2º Orden	116
4.9.2.	Modo Manual	117
4.9.2.1.	Modo Libre	118
4.9.2.2.	Modo Comandos	120
4.9.2.3.	Modo Bluetooth	121
5.	Trabajos Prácticos	122
5.1.	Caracterización del sistema	122
5.2.	Análisis de la respuesta transitoria y error en estado estacionario	126
5.3.	Análisis de estabilidad	128
5.4.	Diseño de controlador tipo PI	130
5.5.	Sistema de control de 2º orden	131
5.6.	Diseño de controlador tipo PID	133
5.7.	Discretización de sistemas	133
5.8.	Ecuaciones en Diferencias	135
6.	Resultados y Análisis	137
6.1.	Pruebas de desempeño	137
6.1.1.	Electroválvula	137
6.1.2.	Bomba de agua	139
6.1.3.	Modo Libre	141
6.1.4.	Modo Bluetooth	142
6.2.	Sistemas de control	143
6.2.1.	Sistema de primer orden	143
6.2.2.	Sistema de segundo orden	149

7. Conclusiones y Trabajos futuros	155
7.1. Conclusiones	155
7.2. Trabajos futuros	155
A. Datasheet de Componentes	157
A.1. Atmega 2560	158
A.2. Distribución de pines Atmega 2560	159
A.3. MOC 3041	160
A.4. L293D	161
A.5. TIP 3055	162
A.6. Sensor MPX5010DP	163
A.7. Motor Stepper	164
B. Diseño de PCB	165
B.1. Especificaciones PCB	169
B.2. Lista de Componentes Utilizados	169
C. Código de Programación	172
D. Recomendaciones	193
Bibliografía	195

Índice de figuras

2.1. Planta piloto industrial para el control de caudal y nivel.	10
3.1. Pendulo de Watt [5]	13
3.2. Diagrama de Bode de un filtro paso bajo Butterworth de primer orden. [6]	13
3.3. Sistema de control Multivariable. [7]	14
3.4. Sistema de control realimentado. [8]	17
3.5. Sistema de control en lazo abierto. [9]	17
3.6. No linealidades. [10]	20
3.7. Misil Tomahawk de ataque terrestre norteamericano. [11]	21
3.8. Tipos de señales. [12]	22
3.9. Representación gráfica de una señal continua $x(t)$. [13]	22
3.10. Representación gráfica de una señal discreta $x[n]$. [13]	23
3.11. Diagrama de un sistema típico de D.C en lazo cerrado. [14]	23
3.12. Diagrama de bloques de un sistema de control de datos muestreados. [15]	24
3.13. Elementos de un diagrama de bloques. [16]	27
3.14. Sistema en lazo cerrado. [17]	28
3.15. Diagrama de bloques de un controlador on-off. [18]	30
3.16. Diagrama de bloques de un controlador PID. [20]	32
3.17. Sistema de nivel de un tanque. [21]	34
3.18. Sistema de nivel de dos tanque. [22]	38
3.19. Relé de estado solido provisto por el fabricante. [23]	39
3.20. Estructura, símbolo y característica . [24]	41
3.21. Circuito de control de fase (potencia) . [25]	42
3.22. Gráfica de ondas de un triac . [26]	42
3.23. Arduino [®] Mega 2560. [27]	43
3.24. Logo Matlab [®] . [28]	44
3.25. Usos de Matlab [®] . [28]	44
3.26. Pantalla Nextion NX8048T070 . [30]	45

3.27. Editor de la pantalla Nextion. [30]	46
3.28. Editor de LaTeX online. [31]	46
4.1. Estructura de la unidad. [32]	48
4.2. Tanque abierto . [32]	49
4.3. Válvula de control . [33]	50
4.4. Esquema básico de un sistema de primer orden . [32]	51
4.5. Esquema básico de un sistema de segundo orden . [32]	54
4.6. Respuesta a lazo abierto al modelo de un tanque . [32]	57
4.7. Respuesta lazo abierto al modelo de dos tanques . [32]	58
4.8. Sensor de presión diferencial Mpx5010DP . [34]	59
4.9. Esquemático sensor de presión. [34]	60
4.10. Ruido amplificado en sensor de presión. [41]	62
4.11. Filtro pasa bajo. [41]	62
4.12. Simulación filtro pasa bajo. [32]	63
4.13. Respuesta en frecuencia de filtro pasa bajo . [32]	63
4.14. Filtro para fuente de alimentación. [41]	63
4.15. Acondicionamiento señal para A/D. [32]	64
4.16. Salida vs presión diferencial. [34]	65
4.17. Sensor ultrasónico HC- SR04. [35]	67
4.18. Variables en un sistema de engranajes. [32]	69
4.19. Motor acoplado al sistema de engranajes. [32]	70
4.20. Eje del motor acoplado a válvula. [32]	71
4.21. Conexiones de pines. [38]	72
4.22. Simulación L293D. [32]	73
4.23. Diagrama de bloques de la electroválvula . [32]	73
4.24. Modelo con un polo real más integrador. [32]	76
4.25. Modelo con dos polos reales. [32]	76
4.26. Modelo con dos polos reales más integrador. [32]	77
4.27. Modelo con un par de polos complejos conjugados. [32]	77
4.28. Caracterización electroválvula. [32]	79
4.29. Herramienta Curve Fitting Tool . [32]	79
4.30. Bomba Marca Plumita . [39]	80
4.31. Diseño esquemático Fuente Regulada. [32]	82
4.32. Fuente diseñada en Kicad, vista en 2D. [32]	83
4.33. Fuente diseñada en Kicad, vista en 3D. [32]	84

4.34. Diagrama de bloques del sistema. [32]	85
4.35. Diagrama de bloques de la planta. [32]	86
4.36. Lugar de raíces del sistema más controlador. [32]	89
4.37. Respuesta al sistema de primer orden. [32]	89
4.38. Diferencias en respuesta al sistema de primer orden. [32]	90
4.39. Diseño control PI y respuesta del sistema. [32]	91
4.40. Lugar de raíces del sistema de segundo orden más controlador PID. [32]	93
4.41. Respuesta al sistema de segundo orden. [32]	93
4.42. Diseño control PID y respuesta del sistema. [32]	94
4.43. Comando para discretizar con retenedor de orden cero. [32]	96
4.44. Comando para discretizar con retenedor de orden cero. [32]	97
4.45. Diagrama de bloques Discreto. [32]	98
4.46. Comparación de T_d	101
4.47. Diagrama de estados programación. [32]	102
4.48. Programación de la pantalla Táctil. [32]	111
4.49. Instrucciones de programación. [32]	111
4.50. Programación pantalla. [32]	112
4.51. Entorno de desarrollo MIT App Inventor. [32]	112
4.52. Programación en bloques de la aplicación. [32]	113
4.53. Display control de primer orden. [32]	115
4.54. Display control de segundo orden. [32]	116
4.55. Display Modo Manual. [32]	118
4.56. Display Modo Libre. [32]	119
4.57. Tablero Didáctico. [32]	120
4.58. Display Modo Comandos. [32]	120
4.59. Display Modo Bluetooth. [32]	121
5.1. Modelo sistema de primer orden.	123
5.2. Diagrama de bloques de la planta.	126
5.3. Especificaciones de respuesta transitoria.	127
5.4. Diagrama de bloques del sistema.	127
5.5. Diagrama de bloques del sistema con compensador K	128
5.6. Diagrama de bloques del sistema con compensador $\frac{K}{s}$	128
5.7. Sistema en lazo cerrado. [17]	129
5.8. Diagrama de bloques del sistema con compensador K	129
5.9. Diagrama de bloques del sistema con compensador $\frac{K}{s}$	130

5.10. Esquema básico de un sistema de segundo orden . [32]	131
5.11. Diagrama de bloques Discreto. [32]	135
6.1. Caracterización del caudal de electroválvula. [32]	138
6.2. Caracterización del triac de bomba periférica. [32]	140
6.3. Entrada externa para el uso de bomba . [32]	140
6.4. Modo libre realizando pruebas de desempeño. [32]	141
6.5. Microcontrolador secundario EDU-CIAA. [32]	142
6.6. Respuesta a lazo abierto. [32]	144
6.7. Respuesta a lazo cerrado. [32]	145
6.8. Respuesta a lazo cerrado con PID. [32]	145
6.9. Respuesta del controlador PID teórico. [32]	146
6.10. Periodo de oscilación vista desde la pantalla. [32]	147
6.11. Regla de sintonía de Ziegler-Nichols. [19]	147
6.12. Fragmento del código implementando la ecuación del controlador. [32]	148
6.13. Respuesta real del sistema con el controlador PID. [32]	149
6.14. Respuesta ante una entrada escalón al sistema de 2º orden. [32] . . .	150
6.15. Respuesta ante una entrada escalón al sistema de 2º orden compensado con un PI. [32]	151
6.16. Respuesta real del sistema con el controlador PI teórico. [32]	151
6.17. Respuesta oscilatoria real del sistema. [32]	152
6.18. Respuesta real con parámetros de Ziegler-Nichols. [32]	154
A.1. Arduino Mega. [36]	158
A.2. Pin Out Arduino Mega. [36]	159
A.3. MOC 3041. [23]	160
A.4. Doble puente H. [38]	161
A.5. Transistor de Potencia. [40]	162
A.6. Sensor de Presión. [34]	163
A.7. Motor paso a paso. [37]	164
B.1. Capa de fabricación del PCB. [32]	165

Índice de tablas

4.1. Caracterización del sensor MPX5010.	66
4.2. Caracterización del sensor HC-SR04.	68
4.3. Caracterización de electroválvula.	75
6.1. Caracterización del caudal de la válvula.	138
6.2. Caracterización del caudal de Bomba periférica.	139
6.3. Tiempo de respuesta de los actuadores.	142
6.4. Tiempo de muestreo de los datos.	143
6.5. Tiempo de muestreo del periodo critico.	153
B.1. Lista de Componentes Placa Principal.	170
B.2. Lista de Componentes Fuente de alimentación.	171
B.3. Lista de Componentes Tablero de modo libre.	171

Capítulo 1

Introducción

En este documento se aborda como tema principal el diseño, desarrollo, construcción e implementación de una unidad didáctica de sistemas de control con enfoque en procesos de nivel.

Cabe mencionar que se escogió este tipo de proceso para implementar el control, ya que se trata de un sistema con constantes de tiempo relativamente grandes lo que lo hace muy útil para ensayos, a pequeña escala en laboratorio.

La unidad montada en el laboratorio de Electrónica representa mejoras en el aprendizaje de los sistemas de control. Su bloque funcional de control principal esta basado en la placa Arduino[®] Mega 2560. Los módulos electrónicos de cada actuador del sistema permite una interacción independiente con cada uno de ellos, siendo posible el uso de cualquier Microcontrolador.

El diseño presentado abarca la manipulación intuitiva a través de una pantalla moderna Touch NEXTION de 7", como a su vez, la obtención del nivel es a través de una señal dada por un sensor, en donde se realiza su acondicionamiento mediante técnicas de instrumentación electrónica, su posterior digitalización, almacenamiento e incluso transmisión mediante el protocolo de comunicación IEEE 802.15.1 Bluetooth.

Por otra parte se realizo una Aplicación para dispositivos Android[®], la cual permite el manejo y visualización de la unidad de manera portátil.

1.1. Contexto

1.1.1. Definición del problema

En la actualidad se observa un déficit en el aprendizaje de ciertos contenidos, ya sea por falta de material o falta de implementación de mejoras en la enseñanza, lo cual se mejoraría notablemente a partir de contar con el equipamiento adecuado, además de ayudar a la visualización y la manipulación de conceptos a partir de su aplicación en equipos físicos.

Si bien existen unidades didácticas en instrumentación de control, y sistemas de control de ciertas variables, su precio es elevado para aficionados y estudiantes de la carrera.

El enfoque al control de nivel se debe a que existen procesos que requieren mantener constante el nivel de tanques para evitar ciertos problemas, como daño en la etapa de succión de bombas propulsoras, evitar reboses de tanques o bien sea para asegurar mezclas perfectas. Esto hace que se recurra monitorear el nivel del tanque para controlar el flujo de entrada o de salida al tanque, asegurando de esta forma, que el nivel permanezca siempre un valor deseado.

Como antecedente, se participó en una entrevista de trabajo con Minera El Altiplano, donde comentaron de la existencia de piletas de gran tamaño donde realizan la extracción de litio y el necesario control del nivel de líquidos para obtener mezclas homogéneas y no presentar pérdidas durante la extracción. Lo que observamos que es una temática de gran uso en la industria.

1.2. Objetivos del proyecto

1.2.1. Objetivo General

Creación, desarrollo, construcción e implementación de la unidad didáctica con enfoque al sistema de control de nivel en tanque, que permita ser usado para realizar prácticas en el laboratorio de electrónica, donde pueda conectarse cualquier Microcontrolador a las diferentes partes de la unidad. Verificando la factibilidad de elaborar una unidad didáctica de fácil manejo, haciendo uso de los conocimientos adquiridos durante mi formación como Ingeniero Electrónico.

1.2.2. Objetivos Específicos

- Implementación de una unidad didáctica en el Laboratorio de Electrónica para que los estudiantes realicen practicas de fácil manejo sin necesidad de trasladarse a un parque industrial.
- Diseñar una unidad didáctica de uso intuitivo, capaz de mostrar el funcionamiento de los sistemas de control y sus conceptos.
- Caracterizar matemáticamente la unidad didáctica para el desarrollo de prácticos.
- Utilizar componentes de alta calidad que se encuentren fácilmente en la provincia.
- Realizar un proceso de fabricación del PCB para obtener una calidad típica de industria y desarrollar comunicación con proveedores tanto locales como nacionales.

1.3. Motivación

A lo largo de una carrera universitaria como así también en los primeros niveles educativos existen contenidos difíciles de interpretar, por lo que el aprendizaje resulta poco eficaz. Para contrarrestar estos resultados se realizan acciones que mejoran la relación enseñanza-aprendizaje, una de las estrategias adoptadas son las llamadas unidades didácticas.

Podemos decir que una unidad didáctica es **un elemento** en el proceso de enseñanza-aprendizaje, el cual se convierte en el eje integrador del tema a tratar.

Esta forma de organizar conocimientos y experiencias debe considerar la diversidad de elementos que contextualizan el proceso (nivel de desarrollo del alumno, medio sociocultural y familiar, proyecto curricular, recursos disponibles) para regular la práctica de los contenidos, seleccionar los objetivos básicos que pretende conseguir, las pautas metodológicas con las que trabajará, las experiencias de enseñanza-aprendizaje necesarios para perfeccionar dicho proceso. [42]

En este escenario se hace útil la posibilidad de implementar unidades físicas en el ámbito académico para mejoras en el aprendizaje.

Desde la aparición de la electrónica surgió un gran incremento en maquinarias, herramientas y dispositivos usados en la industria que realizan un trabajo más eficiente, por este motivo resulta interesante capacitarse constantemente en estos sistemas y adquirir habilidades industriales profesionales.

De lo anterior resulta importante poder contar con los conceptos claros en estos sistemas y poder realizar practicas en equipamientos a escala de lo que se encuentra en una planta industrial.

Hoy en día con la electrónica en gran avance es posible desarrollar una unidad didáctica de fácil manejo e intuitiva que contenga la aplicación de conceptos referidos a los sistemas de control, combinado con el uso de dispositivos portátiles, como son los SmartPhones y Tablets y los sistemas gráficos computarizados para mostrar el estado del proceso, resulta en una solución tecnológica apropiada moderna, la cual es viable y son tendencias en universidades nacionales e internacionales con un gran numero de estudiantes.

Desde el punto de vista académico realizado en el cursado de la carrera de Ingeniería Electrónica, la principal motivación en realizar este proyecto es que reúne prácticamente todas las áreas, de las cuales se pueden citar materias como, Señales y Sistemas, Teoría de control, Electrónica II, Electrónica III, Sistemas Microprogramables, Instrumentación Electrónica y Formulación y Evaluación de proyectos.

Desde el comienzo, el proyecto presenta una complejidad tratable con los conocimientos adquiridos durante la formación como ingeniero, a su vez se acompaña con investigación en ámbitos externos a la Ingeniería Electrónica, como es la mecánica en soldaduras eléctricas, conexión de caños de agua y niveles de turbidez, uso de varios software de diseño y un nuevo lenguaje de programación en LaTeX¹.

¹Es un sistema de composición de textos, orientado a la creación de documentos escritos que presentan una alta calidad tipográfica.

Capítulo 2

Metodología de Trabajo y Antecedentes Técnicos

2.1. Revisiones Preliminares

Lo primero que se realizó para el desarrollo del proyecto fue la lectura exhaustiva sobre el tema abordado en una revisión bibliográfica de artículos científicos, libros, manuales, publicaciones, trabajos finales y reportes técnicos.

Por otra parte, se realizaron consultas a profesionales y profesores especialistas en el área del trabajo, como así también se examinaron diferentes unidades existentes en el mercado tanto nacional como internacional con el fin de replicar algunas novedades interesantes y modernas que estas poseen.

Finalmente fueron evaluados diferentes aspectos para el desarrollo de la unidad, como son el costo, disponibilidad de materiales mecánicos y electrónicos, junto con las especificaciones de los circuitos integrados, y la disponibilidad de software de desarrollos OpenSource.

2.1.1. Diseño de la Estructura y Hardware

Para el diseño de las partes mecánicas de la unidad, se analizaron las necesidades para que resulte con terminación industrial pero a su vez de fácil manipulación, se observaron especificaciones y criterios de selección para las partes claves del sistema de control, como la electroválvula diseñada para este proyecto. Luego se realizó una clasificación y elección de los diseños planteados en el anteproyecto de este trabajo, y se optó por aquel que alcanzaba un equilibrio técnico-económico para su realización.

Como dispositivo central del proyecto se escogió la tarjeta de desarrollo Open-Source Arduino Mega, la cual alberga un microcontrolador de la familia Atmel Semiconductors[®], a su vez se usaron Arduino[®] y App Inventor[®] como plataformas de desarrollo. Simultáneamente se agregó una pantalla Touch Nextion[®] para el diseño de una interfaz gráfica HMI (*Human Machine Interface*), y se trabajó con circuitos integrados específicos para el diseño de las etapas de potencia y procesamiento de señales.

Para el diseño y fabricación de la placa central de la unidad, se realizaron los diagramas esquemáticos utilizando un moderno y potente software libre de diseño de PCB (*Printed Circuit Board*), como lo es KiCad[®]. Luego se adquirieron los componentes con diferentes proveedores tanto locales como nacionales.

2.1.2. Diseño e Implementación del Software

Como primer paso para el diseño e implementación del software, se realizaron diversas librerías programadas en el IDE (*Integrated Development Environment*) de Arduino[®] haciendo uso de programación orientada a objetos¹.

El siguiente paso fue diseñar una interfaz gráfica HMI en Nextion[®] Editor, la cual realiza la comunicación entre el usuario y el microcontrolador permitiendo interactuar de manera sencilla e intuitiva con todo el sistema.

Posteriormente se diseñó una App² en el entorno de desarrollo App Inventor de OpenSource, la cual nos proporciona un manejo de manera remota a cada parte de la unidad.

2.1.3. Verificación de Desempeño de la Unidad

Una vez realizada la estructura mecánica, fabricado el hardware, por consiguiente depurado el software, y luego de haber realizado las diferentes interfaces gráficas, se realizaron pruebas del desempeño del sistema.

¹Tratan a los programas como conjuntos de objetos que se ayudan entre ellos para realizar acciones. Entendiendo como objeto a entidades que contienen datos.

²Aplicación para dispositivos portátiles.

Las pruebas desarrolladas fueron exclusivamente con el propósito de que el sistema global cumpla con los requerimientos y objetivos planteados, verificando el correcto funcionamiento de todas las partes involucradas.

2.2. Antecedentes

2.2.1. Situación actual de Enseñanza-Aprendizaje en el Mundo

Actualmente existen diversos informes internacionales que nos brindan información sobre como se encuentra el nivel de enseñanza y aprendizaje alrededor del mundo y nos dan las pautas para analizar más profundamente el problema a solucionar que presenta este trabajo.

El reporte “NMC Horizon Report: 2017 Higher Education” [1], refleja los resultados de una investigación realizada conjuntamente entre The New Media Consortium¹ y The EDUCAUSE Learning Initiative², con el apoyo de 78 especialistas internacionales, permitiendo analizar el futuro inmediato de la educación superior y el impacto que tendrán las tecnologías emergentes.

El resumen ejecutivo del informe destaca puntos que capturan los grandes temas del cambio educativo:

- Se necesitan habilidades reales para reforzar la empleabilidad y el desarrollo del lugar de trabajo.
- A pesar del aumento de tecnologías y materiales de aprendizaje en línea, el acceso sigue siendo desigual.
- Se necesitan realizar procesos para evaluar las habilidades a nivel personal.
- El dominio del ámbito digital es más que sólo entender cómo usar la tecnología.
- Los ecosistemas de aprendizaje deben ser lo suficientemente ágiles como para apoyar las prácticas del futuro.
- La educación superior es una incubadora para desarrollar computadoras más intuitivas.

¹Organización dedicada a la exploración y al uso de nuevos medios y nuevas tecnología para el aprendizaje.

²Comunidad dedicada al avance del aprendizaje a través de las nuevas tecnologías.

- El aprendizaje permanente es el alma o la sangre vital de la educación superior.

El Banco Mundial advierte sobre una crisis del aprendizaje en la educación a nivel mundial, en el “Informe sobre el desarrollo mundial 2018: Aprender para hacer realidad la promesa de la educación ”, se sostiene que, sin aprendizaje, la educación no podrá ser el factor determinante para poner fin a la pobreza extrema, generar oportunidades y promover la prosperidad compartida. [43]

En el caso de los jóvenes, la educación, cuando funciona como es debido, fomenta el empleo, incrementa los ingresos, mejora la salud y reduce la pobreza. A nivel social, estimula la innovación, fortalece las instituciones y promueve la cohesión social. Pero estos beneficios dependen del aprendizaje, y la escolarización **sin aprendizaje es una oportunidad desaprovechada**. Más aún, es una gran injusticia: los niños con los que la sociedad está más en deuda son aquellos que más necesitan de una buena educación para prosperar en la vida.

Dados a estos informes resultan diferentes medidas concretas para ayudar a resolver la deficiencia en el aprendizaje, una de ellas, a la que apunta este trabajo, es desarrollar equipamientos adecuados para afianzar los conocimientos, destacando el acceso para todos los estudiantes a este tipo de instrumental.

2.2.2. Antecedentes Técnicos

Desde tiempo empresas multinacionales trabajan conjuntamente con universidades, con el objetivo de que el estudiante tenga los conocimientos teóricos-prácticos, relacionados con las últimas tecnologías presentes en las industrias modernas, y así adquirir habilidades previas al desarrollo en campo.

A efecto se crean convenios entre ambas partes para el desarrollo de laboratorios y unidades didácticas, con la finalidad de impulsar, intercambiar y generar espacios de desarrollo e investigación técnico-científico entre ambas organizaciones [44].

A modo de ilustrar el funcionamiento y desarrollo de una unidad didáctica de sistema de control, se cita un desarrollo realizado por la Universidad Nueva Esparta en 2009, Caracas, Venezuela. [45]

Este trabajo muestra un procedimiento para la creación de un moderno laboratorio de sistemas y comunicaciones industriales con la finalidad de mejorar el proceso en-

señanza-aprendizaje. Describe la actualización tecnológica que se agrega a una planta piloto industrial equipada con instrumentación convencional.

La modernización permite dotar a la planta con instrumentos inteligentes y herramientas tecnológicas para la enseñanza de conceptos tales como, sistemas de control, buses de campo, OPC¹, SCADA, control y supervisión a través de la web, comunicación inalámbrica, etc.

2.2.3. Componentes de una Planta Piloto

Para simplificar la descripción podemos dividir en dos grandes grupos, **Hardware** (Adquisición de datos y actuadores) y **Software** (Control y supervisión de la planta)

Diseño

Para una adecuada interpretación de contenidos teóricos sobre sistemas de control, automatización y programación, se requiere una buena adaptación del diseño y construcción de la unidad equipada con actuadores y aplicaciones en una escala reducida, que dan una idea del equipamiento existente en las instalaciones industriales, las cuales encontrará el estudiante en el campo laboral.

La planta es un sistema multivariable que consta de un tanque, un reservorio y dos líneas de bombeo con válvulas reguladoras para manipular el flujo a la entrada y a la salida del tanque. Posee un medidor de flujo a la entrada del tanque y un medidor de nivel lo que permite realizar el control de estas variables.

Toda la instrumentación de la planta: los actuadores (válvulas), y los medidores (flujo y nivel) son completamente neumáticos. Posee un par de conversores presión/-corriente (P/I), para transformar las señales de presión provenientes de los medidores en señales de corriente estándar (4-20mA), y un par de conversores corriente/presión (I/P) para convertir las señales de corriente (4-20mA) en señales de presión necesarias para los actuadores neumáticos de las válvulas.

¹El OPC (OLE for Process Control) es un estándar de comunicación en el campo del control y supervisión de procesos industriales, basado en una tecnología Microsoft, que ofrece una interfaz común para comunicación que permite que componentes de software individuales interactúen y compartan datos. [46]

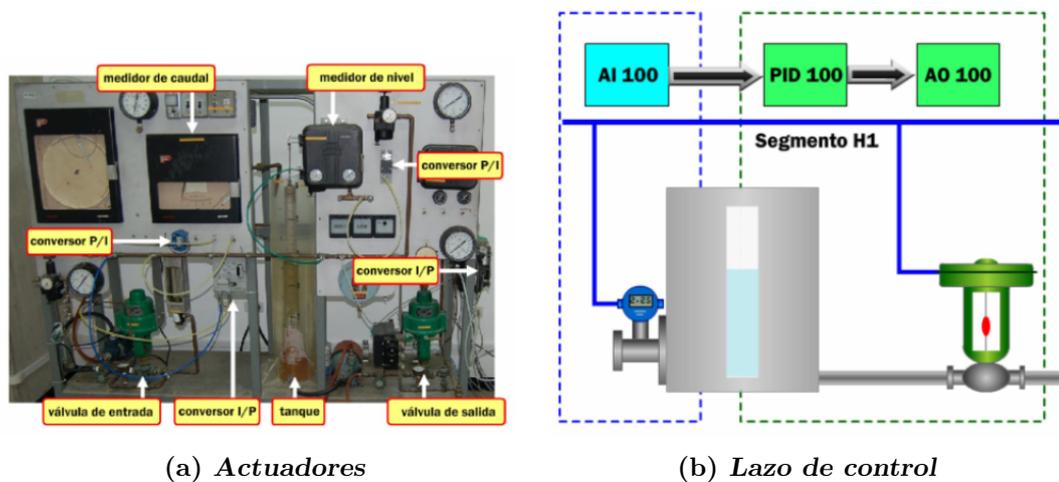


Figura 2.1: Planta piloto industrial para el control de caudal y nivel.

Control y supervisión de la planta

Un elemento indispensable en el laboratorio de sistemas y comunicaciones industriales es, una aplicación SCADA/HMI (interfaz humano máquina).

En este sentido, la universidad que realizó este trabajo cuenta con un Software (pago) de Información y Automatización Industrial Wonderware (2009). Este programa es líder mundial en soluciones de aplicaciones industriales. Dispone de las herramientas de software necesarias para desarrollar, en un tiempo relativamente breve, aplicaciones de automatización que incluyen la visualización y la comunicación vía OPC con los dispositivos de campo.

También tiene incluido un componente para servidores web que permite el control y la visualización remota vía Internet.

2.3. Ideas de Cierre

De acuerdo al estudio y análisis de la revisión histórica que se realizó, y examinado las diferentes propuestas desarrolladas en universidades, la elaboración de este proyecto busca seguir con el incremento en el aprendizaje de conceptos sobre los sistemas de control, programación y comunicaciones.

Se observa que el control y supervisión es un punto vital en una planta industrial, por lo cual la presentación y forma de visualización de los parámetros, debe cumplir

con la expectativa de los usuarios, de que sea lo más clara posible y de fácil manejo. También se menciona que las funcionalidades de comunicación remota, son un valor agregado solo presente en los equipos de más alta gama y por lo tanto no se encuentra presente en la mayoría de las unidades del mercado.

Cabe aclarar que el presente trabajo no es una copia del expuesto líneas arriba, si no que se llevó a cabo a partir de analizar ideas sobre ese desarrollo y la probabilidad de realizar una unidad didáctica sin convenios con terceros, usando exclusivamente herramientas locales y nacionales.

Capítulo 3

Marco Teórico

3.1. Sistemas de Control

En una simple definición, podemos decir que un sistema de control es un conjunto de actuadores que buscan satisfacer ciertas especificaciones de diseño.

El control automático ha desempeñado un importante papel en el avance de la ciencia e ingeniería. Se puede observar su parte vital en los sistemas de vehículos espaciales, en los sistemas robóticos y en procesos modernos industriales.

Podemos citar 3 teorías de control que se utilizan habitualmente, teoría **clásica** (también conocida como convencional), teoría de control **moderna** y teoría de control **robusta**.

3.1.1. Breve reseña histórica de la evolución de las Teorías de Control

El primer trabajo significativo en control automático fue el regulador de velocidad centrífugo de James Watt ¹ para el control de la velocidad de una máquina de vapor, en el Siglo XVIII.

En 1922, Minorsky ² trabajó en controladores automáticos para el guiado de embarcaciones, y mostró que la estabilidad puede determinarse a partir de las ecuaciones diferenciales que describen el sistema.

¹Fue un ingeniero mecánico e inventor escocés. Las mejoras que realizó en la máquina de Newcomen dieron lugar a la conocida como máquina de vapor de agua, que resultaría fundamental en el desarrollo de la primera Revolución Industrial, tanto en el Reino Unido como en el resto del mundo.

²Nicolas Minorsky (nacido Nikolai Fyodorovich Minorsky, 23 de septiembre de 1885, Korcheva, Imperio ruso - 31 de julio de 1970, Italia) fue un matemático de la teoría del control, ingeniero y científico aplicado ruso afincado en los Estados Unidos [2]

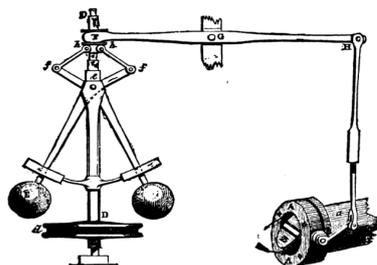


Figura 3.1: Pendulo de Watt [5]

En 1932, Nyquist diseñó un procedimiento relativamente simple para determinar la estabilidad de sistemas en lazo cerrado¹, a partir de la respuesta en lazo abierto a entradas sinusoidales en estado estacionario.

En 1934, Hazen, quien introdujo el término servomecanismos para los sistemas de control de posición, analizó el diseño de los servomecanismos con relé, capaces de seguir con precisión una entrada cambiante.

Durante la década de los cuarenta, los métodos de la respuesta en frecuencia (especialmente los diagramas de Bode) hicieron posible que los ingenieros diseñaran sistemas de control lineales en lazo cerrado que cumplieran los requisitos de comportamiento.

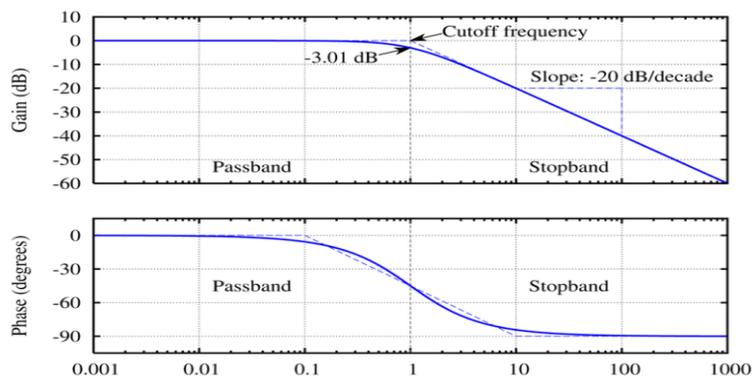


Figura 3.2: Diagrama de Bode de un filtro paso bajo Butterworth de primer orden. [6]

En los años cuarenta y cincuenta muchos sistemas de control industrial utilizaban controladores PID² para el control de la presión, de la temperatura, posición de motores, etc.

¹El diagrama de Nyquist es una representación paramétrica de una función de transferencia, se utiliza en control automático y procesamiento de señales.

²Proporcional Integral Derivativo.

A comienzos de la década de los cuarenta fueron Ziegler y Nichols quienes establecieron reglas para sintonizar controladores PID, las denominadas reglas de sintonía de Ziegler-Nichols.

A finales de los años cuarenta y principios de los cincuenta, se desarrolló por completo el método del lugar de las raíces¹ propuesto por Evans.

Los métodos de respuesta en frecuencia y del lugar de las raíces, que forman el núcleo de la teoría de control clásica, conducen a sistemas estables que satisfacen un conjunto más o menos arbitrario de requisitos de comportamiento (aplicaciones en las cuales no se requiere una precisión fina). En general, estos sistemas son aceptables pero no óptimos.

Desde el final de la década de los cincuenta, el énfasis en los problemas de diseño de control, se ha desplazado del diseño de uno de los posibles sistemas que funciona adecuadamente al diseño de un sistema óptimo respecto de algún criterio.

Se puede establecer que las plantas modernas con muchas entradas y salidas se vuelven más y más complejas, y la descripción de un sistema de control moderno requiere una gran cantidad de ecuaciones.

La teoría de control clásica, que trata de los sistemas con una entrada y una salida, pierde su potencialidad cuando se trabaja con sistemas con entradas y salidas múltiples.

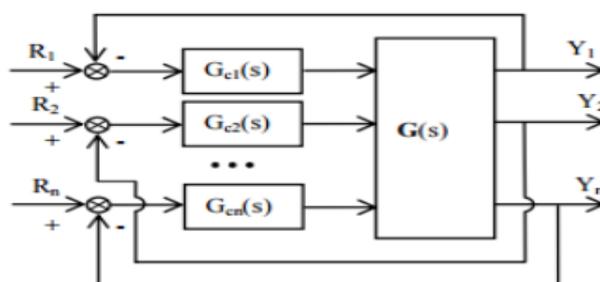


Figura 3.3: Sistema de control Multivariable. [7]

Hacia 1960, debido a la disponibilidad de las computadoras digitales fue posible el análisis en el dominio del tiempo de sistemas complejos. La teoría de control moderna, basada en el análisis en el dominio del tiempo y la síntesis a partir de variables de

¹Es el lugar geométrico de los polos y ceros de una función de transferencia a medida que se varía la ganancia del sistema en un determinado intervalo.

estados¹, se ha desarrollado para manejar la creciente complejidad de las plantas modernas y los requisitos cada vez más exigentes sobre precisión, peso y coste en aplicaciones militares, espaciales e industriales.

Durante los años comprendidos entre 1960 y 1980, se investigó a fondo el control óptimo tanto de sistemas determinísticos como estocásticos, así como el control adaptativo y con aprendizaje de sistemas complejos.

Desde la década de los ochenta hasta la de los noventa, los avances en la teoría de control moderna se centraron en el control robusto y temas relacionados.

La teoría de control moderna simplificó el diseño de los sistemas de control porque se basa en un modelo del sistema real que se quiere controlar. Sin embargo, la estabilidad del sistema depende del error entre el sistema real y su modelo. Esto significa que cuando el controlador diseñado basado en un modelo se aplica al sistema real, éste puede tener diferencias y no ser estable.

Para evitar esta situación, se diseña el sistema de control definiendo en primer lugar, el rango de posibles errores y después diseñando el controlador de forma que, si el error del sistema está en dicho rango, el sistema de control diseñado permanezca estable. El método de diseño basado en este principio se denomina teoría de control robusto². Esta teoría incorpora tanto la aproximación de respuesta en frecuencia como la del dominio temporal.

En este proyecto se realizó el estudio con la teoría clásica de control, puesto que el análisis de este sistema de control es univariable y con fines didácticos a la teoría de control vista en el cursado.

3.1.2. Términos básicos en los sistemas de control

- **Variable controlada y señal de control o variable manipulada:** La variable controlada es la cantidad o condición que se mide y controla. La señal de control o variable manipulada es la cantidad o condición que el controlador modifica para afectar el valor de la variable controlada. Normalmente, la variable controlada es la salida del sistema.

¹En ingeniería de control, es una representación de espacios de estados un modelo matemático de un sistema físico descrito mediante un conjunto de entradas, salidas y variables de estado relacionadas por ecuaciones diferenciales de infinito orden que se combinan en una ecuación diferencial matricial de primer orden.

²Este tipo de control es matemáticamente más complejo que los anteriores.

- **Plantas:** Una planta puede ser una parte de un equipo, tal vez un conjunto de los elementos de una máquina que funcionan juntos, y cuyo objetivo es efectuar una operación particular (como un dispositivo mecánico, un horno de calefacción, un reactor químico o una nave espacial).
- **Procesos:** Un proceso es una operación o un desarrollo natural y/o artificial progresivamente continuo. El proceso artificial es aquel que consta de una serie de acciones o movimientos controlados, sistemáticamente dirigidos hacia un resultado o propósito determinado.
- **Sistema:** Un sistema es una combinación de componentes que actúan juntos y realizan un objetivo determinado.
- **Perturbaciones:** Una perturbación es una señal que tiende a afectar negativamente el valor de la salida de un sistema. Si la perturbación se genera dentro del sistema se denomina interna, mientras que una perturbación externa se genera fuera del sistema y actúa como una entrada.
- **Control realimentado:** El control realimentado se refiere a una operación que, en presencia de perturbaciones¹, tiende a reducir la diferencia entre la salida de un sistema y alguna entrada de referencia, y lo realiza tomando en cuenta esta diferencia.

3.1.3. Comparación entre sistemas de control realimentado y de lazo abierto

- **Sistemas de control realimentado o en lazo cerrado:**

Es aquel sistema que mantiene una **relación** entre la salida y la entrada de referencia, comparándolas y usando la diferencia como medio para asistir al control.

En un sistema de control en lazo cerrado, como se observa en la figura 3.4, se alimenta al controlador con la señal de error de actuación, que es la diferencia entre la señal de entrada y la señal de realimentación (que puede ser la propia señal de salida o una función de la señal de salida y sus derivadas y/o integrales), con el fin de reducir el error y llevar la salida del sistema al valor deseado.

¹Se especifican con este término las perturbaciones impredecibles, ya que las perturbaciones predecibles o conocidas siempre pueden compensarse dentro del sistema.

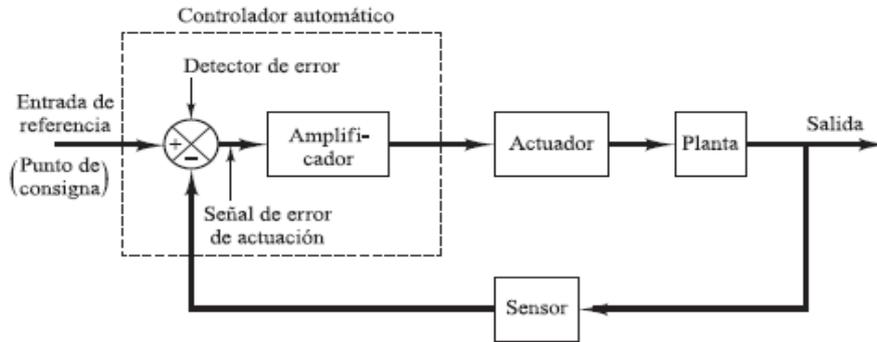


Figura 3.4: Sistema de control realimentado. [8]

El término control en lazo cerrado siempre implica el uso de una acción de control realimentado para reducir el error del sistema.

▪ Sistemas de control en lazo abierto:

Son aquellos sistemas en los cuales la salida **no tiene efecto** sobre la acción de control.

En este tipo de sistema de control no se mide la salida ni se realimenta para compararla con la entrada. Un ejemplo práctico es un lavarropa. El remojo, el lavado y el centrifugado en el lavarropa operan con una base de tiempo.

La máquina no mide la señal de salida, que sería, que tan limpia se encuentra la ropa, por ejemplo.

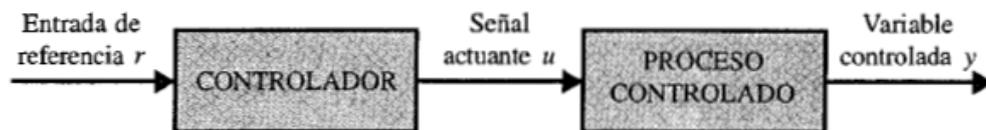


Figura 3.5: Sistema de control en lazo abierto. [9]

En cualquier sistema de control en lazo abierto, la salida no se compara con la entrada de referencia. Así, a cada entrada de referencia le corresponde una condición de operación fija; como resultado de ello, la precisión del sistema depende de la calibración. Ante la presencia de perturbaciones, un sistema de control en lazo abierto no realiza la tarea deseada. En la práctica, el control en lazo abierto sólo se usa si se

conoce la relación entre la entrada y la salida y si no hay perturbaciones internas ni externas tan drásticas.

Ventajas en los sistemas de control en lazo abierto:

- Construcción simple y facilidad de mantenimiento.
- A ser de fácil mantenimiento resulta menos costoso.
- Convenientes cuando la salida es difícil de medir o cuando medir la salida de manera precisa no es económicamente viable.

Desventajas en los sistemas de control en lazo abierto:

- Las perturbaciones y los cambios en la calibración originan errores, y la salida puede ser diferente de lo que se desea.
- Para mantener la calidad requerida en la salida, es necesaria una constante calibración.

Comentarios acerca de los sistemas de control realimentados

Este trabajo presenta un control del tipo realimentado, ya que tiene ciertas ventajas antes perturbaciones. Haciendo uso del lazo realimentado nos permite tener una mejora en la respuesta del sistema no tan solo a perturbaciones externas sino también a variaciones internas en los parámetros de los componentes del sistema.

Otro de los motivos del porque se escogió este tipo de control, es que resulta posible usar componentes pocos precisos y baratos para alcanzar un control adecuado en ciertas plantas.

Desde un punto de vista de la estabilidad, es un problema estimulante para aplicar los conceptos de teoría de control, pero se debe tener en cuenta que puede conducir a corregir en exceso errores que producen oscilaciones de amplitud constante o cambiante afectando la estabilidad del sistema.

3.1.4. Tipos de sistemas de control realimentados

Existen diferentes criterios a la hora de clasificar los sistemas de control, podemos mencionar algunos de acuerdo al:

Método de análisis y diseño:

- Lineales.
- No lineales.
- Variantes en el tiempo.
- Invariantes en el tiempo.

Tipo de señales usadas:

- Tiempo continuo.
- Tiempo discreto.

También podemos citar algunas clasificaciones de acuerdo al propósito principal del sistema, como por ejemplo, sistema de control de nivel, sistema de control de posición, sistema de control de velocidad, sistema de control de temperatura, entre otros.

En general existen varias formas de clasificar un sistema de control de acuerdo con alguna función especial del sistema¹. Sin embargo es importante tener presente algunas de las mencionadas anteriormente al momento del análisis y diseño de un sistema de control.

3.1.5. Sistemas lineales vs. No lineales

Los sistemas físicos que se encuentran en la práctica son **no** lineales en cierto grado, por lo que podemos concluir que los sistemas lineales no existen. Sin embargo estos sistemas presentan ciertos intervalos en los cuales sus componentes tienen un comportamiento lineal (se aplica el principio de superposición²), por lo tanto se pueden aproximar a un sistema lineal y nos ayudan a simplificar el análisis y diseño.

Cuando las magnitudes de las señales se extienden más allá del intervalo lineal, el sistema debe tratarse como no lineal. En estas situaciones, se puede encontrar una

¹También se puede clasificar atendiendo a la interacción del sistema con la señal de entrada, por ejemplo, con memoria o sin memoria, causal o no causal, estable o inestable, determinista o no determinista. Para más información diríjase [3]

²Este hecho tan importante ha sido la causa de que se hayan desarrollado tanto las técnicas de análisis de sistemas lineales. Este principio establece que la respuesta de un sistema a una suma de señales de entrada es la suma de la respuesta del sistema a cada señal de entrada por separado. [3]

solución para una excitación y condiciones iniciales dadas, bien analíticamente o con la ayuda de algún software.

Existen diversas no linealidades como por ejemplo, en los amplificadores usados en los sistemas de control exhiben un efecto de saturación cuando la señal de entrada es muy grande, a su vez el campo magnético de un motor también presenta saturación. Otras no linealidades presentes son, el juego entre dos engranajes, zona muerta de un potenciómetro, fuerza de fricción no lineal, etc.

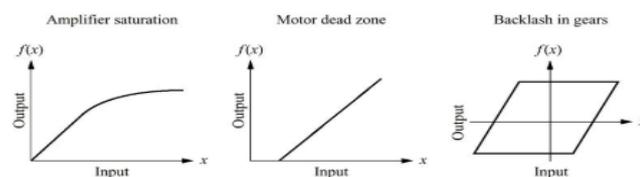


Figura 3.6: No linealidades. [10]

A menudo, las características no lineales son introducidas de forma intencional en los sistemas de control para mejorar su desempeño o tener un control más efectivo. Por ejemplo en los sistemas de generación de calor —calefacción— la histéresis (similar a la no linealidad ON-OFF) es la diferencia de temperaturas establecida automáticamente entre el encendido y el apagado o el apagado y el encendido del sistema de calefacción.

Frecuentemente se necesita determinar el comportamiento en las cercanías de ese punto, por lo que se aproxima a un sistema lineal, esta técnica se denomina **linealización**. Algunos ejemplos son el uso de la Función Descriptiva, Plano de fase, análisis de pequeña señal para transistores, Aproximaciones por Serie de Taylor, entre otras formas de linealizar.

3.1.6. Sistemas invariantes vs. variantes en el tiempo

Se dice que un sistema es invariante en el tiempo si un desplazamiento temporal de la señal de entrada causa un desplazamiento temporal idéntico en la salida del sistema. Matemáticamente esto es:

Si $y(t)$ es la salida que corresponde a la entrada $x(t)$, un sistema invariante en el tiempo producirá una salida $y(t - t_0)$ para una entrada $x(t - t_0)$.

Desde el punto de vista de los parámetros del sistema de control, cuando estos son estacionarios con respecto al tiempo durante la operación del sistema, el sistema es invariante.

Por su parte, los sistemas variantes¹ en el tiempo, en la mayoría de los sistemas físicos incluyen elementos que derivan o varían con el tiempo. Por ejemplo, la resistencia de la bobina de un motor eléctrico varía de acuerdo a la corriente y a la temperatura que alcanza. Otro ejemplo, es el sistema de control de un misil guiado en el cual la masa del misil decrece a medida que el combustible a bordo se consume durante el vuelo.

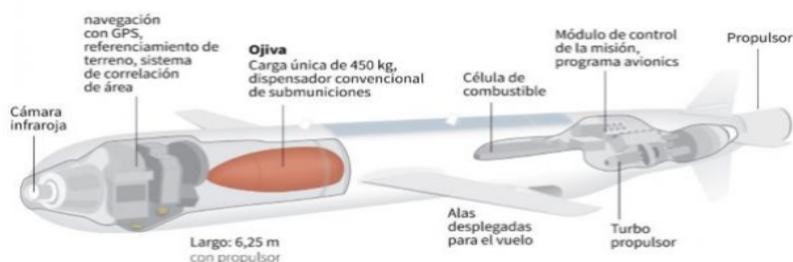


Figura 3.7: Misil Tomahawk de ataque terrestre norteamericano. [11]

3.1.7. Tipos de Señales

Podemos decir que señal, son todas aquellas magnitudes físicas o variables detectables mediante las cuales permiten transmitir mensajes o información. Existen una gran variedad de señales que son de importancia práctica en la descripción de fenómenos físicos.

Las señales eléctricas componen el tipo de señales que se pueden medir con mayor facilidad y permiten representarlas de manera más sencilla. Es por eso que, se prefiere convertir magnitudes físicas como la temperatura, la presión, la humedad, la velocidad, en señales de tensión o corriente que varían con el tiempo.

Matemáticamente, las señales se representan como funciones de una o más variables independientes. Por ejemplo, las señales de tensión y corriente que varían a lo

¹Aunque un sistema variante sin no linealidades es aun un sistema lineal, el análisis y diseño de esta clase de sistemas son muchos mas complejos que los invariantes en el tiempo.

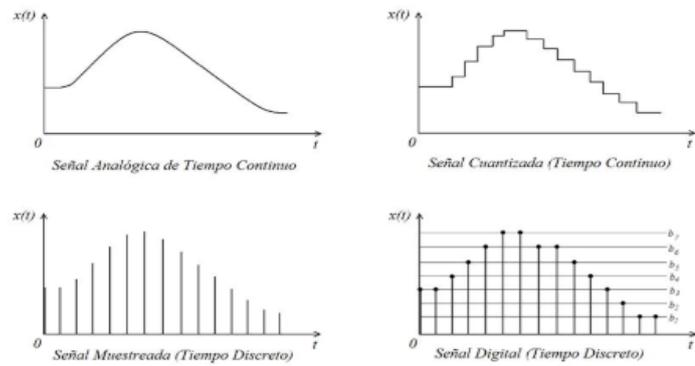


Figura 3.8: Tipos de señales. [12]

largo del tiempo son funciones solo del tiempo, en cambio la intensidad de un campo eléctrico puede tomarse como una función de dos variables espaciales (tiempo y espacio).

Señales continuas

Son aquellas que se definen para una sucesión continua de valores en un determinado intervalo de tiempo (variable independiente).

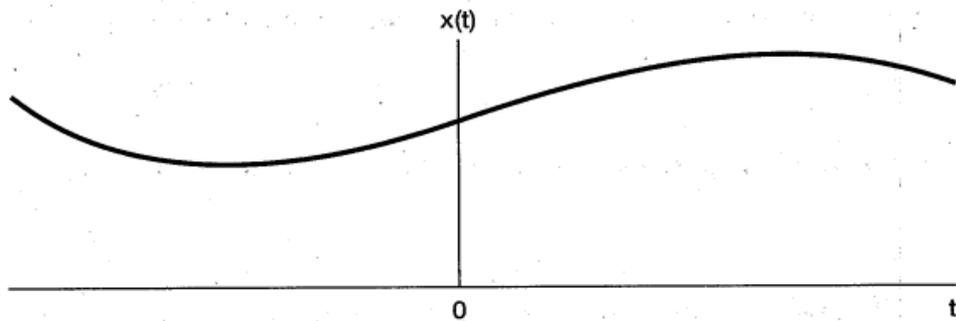


Figura 3.9: Representación gráfica de una señal continua $x(t)$. [13]

Señales discretas

Las señales discretas difieren de las anteriores en que solo están definidas en tiempos discretos, por lo tanto, la variable independiente toma solamente un conjunto de valores discretos.

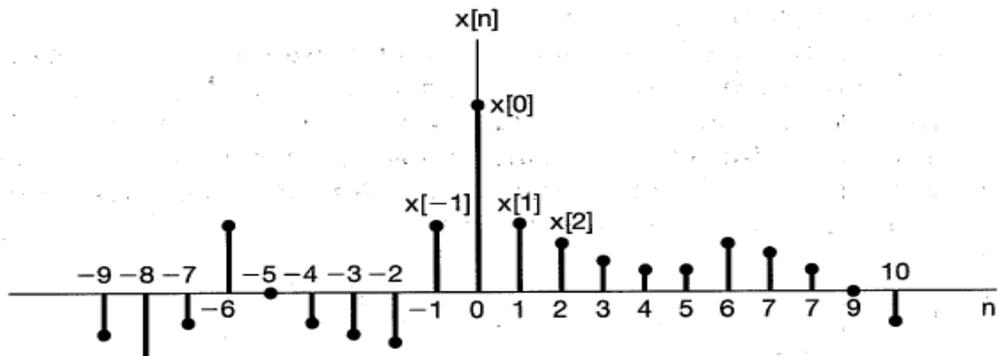


Figura 3.10: Representación gráfica de una señal discreta $x[n]$. [13]

3.1.8. Sistemas de control en tiempo continuo

Un sistema continuo es aquel, en el cual las señales de entrada en tiempo continuo se transforman en señales de salida en tiempo continuo.

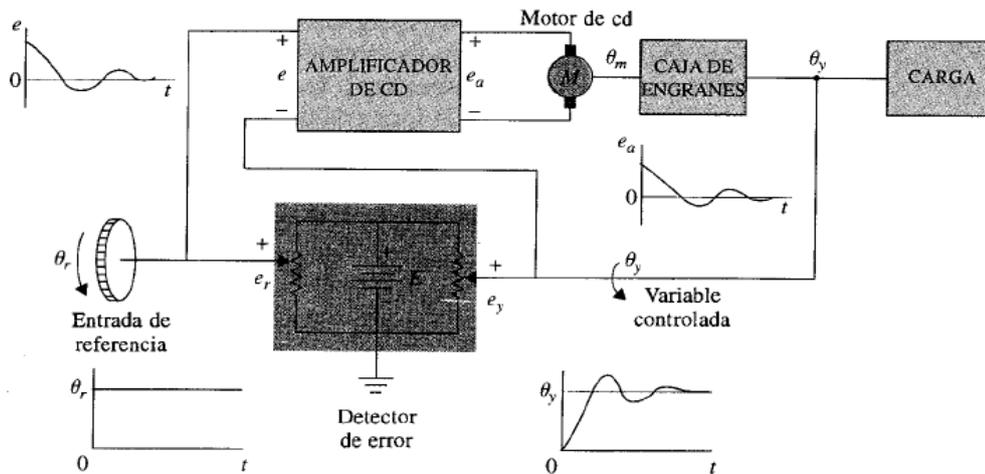


Figura 3.11: Diagrama de un sistema típico de D.C en lazo cerrado. [14]

3.1.9. Sistemas de control en tiempo discreto

Un sistema discreto se diferencia de un sistema continuo en que, las señales en uno o más puntos del sistema pueden presentarse en forma de pulsos o en un código digital que utilizan los procesadores.

Los sistemas en tiempo discreto se pueden dividir en **sistemas de control de datos muestreados y sistemas de control digital**.

Los sistemas de control de datos muestreados se refieren a una clase más general en que las señales están en la forma de pulsos de datos, es decir, el sistema recibe los datos o información en forma intermitente en instantes específicos.

Un sistema de control digital hace referencia al uso de una computadora o Microcontrolador digital, de tal forma que las señales están presentes en la forma de un código digital, por ejemplo, código binario.



Figura 3.12: Diagrama de bloques de un sistema de control de datos muestreados. [15]

3.1.10. Criterios para diseñar un sistema de control

Especificaciones de comportamiento

El primer paso a la hora de diseñar, es saber con certeza que deseamos que haga nuestro sistema, para ello se establecen ciertas especificaciones de comportamiento. Estas especificaciones pueden venir dadas como requisitos en la respuesta tanto temporal como estacionaria, por ejemplo en tiempo de asentamiento, máxima sobre elongación, error estacionario, etc.

Es importante recalcar que la parte del diseño de un sistema de control más importante, es la precisión con la que se definen las especificaciones de comportamiento con el fin de obtener un sistema de control óptimo para el planteado.

Cuando se diseña un sistema se establecen valores precisos y algunos cualitativos, esto es para realizar modificaciones en las especificaciones durante el proceso del diseño ya que, pueden haber casos en los cuales las especificaciones planteadas no ocurran o dirijan a un sistema muy costoso.

Compensación del sistema

Una de las posibles soluciones para llevar el sistema a un comportamiento adecuado, es incrementar su ganancia, aunque esto puede mejorar el comportamiento, en estado estacionario puede producir una estabilidad inadecuada incluso llegar a la inestabilidad total del sistema.

Una solución más acertada sería agregar componentes adicionales al diseño. Este elemento agregado para satisfacer las especificaciones se denomina **compensador** y tiene como función principal modificar el comportamiento deficiente del sistema.

Procedimientos de diseño

Se necesita partir de un modelo matemático que represente el sistema real, para ello se utiliza una aproximación de prueba y error haciendo uso de un software de calculo numérico, como Matlab, que simplifique los cálculos.

Una vez obtenido el modelo matemático, se ajustan los parámetros del compensador y se prueba la estabilidad del sistema en lazo abierto. Si se asegura la estabilidad, se debe cerrar el lazo y probar el comportamiento del sistema.

Tal vez el modelo representado difiera del sistema real, ya sea por efectos de ciertos componentes, falta de linealidad, parámetros externos, como la temperatura, etc. Por lo tanto, el diseñador debe cambiar el modelo hasta que el sistema cumpla con las especificaciones de comportamiento y, a su vez, que sea económico y viable de llevarlo a cabo. Esta etapa es la que necesita más tiempo en el diseño.

3.2. Modelado Matemático de sistemas de control

El modelado matemático de los procesos controlados, es decir, la representación en un conjunto de **ecuaciones** de la dinámica del sistema, es un asunto importante en el análisis y diseño de sistemas de control.

No existe un modelo único para un sistema. Los modelos matemáticos pueden adaptarse a muchas formas de acuerdo a diferentes análisis. Por ejemplo, para analizar la respuesta transitoria con una entrada y una salida invariante en el tiempo, la representación en función de transferencia suele ser más conveniente que la representación en espacios de estados que se utilizan en problemas de control óptimo.

En el modelado de un sistema se debe tener una relación de compromiso entre, que tan preciso es el sistema y que tan simple resulta ser su modelado. Para ello, generalmente se omiten no linealidades presentes en los sistemas físicos, siempre y cuando los efectos de estas sean pequeños sobre el sistema.

3.2.1. Función de transferencia

Una función de transferencia de un sistema descrito mediante ecuaciones diferenciales lineales e invariantes en el tiempo se define como el cociente entre la transformada de Laplace de la salida (a veces llamada función de respuesta) y la transformada de Laplace de la entrada (función de excitación). Es un método matemático para expresar la ecuación diferencial que relaciona la variable de salida con la variable de entrada.

$$G(s) = \frac{Y(s)}{X(s)} \quad (3.1)$$

- $G(s)$: Función de transferencia.
- $Y(s)$: Transformada de Laplace de la salida.
- $X(s)$: Transformada de Laplace de la entrada.

El concepto de función de transferencia está ligado a los sistemas que fueron descritos mediante ecuaciones lineales invariantes en el tiempo. La función de transferencia es una propiedad de un sistema, es independiente de la señal de entrada y nos proporciona una descripción de las características dinámicas del sistema.

Conocer la función de transferencia de un sistema, nos ayuda a comprender la naturaleza del sistema aplicando diferentes tipos de entradas y viendo que salidas se obtienen a partir de estas.

3.2.2. Diagramas de bloques

Siguiendo con la presentación de los contenidos teóricos vistos para este proyecto, se encuentran los diagramas de bloques. Estos diagramas son **una representación gráfica de las funciones de cada componente y del flujo de las señales** presentes en el sistema.

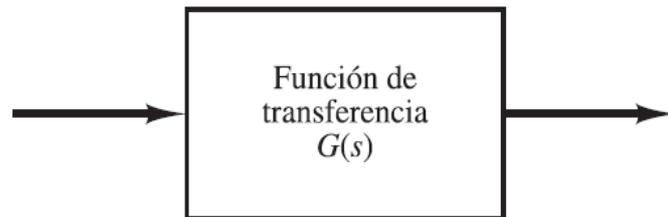


Figura 3.13: Elementos de un diagrama de bloques. [16]

En un diagrama de bloques todas las variables del sistema se unen, unas con otras mediante bloques funcionales. Este bloque, como muestra la figura 3.13, es un símbolo usado para representar la operación matemática que actúa sobre la señal de entrada para obtener una señal de salida. Estas señales son representadas a través de flechas. La punta de la flecha indica si es entrada o salida.

Las ventajas de la representación mediante diagramas de bloques de un sistema convergen en que, es más fácil formar el diagrama de bloques general de todo el sistema con sólo conectar los bloques de los componentes de acuerdo con el flujo de señales y que es posible, evaluar la contribución de cada componente al desempeño general del sistema. Es decir, nos da una idea más clara del **funcionamiento** del sistema pero no nos brinda información acerca del sistema físico mismo.

3.2.3. Función de transferencia en lazo abierto

Observando la figura 3.14, el cociente entre la señal de realimentación $B(s)$ y la señal de error $E(s)$, recibe el nombre de función de transferencia en lazo abierto.

$$\frac{B(s)}{E(s)} = G(s)H(s) \quad (3.2)$$

El cociente entre la salida $C(s)$ y la señal de error $E(s)$ se denomina función de transferencia de la trayectoria directa, por lo que:

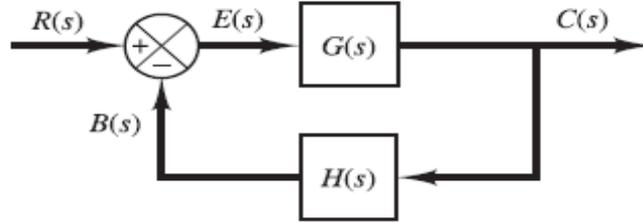


Figura 3.14: Sistema en lazo cerrado. [17]

$$\frac{C(s)}{E(s)} = G(s) \quad (3.3)$$

Si la función de transferencia de la trayectoria de realimentación $H(s)$ es la unidad, la función de transferencia en lazo abierto y la función de transferencia de la trayectoria directa son iguales.

3.2.4. Función de transferencia en lazo cerrado

Considerando el sistema que aparece en la figura 3.14, la función de transferencia se obtiene como :

$$C(s) = E(s)G(s) \quad (3.4)$$

$$E(s) = R(s) - B(s) \quad (3.5)$$

$$E(s) = R(s) - C(s)H(s) \quad (3.6)$$

Reemplazando la ecuación (3.6) en (3.4) se tiene:

$$C(s) = G(s)[R(s) - H(s)C(s)]$$

$$C(s) = G(s)R(s) - G(s)H(s)C(s)$$

$$C(s) + G(s)H(s)C(s) = G(s)R(s)$$

$$C(s)[1 + G(s)H(s)] = G(s)R(s)$$

$$\frac{C(s)}{R(s)} = \frac{G(s)}{1 + G(s)H(s)} \quad (3.7)$$

Esta función de transferencia relaciona la salida con respecto a la entrada, y recibe el nombre de función de transferencia en lazo cerrado. Esta función asocia la dinámica del sistema en lazo cerrado con las trayectorias directas y de realimentación.

De la ec. (3.7), la salida $C(s)$ se obtiene como:

$$C(s) = \frac{G(s)}{1 + G(s)H(s)}R(s) \quad (3.8)$$

Resumiendo, la salida del sistema depende tanto de la función de transferencia del sistema como del tipo de entrada aplicada.

3.3. Tipos de controladores

Otro de los conceptos aplicados a este proyecto es el controlador automático. Este controlador es una pieza sumamente importante a la hora de diseñar ya que se encarga del buen funcionamiento del sistema.

La principal función que presentan los controladores es la producir una señal de control sobre un actuador de manera que esa señal, que se obtiene de comparar el valor medido en la salida de la planta con la señal de referencia (valor deseado), permita reducir la desviación a cero o a un valor pequeño considerable.

3.3.1. Controladores On-Off

Este tipo de controlador es el más usado, debido a que solo actúa en dos posiciones fijas, que por lo general, son simplemente apagado y encendido. Es relativamente simple y barato de realizar en numerosas aplicaciones tanto industriales y como domesticas.

Este tipo de controlador fue usado en una primera etapa del proyecto. El actuador que se utilizó fue una válvula electromagnética de 220 voltios A.C operada por solenoides, de las que se utilizan en lavarropas automáticos.

Matemáticamente, podemos expresar este tipo de controlador como:

$u(t)$: Salida del controlador en el tiempo.

$e(t)$: Señal de error.

$$u(t) = U_1 \text{ para } e(t) > 0$$

$$u(t) = U_2 \text{ para } e(t) < 0$$

Donde U_1 y U_2 son constantes. En este controlador, la señal de salida permanece en un valor sea máximo o mínimo, dependiendo del valor que toma la señal de error¹. .

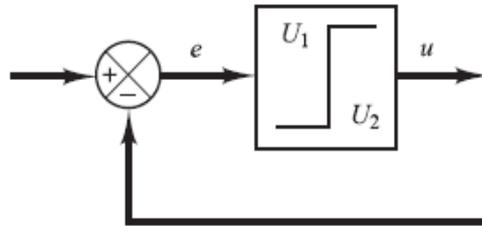


Figura 3.15: Diagrama de bloques de un controlador on-off. [18]

3.3.2. Controlador de acción Proporcional

El controlador de acción proporcional, si bien es más complejo que el controlador de on-off, presenta mejoras en el tiempo de respuesta del sistema, ya que permite modular la señal de salida del actuador de control.

Este tipo de controlador utilizan la mayoría de los controladores que regulan la velocidad de un automóvil. El controlador proporcional es mas sencillo que un sistema de control proporcional integral derivativo (PID).

Matemáticamente podemos expresar al controlador como:

$$u(t) = K_p e(t)$$

La salida del controlador $u(t)$ es proporcional a la señal de error, que es la diferencia entre el punto deseado y la variable de proceso. Cualquiera que sea el mecanismo real y la forma de la potencia de operación, el controlador proporcional es, en esencia, un amplificador con una ganancia ajustable.

¹Para mas información sobre este controlador puede dirigirse al siguiente enlace: <https://drive.google.com/open?id=1aY-Pmah960OZHWIUcAzXvmp3SITgfady>.

3.3.3. Controlador de acción Integral

El modo de control integral tiene como propósito disminuir y eliminar el error en estado estacionario, provocado por perturbaciones exteriores, el cual no puede ser corregido por el control proporcional. Actúa integrando el error, entre la salida y el valor deseado, en el tiempo y sumando a la acción proporcional.

Matemáticamente, esto es

$$\frac{du(t)}{dt} = K_i e(t)$$

La ecuación anterior indica que el valor de la salida de este controlador cambia a una razón proporcional a la señal del error.

$$u(t) = K_i \int_0^t e(t) dt$$

La función de transferencia para este controlador es:

$$\frac{U(s)}{E(s)} = \frac{K_1}{s}$$

A modo de ilustración se puede considerar este tipo de controlador como, mover la electroválvula que controla el nivel del tanque a una **velocidad** proporcional al error.

3.3.4. Controlador de acción Derivativo

Siguiendo con la presentación de los tipos de controladores, se encuentra el controlador de acción Derivativo. Este tipo se expone cuando el error del sistema no es constante, es decir cuando se produce un cambio en el error absoluto del sistema.

Entonces la función principal de este controlador es la de evitar que el error se incremente, para eso debe mantener el error al mínimo corrigiéndolo proporcionalmente a medida aparezca.

Matemáticamente esto es:

$$u(t) = T_d \frac{de(t)}{dt}$$

donde :

- T_d : Tiempo derivativo.

- $u(t)$: Salida del controlador.
- $e(t)$: Error del sistema.

La acción de control derivativa a veces se denomina control de velocidad, T_d es el intervalo de tiempo en el que la acción de velocidad se adelanta al efecto de acción proporcional. Cuando el tiempo de acción derivada es grande, hay inestabilidad en el proceso ahora cuando el tiempo de acción derivada es pequeño la variable oscila demasiado con relación al valor deseado. El tiempo óptimo de acción derivativa es el que retorna la variable al punto de consigna con las mínimas oscilaciones.

3.3.5. Controlador Proporcional-Integral-Derivativo

Resulta interesante resaltar que este tipo de controlador es el más usado en aplicaciones industriales. Este controlador reúne las ventajas de los 3 controladores vistos previamente.

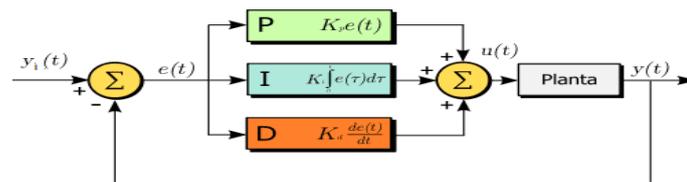


Figura 3.16: Diagrama de bloques de un controlador PID. [20]

La ecuación que representa a este controlador es:

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt + K_p T_d \frac{de(t)}{dt}$$

Su función de transferencia se expresa como:

$$\frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i s} + T_d s \right)$$

3.4. Sistemas de nivel de líquido

La base central en el desarrollo de este proyecto es el control de nivel de un líquido. Este tema fue revisado en la bibliografía de Ingeniería de Control Moderna - Ogatta 5ta edición. Aquí se presenta un resumen de esas líneas.

Cuando se habla de sistemas que implican flujo de líquidos, es necesario definir 2 conceptos con respecto al tipo de flujo que se presenta.

Flujo Laminar: Si el número de Reynolds¹ es menor que unos 2000 estamos en presencia de un flujo laminar. En este caso el flujo es estacionario y se comporta como si estuviese formado por laminas delgadas. Una de las ventajas de los sistemas que contienen un flujo laminar es que se pueden representar mediante ecuaciones diferenciales lineales.

Flujo Turbulento: Si el número de Reynolds es mayor a 4000, el flujo se considera como turbulento. Es caracterizado por un movimiento desordenado, no estacionario y tridimensional. Los sistemas que presentan este tipo de flujo se representan por ecuaciones diferenciales no lineales. Aunque, si limitamos la región de operación, podemos linealizar las ecuaciones por diferentes métodos.

A menudo los procesos industriales implican un flujo de líquidos a través de tubos, válvulas de control y tanques interconectados, por lo que resulta un tipo de flujo turbulento. Este tipo de flujo es el que se encuentra en este proyecto y el cual fue analizado para su desarrollo.

3.4.1. Resistencia y capacitancia.

Estos dos conceptos nos permiten entender fácilmente el análisis de los sistemas de nivel, ya que nos brinda una cierta analogía con los conceptos de electrónica.

Consideremos el flujo que pasa a través de un tubo que conecta a dos tanques. La resistencia R para ese flujo se define como, **el cambio en la diferencia de nivel (la diferencia entre el nivel de líquido en los dos tanques) necesaria para producir un cambio de una unidad en el caudal.** Esto es:

$$R = \frac{\text{Cambio en la diferencia de nivel [m]}}{\text{Cambio en la velocidad de flujo } \left[\frac{m^3}{seg}\right]} \quad (3.9)$$

El concepto de resistencia es el mismo para ambos tipos de flujo, sin embargo su expresión tiene ciertas diferencias con respecto al caudal y la diferencia de nivel.

El sistema que se presenta en la fig. 3.17, se observa que el líquido sale a chorros por medio de la válvula de carga a un lado del tanque. Si el flujo a través de esta

¹El número de Reynolds se define como la relación entre las fuerzas inerciales y las fuerzas viscosas presentes en un fluido. Éste relaciona la densidad, viscosidad, velocidad y dimensión típica de un flujo en una expresión adimensional, que interviene en numerosos problemas de dinámica de fluidos. [4]

restricción es laminar, la relación entre el caudal en estado estable y la altura en estado estable se obtiene como:

$$Q = KH \quad (3.10)$$

Donde:

- Q: Caudal del liquido en estado estable. [m^3/seg]
- K: Coeficiente. [m^2/seg]
- H: Altura en estado estable. [m]

La Resistencia del flujo laminar R_l se obtiene como:

$$R_l = \frac{dH}{dQ} = \frac{H}{Q} \quad (3.11)$$

El flujo laminar, como dijimos anteriormente, es constante y tiene cierta analogía con la resistencia eléctrica.

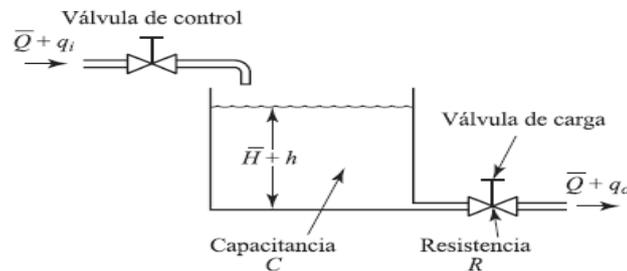


Figura 3.17: Sistema de nivel de un tanque. [21]

Si el flujo es turbulento a través de la válvula (restricción), el caudal en estado estable se obtiene mediante¹:

$$Q = K\sqrt{H} \quad (3.12)$$

La resistencia del flujo turbulento R_t es:

$$R_t = \frac{dH}{dQ} \quad (3.13)$$

Aplicando el diferencial a la ec. (3.12) se tiene:

$$dQ = \frac{K}{2\sqrt{H}}dH \quad (3.14)$$

de (3.14) resulta que:

$$\frac{dH}{dQ} = \frac{2\sqrt{H}}{K} = \frac{2\sqrt{H}\sqrt{H}}{Q} = \frac{2H}{Q} \quad (3.15)$$

Relacionando (3.13) y (3.14) tenemos que:

$$R_t = \frac{2H}{Q} \quad (3.16)$$

De la ecuación anterior se observa que, el valor de la resistencia de flujo turbulento depende tanto del caudal como de la altura.

Podemos expresar al caudal Q como:

$$Q = \frac{2H}{R_t} \quad (3.17)$$

Esta linealización es válida, siempre y cuando las variaciones en la altura y en el caudal, a partir de su estado estable, sean pequeños.

¹En el capítulo 4 se extiende su desarrollo para el sistema de este proyecto.

Capacitancia de un tanque

Otro de los conceptos que nos ayudaron a entender este tipo de sistemas, es el de capacitancia C de un tanque. Se define como el cambio necesario en la cantidad de liquido almacenado, para poder producir un cambio de una unidad en el potencial (altura).

$$C = \frac{\text{Cambio en el liquido almacenado}}{\text{Cambio en la altura}} \quad (3.18)$$

Cabe aclarar que la capacidad (m^3) y la capacitancia (m^2) son diferentes. La capacitancia del tanque es igual a su área transversal. Si esta es constante, la capacitancia es constante para cualquier altura.

3.4.2. Análisis de sistemas de nivel de liquido

Los conceptos anteriormente vistos, nos dieron la base para desarrollar el análisis y estudio de los sistemas de nivel de liquido.

En esta subsección se desarrolla el análisis de los sistemas de control que nos brindó los conocimientos para el desarrollo del sistema en este proyecto. Tal desarrollo se encuentra explayado en todo el capítulo 4.

Consideremos el sistema de la fig. 3.17 . Las variables que intervienen son:

- \bar{Q} : Caudal en estado estable. $[\frac{m^3}{seg}]$
- q_i : Desviación pequeña de la velocidad en entrada. $[\frac{m^3}{seg}]$
- q_o : Desviación pequeña de la velocidad en salida. $[\frac{m^3}{seg}]$
- \bar{H} : Altura en estado estable. $[m^3]$
- h : Desviación pequeña de la altura a partir de su valor en estado estable. $[m]$

Como vimos anteriormente, un sistema se considera lineal si el flujo es laminar. Aunque en aquellos sistemas con flujo turbulento se puede linealizar las ecuaciones no lineales, siempre que los cambios en las variables se mantengan pequeños.

De esta manera tenemos que la ecuación diferencial del sistema se obtiene como:

$$C \cdot dh = (q_i - q_o) \cdot dt \quad (3.19)$$

La ecuación anterior indica que, todo lo almacenado en el tanque es igual a todo el caudal que entra menos el caudal que sale, en un pequeño intervalo de tiempo.

A partir de la definición de resistencia, la relación entre el caudal y la altura se obtiene como:

$$q_o = \frac{h}{R} \quad (3.20)$$

Reemplazando (3.20) en (3.19), se encuentra la ecuación diferencial para un valor constante de R como:

$$RC \cdot \frac{dh}{dt} + h = R \cdot q_i \quad (3.21)$$

El termino RC significa la constante de tiempo del sistema, este punto es importante ya que para los fines didácticos se pensó en este tipo de sistemas de acuerdo a su constante de tiempo relativamente grande.

Aplicando la transformada de Laplace a la ec. (3.21) obtenemos la siguiente función de transferencia.

$$\frac{H(s)}{Q_i(s)} = \frac{R}{RCs + 1} \quad (3.22)$$

La función de transferencia anterior indica la relación entre la altura del tanque con respecto al caudal de entrada.

3.4.3. Análisis de sistemas de nivel en 2 tanques

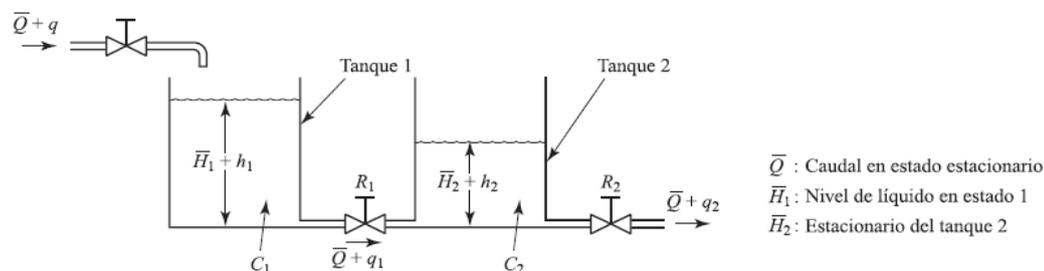


Figura 3.18: Sistema de nivel de dos tanque. [22]

La figura anterior muestra la interacción de 2 tanques conectados. Podríamos pensar que la función de transferencia será el producto de las dos funciones de transferencia de primer orden, sin embargo resulta interesante el análisis, ya que se obtiene un sistema más complejo de segundo orden.

Las ecuaciones diferenciales para este sistema son:

$$\frac{h_1 - h_2}{R_1} = q_1 \quad (3.23)$$

$$C_1 \frac{dh_1}{dt} = q - q_1 \quad (3.24)$$

$$\frac{h_2}{R_2} = q_2 \quad (3.25)$$

$$C_2 \frac{dh_2}{dt} = q_1 - q_2 \quad (3.26)$$

Se considera a q como entrada al sistema y q_2 la salida, por lo tanto la función de transferencia se obtiene como:

$$\frac{Q_2(s)}{Q(s)} = \frac{1}{R_1 C_1 R_2 C_2 s^2 + (R_1 C_1 + R_2 C_2 + R_2 C_1) s + 1} \quad (3.27)$$

Cabe hacer notar que la ecuación anterior nos brinda la relación entre caudales que ingresan y salen del sistema. En este proyecto se relacionó el caudal con respecto a la altura del tanque 2, cuando se elige la opción de trabajar con los tanques en interacción.

3.5. Relé de estado sólido

Una de las consideraciones que se debe tener en cuenta a la hora de trabajar con 220 voltios en corriente alterna, es la de poder aislarla con respecto a los componentes de corriente continua y así evitar inconvenientes. De esta manera se optó por diseñar un Relé de Estado Sólido (SSR)¹. En este capítulo se presenta su funcionamiento conceptualmente, mientras que en el capítulo posterior se encuentra desarrollado el diseño para este proyecto.

Un relé de estado sólido es un dispositivo interruptor electrónico, que conmuta el paso de la corriente cuando es activado por una señal en sus terminales de control.

Reciben el nombre de estado sólido, porque están formados por semiconductores de estado sólido, como son los tiristores, triac y transistores.

3.5.1. Acoplamiento

Por lo general el circuito de control con respecto a la carga conectada, se encuentra acoplado de una forma que se produzca un aislamiento galvánico².

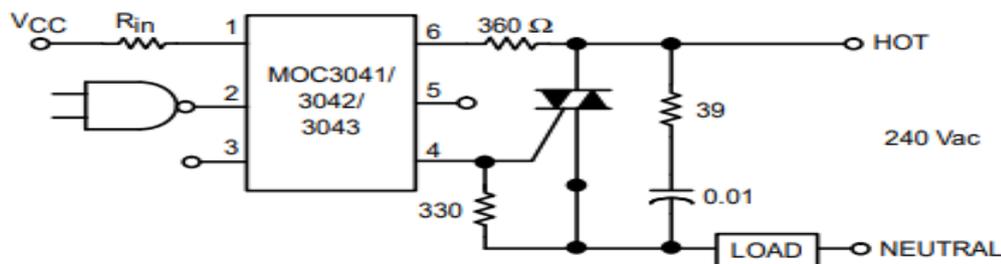


Figura 3.19: Relé de estado sólido provisto por el fabricante. [23]

La mayoría de los SSR utilizan un acoplamiento óptico, es decir hacen uso de un dispositivo electrónico llamado optoacoplador. El optoacoplamiento permite que el circuito de control esté eléctricamente aislado de la carga. Esto se puede visualizar en la figura 3.19, donde el fabricante de un optoacoplador (MOC 3041), brinda información sobre un relé de estado sólido con este dispositivo.

¹Del inglés Solid State Relay.

²El aislamiento galvánico consiste en la separación de partes funcionales de un circuito eléctrico para prevenir el traspaso de portadores de carga. [48]

3.5.2. Funcionamiento

Su funcionamiento es sencillo, la señal de control enciende un led interno que ilumina y activa un diodo fotosensible, la corriente de este diodo activa un tiristor, triac o mosfet para conmutar la carga.

Para corriente alterna, se suele emplear un triac que consta de dos tiristores conectados en direcciones opuestas. Los triac se utilizan porque la corriente alterna está constantemente cambiando de dirección; cuando la puerta del triac deja de recibir corriente, el triac cortará el paso de electricidad cuando el ciclo de la alterna pase por 0 (que sucede cada 20 ms para una frecuencia de 50 Hz), por lo que nunca se interrumpe el paso en un pico de la alterna, evitando los altos voltajes transitorios que de otra forma se producirían si se interrumpiera el paso bruscamente debido al colapso repentino del campo magnético sobre el inducido. [49]

Esta propiedad se denomina conmutación en paso por cero.

3.5.3. Ventajas sobre relés electromecánicos

- Diseño simple y económico para aplicaciones didácticas.
- Tamaño más reducido.
- Baja tensión de trabajo, por lo general se activan con 1.5 V. (dependiendo del optoacoplador)
- Funcionamiento totalmente silencioso.
- Vida útil más larga, ya que no posee parte mecánicas que produzca desgaste.
- No produce ruido debido a las chispas u ondas electromagnéticas que puedan producir interferencias en otros equipos.

3.6. Control de velocidad de un motor AC

Cuando se desea controlar la velocidad de un taladro, el nivel de iluminación o cualquier otro dispositivo que funcione con un motor de corriente alterna, es necesario realizar este tipo de control.

3.6.1. Triac

Este control utiliza como elemento central activo a un triac. Este es un dispositivo semiconductor de la familia de los tiristores, conocido como triodo AC.

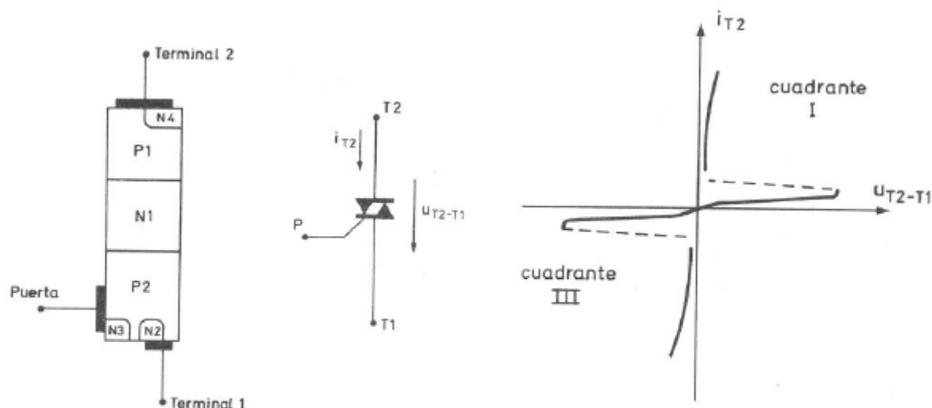


Figura 3.20: Estructura, símbolo y característica . [24]

Tiene la ventaja de ser un dispositivo capaz de bloquear tensión y conducir en ambos sentidos entre sus terminales T1 y T2. El hecho de que entre en conducción, si se supera la tensión de ruptura en cualquier sentido, lo hace inmune a destrucción por sobretensión.

La estructura contiene 6 capas, aunque siempre funciona como un tiristor de 4. En sentido T2-T1 conduce a través de P1N1P2N2 y de T1-T2 a través de P2-N1-P1-N4. La capa N3 facilita el disparo con corriente de puerta negativa.

Dado que su estructura es más complicada que la del SCR lo hace más sensible a la $\frac{di}{dt}$, $\frac{dv}{dt}$ y sobrecorrientes.

Se fabrican para corrientes de unos pocos amperios hasta 200 amperios y desde 400 V a 1000 V.

3.6.2. Modo de funcionamiento

Consideremos la imagen 3.21, donde muestra un diagrama del controlador de velocidad para un motor AC.

El funcionamiento de este tipo de control es sencillo, debido a que, al variar el valor de un potenciómetro (R2), se varía el ángulo de conducción del triac (T1), por

lo que la carga de los capacitores (C1 y C2) sera más rápida o lenta, de acuerdo a este valor de resistencia en el potenciómetro.

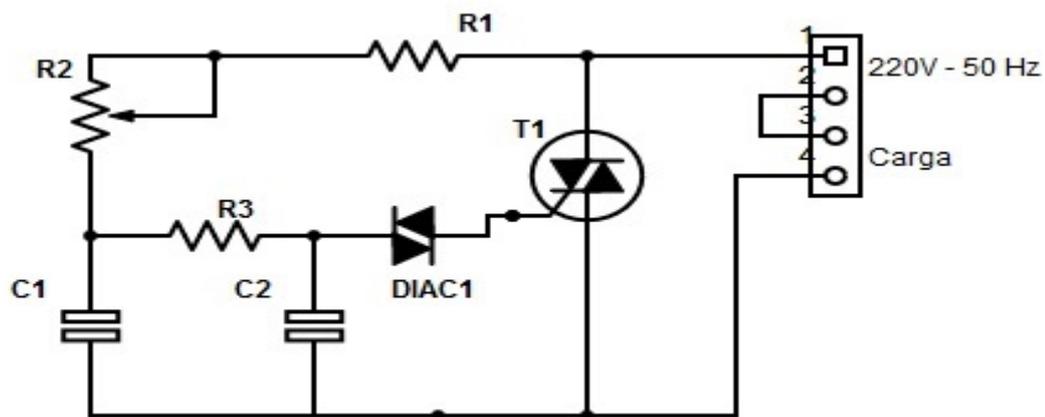


Figura 3.21: Circuito de control de fase (potencia) . [25]

Esa tensión almacenada en los capacitores activan el diac, el cual se utiliza para que el ángulo de conducción en los dos semiciclos (positivo y negativo) sean iguales. Luego activa el triac y este permite que se mantenga conduciendo corriente por el motor hasta que la señal senoidal (proveniente de la red) llegue al paso por cero.

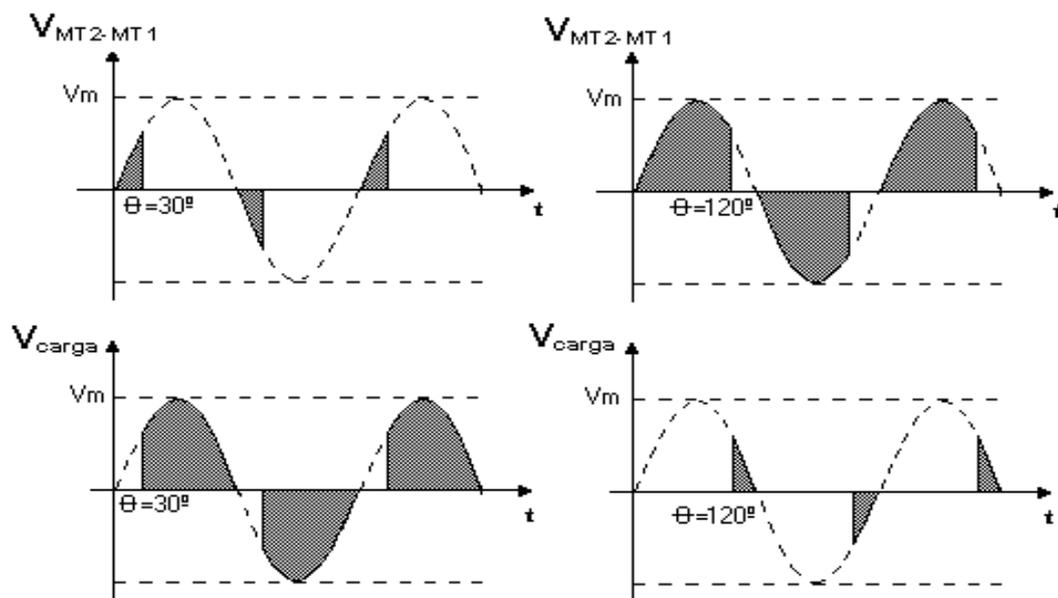


Figura 3.22: Gráfica de ondas de un triac . [26]

3.7. Plataformas de desarrollo

Se utilizaron diversas plataformas de desarrollo para el software de este proyecto, a continuación se detallan algunas.

3.7.1. Arduino[®] Mega 2560

Arduino[®] Mega 2560 es una placa electrónica de desarrollo, la cual esta basada en el microcontrolador ATmega 2560 del fabricante norteamericano Atmel[®]. Este dispositivo es el elemento central del proyecto donde tiene incorporado, 54 pines digitales de entradas/salidas (12 de las cuales se puede utilizar para salidas PWM), 16 entradas analógicas, un cristal oscilador de 16 MHz, una conexión USB para alimentación y comunicación con la PC, una entrada plug de alimentación y un botón de reset¹.

A continuación en la figura 3.23 se ilustra la placa Arduino[®] Mega 2560

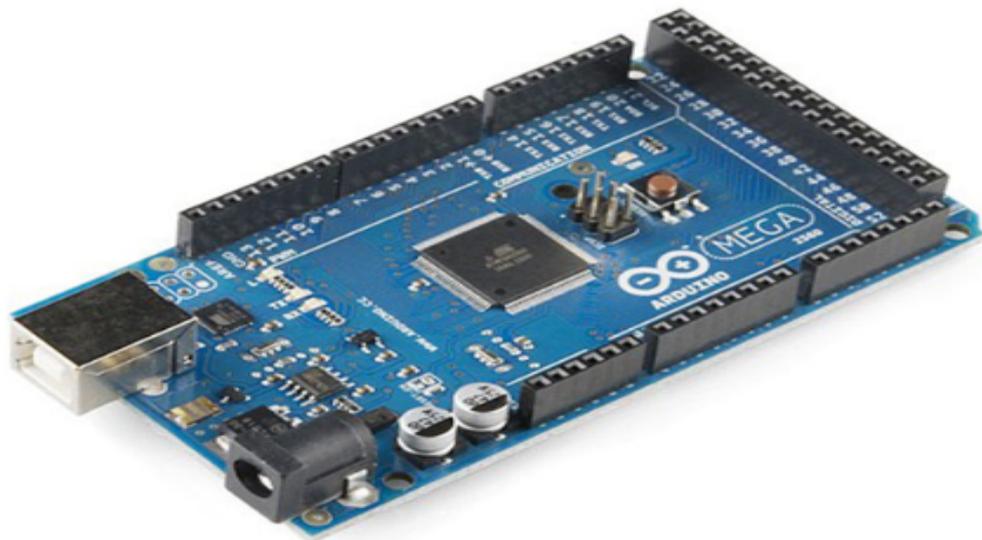


Figura 3.23: Arduino[®] Mega 2560. [27]

3.7.2. Matlab[®]

Matlab[®] es uno de los software más usado en el ámbito de ingenieros y científicos. Su nombre deriva del ingles MATrix LABoratory (Laboratorio de matrices).

¹En el Apéndice Datasheet de Componentes se encuentra mayor información sobre esta placa.

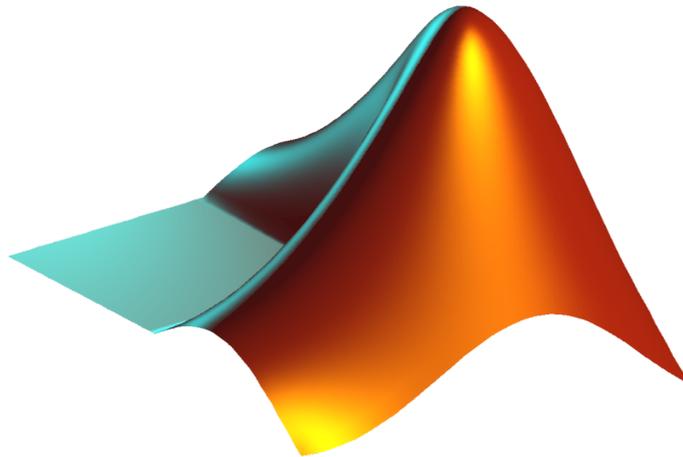


Figura 3.24: Logo Matlab[®] . [28]

Entre sus prestaciones básicas se hallan: la manipulación de matrices, la representación de datos y funciones, la implementación de algoritmos, la creación de interfaces de usuario (GUI) y la comunicación con programas en otros lenguajes y con otros dispositivos hardware. El paquete MATLAB dispone de dos herramientas adicionales que expanden sus prestaciones, a saber, Simulink (plataforma de simulación multidominio) y GUIDE (editor de interfaces de usuario - GUI). Además, se pueden ampliar las capacidades de MATLAB con las cajas de herramientas (toolboxes); y las de Simulink con los paquetes de bloques (blocksets) [28].

Utilice MATLAB para:

 <p>Analítica de Datos Averiguar cómo utilizar MATLAB con Big Data, machine learning y análisis de producción.</p>		 <p>Comunicaciones Inalámbricas Descubrir cómo puede ayudarle MATLAB a desarrollar algoritmos y realizar una simulación completa de sistemas inalámbricos.</p>			
 <p>Deep Learning</p>	 <p>Visión Artificial</p>	 <p>Procesamiento de señales</p>	 <p>Finanzas Cuantitativas y Gestión de Riesgos</p>	 <p>Robótica</p>	 <p>Sistemas de control</p>

Figura 3.25: Usos de Matlab[®] . [28]

3.7.3. Pantalla Táctil Nextion 7" HMI

La pantalla Nextion es una solución de interfaz de máquina humana (HMI), que proporciona una interfaz de control y visualización entre un humano y un proceso, una máquina, una aplicación o un dispositivo. La pantalla Nextion se aplica principalmente a Internet of thing (Internet de las cosas) o al campo de la electrónica de consumo. Es la mejor solución para reemplazar la tradicional pantalla LCD y el tubo Nixie LED.

Las características incluyen: una pantalla táctil resistiva TFT 800x480 de 7.0 ", 16M Flash, 2KByte RAM, 65k colores. [30]

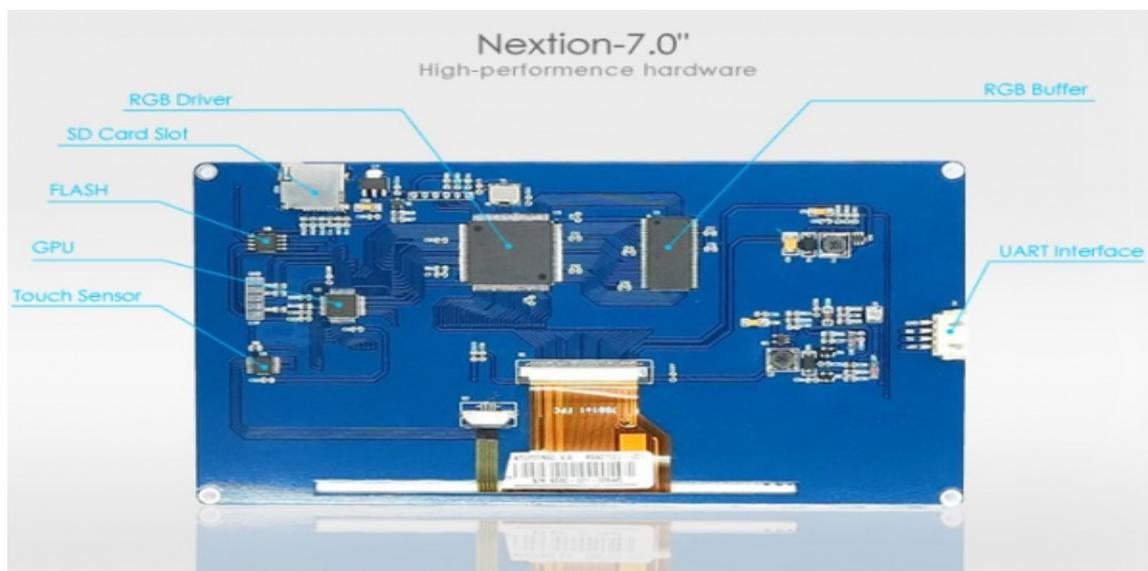


Figura 3.26: Pantalla Nextion NX8048T070 . [30]

Esta pantalla incluye, además de la parte de hardware (una serie de tableros TFT), una parte de software, el editor de Nextion.

3.7.4. Overleaf [®]

Otra herramienta poderosa y novedosa que se usó en este trabajo, es la de programar en LaTeX a través de un editor online.

Overleaf es un servicio de LaTeX colaborativo en línea con el cual se puede realizar trabajos finales, escribir artículos técnicos, realizar posters y slides¹ utilizando LaTeX

¹Diapositivas animadas.

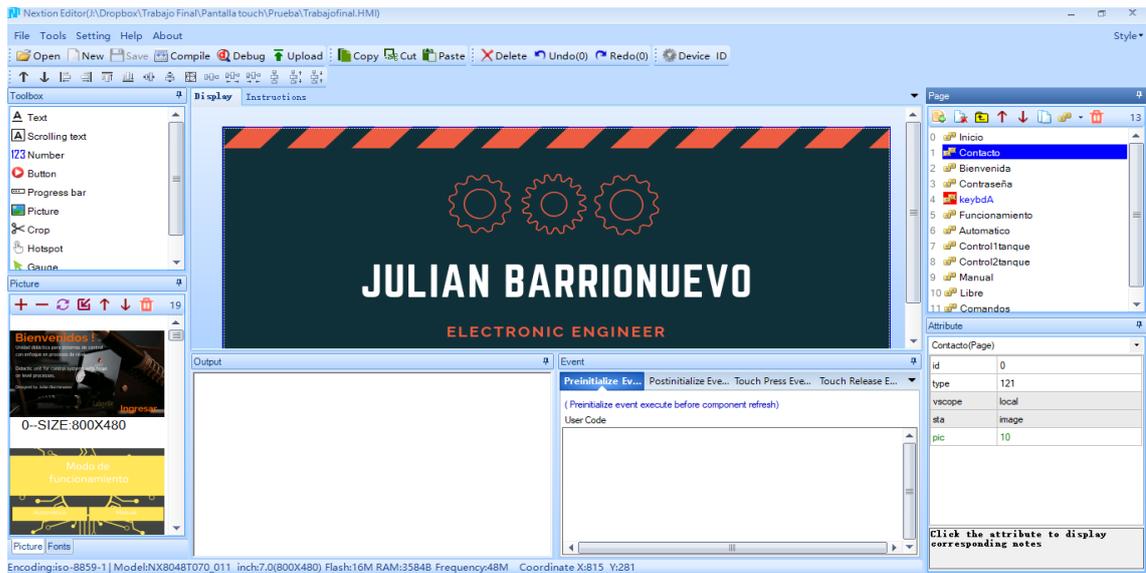


Figura 3.27: Editor de la pantalla Nextion. [30]

en la nube. Tiene una facilidad de uso sin necesidad de las instalaciones complejas de LaTeX.

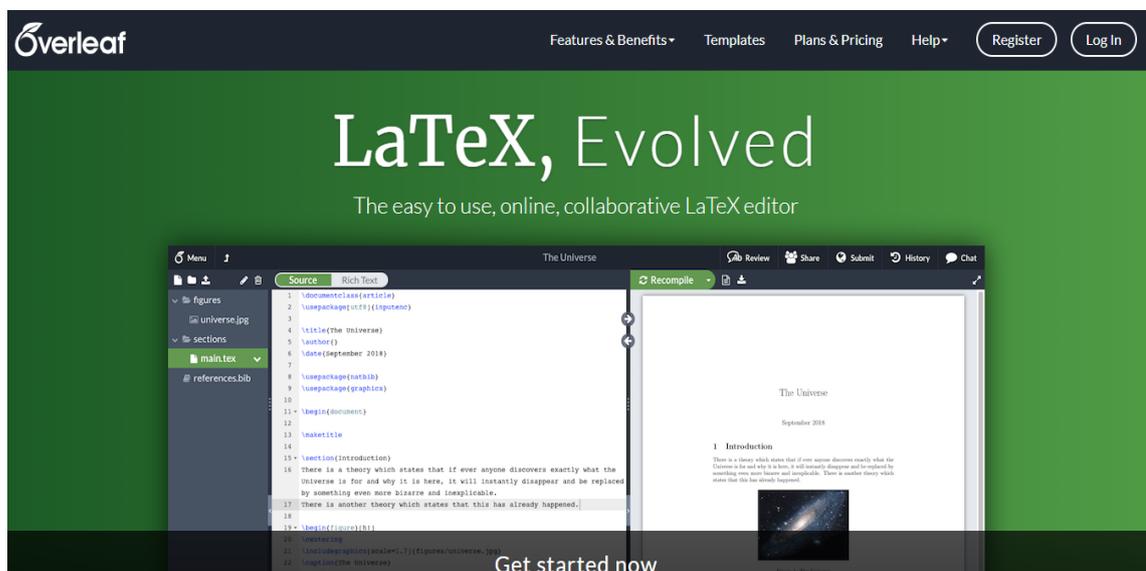


Figura 3.28: Editor de LaTeX online. [31]

Overleaf es una empresa emergente y social que construye modernas herramientas de creación colaborativa para ayudar a que la ciencia y la investigación sean más rápidas, más abiertas y más transparentes.

Capítulo 4

Memoria Descriptiva

4.1. Diseño y Construcción - Hardware

4.1.1. Diseño de la unidad didáctica

Lo primero que se efectuó fue, determinar la forma y dimensiones de la unidad, de tal modo que su visualización e interacción con las partes sean de manera cómoda y de fácil acceso.

Se optó por un diseño con terminación industrial de bajo peso y de gran tamaño, de manera que pueda contener todos los actuadores y se asimile más a las unidades comerciales existentes. Se usaron maderas tipo MDF¹ de 9 mm de espesor y tubos estructurales de 1' para el armado de la estructura.

Las medidas que posee son:

- Altura: **1500 mm.**
- Anchura: **1000 mm..**
- Profundidad: **500 mm.**

Se buscaron colores acordes a la temática del proyecto y con la primicia de que alcancen una agradable visualización. La unidad presenta un acabado metalizado de color plata 2657 en sus soportes estructurales y un color azulado en esmalte sintético de manera que contrasta con el color natural del agua.

Si bien el peso que presenta la estructura no es de gran porte, se introdujeron dos ruedas y manijas de modo que el traslado sea de mayor facilidad.

¹Del inglés Medium Density Fibreboard.



Figura 4.1: Estructura de la unidad. [32]

4.1.2. Modelo y simulación de dimensiones de los tanques

Dado que se trata de un modulo didáctico, se ha preestablecido que la unidad trabaje en diferentes modos, dicho de otra manera, que la planta del sistema actúe como un simple sistema de primer orden o como un sistema más complejo de segundo orden. Para ello, se ha construido un sistema de 2 tanques acoplados entre si a través de válvulas de control de flujo, que permiten variar el comportamiento del sistema.

El orden del sistema esta dado por el numero de tanques con que se quiera trabajar. Para determinar las medidas de los tanques es necesario realizar un modelado y simulación de la planta a fin de conocer las dimensiones adecuadas, de forma que el sistema cumpla con nuestro propósito. A continuación se presentan las ecuaciones necesarias para representar al sistema de acuerdo al orden elegido.

4.1.3. Modelado estático del sistema de nivel

Para conocer las ecuaciones que representan el sistema de nivel, comenzamos analizando un tanque para determinar la velocidad de flujo que sale por una abertura

lisa y redonda. Para ello, se aplica la teoría de Bernoulli entre el punto de referencia ‘1’ en la superficie y el punto de referencia ‘2’ en la abertura de salida, figura 4.2.

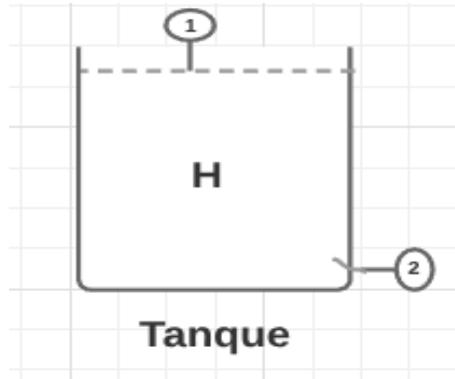


Figura 4.2: Tanque abierto . [32]

Considerando que la presión manométrica entre los puntos 1 y 2 es igual a cero, que ‘m’ es la masa de las partículas de fluido y que ‘v’ es la velocidad de salida de las partículas 1 y 2, por conservación de energía se tiene :

$$\frac{1}{2}mv^2 = mgH \quad (4.1)$$

$$v = \sqrt{2gH} \quad (4.2)$$

El flujo Q que emerge por la salida del tanque para el caso de una sección transversal ‘S’ es:

$$Q = S.v \quad (4.3)$$

Reemplazando (4.2) en la ecuación anterior se consigue:

$$Q = S\sqrt{2gH} \quad (4.4)$$

Debido a que este sistema de control presenta válvulas, analizamos su funcionamiento interno.

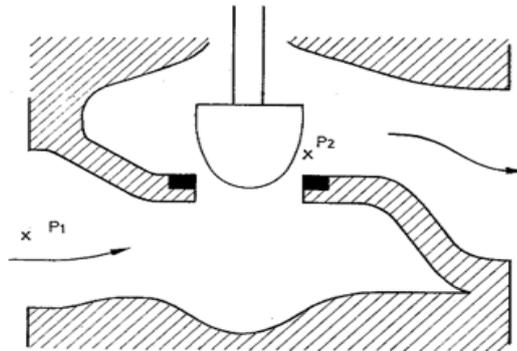


Figura 4.3: Válvula de control . [33]

La válvula se comporta, esencialmente, como un orificio de paso variable que permite la circulación de un cierto caudal con una determinada pérdida de carga. De la figura 4.3, consideramos de manera general, que el flujo que pasa por una válvula en estado estacionario es dado por:

$$Q_v = K A_s \sqrt{\Delta P} \quad (4.5)$$

Donde:

- Q_v = Flujo a través de la válvula.
- K = Constante.
- A_s = Área de paso.
- ΔP = Presión diferencial a través de la válvula en P1-P2

Se puede concluir que el flujo que pasa por la válvula, es proporcional al área de abertura de la válvula siempre y cuando la diferencia de presión sea constante.

Para hacerlo de una manera práctica consideramos a la válvula con un comportamiento inteligente, de modo que sea posible realizar una aproximación más o menos lineal entre el flujo Q_v y la abertura de la válvula.

Por (4.4) tenemos que:

$$Q_v = K a_s \sqrt{2gH} \quad (4.6)$$

Relacionando el caudal con la constante de la válvula obtenemos:

$$Q_v = K_v \sqrt{H} \quad (4.7)$$

La constante K_v conocida como coeficiente o factor de la válvula, corresponde por definición **al caudal en m^3/h para una pérdida $\Delta P = 1Bar$ y la densidad del agua entre $5\text{ }^\circ\text{C}$ y $30\text{ }^\circ\text{C}$ es de $1000\text{ Kg}/m^3$, entonces resulta [33]:**

$$K_v = 5.091s\beta \quad (4.8)$$

Donde:

- s = Sección del orificio de paso, en m^2 .
- β = coeficiente de resistencia (sin dimensiones) y < 1 .

4.1.4. Modelado dinámico del sistema de nivel

Consideremos a la planta como un sistema de primer orden (solo un tanque), como se muestra en la figura 4.4.

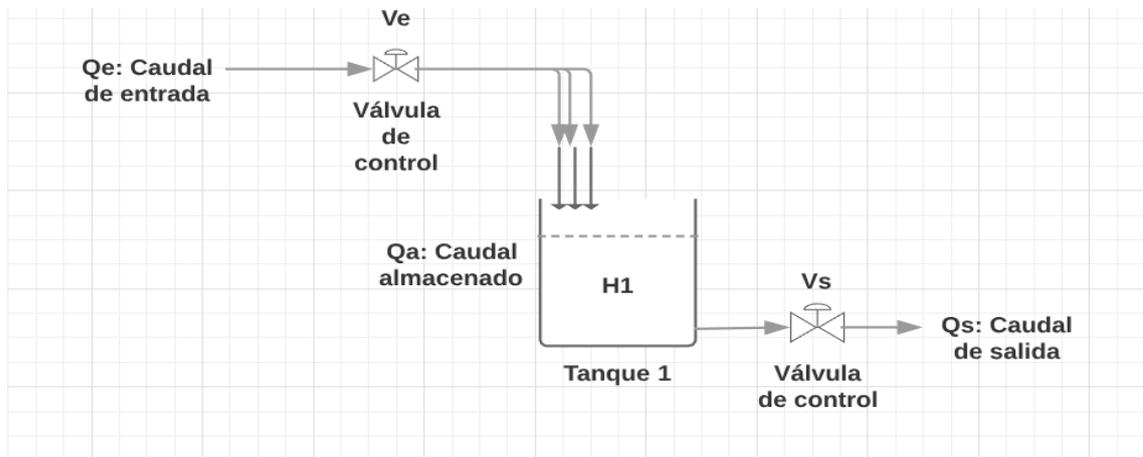


Figura 4.4: Esquema básico de un sistema de primer orden . [32]

Existen diversas maneras de controlar el nivel de un tanque, por ejemplo:

- Tanque controlado por la **válvula de entrada** V_e , considerando a la válvula de salida V_s como perturbación.
- Tanque controlado por la **válvula de salida** V_s , considerando a la válvula de entrada V_e como perturbación.

Para este trabajo se analizó el segundo caso, dado que se utilizó una electroválvula de elaboración propia para controlar el nivel.

En estas condiciones las ecuaciones dinámicas que representan al sistema son:

$$Q_e = Q_a + Q_s \quad (4.9)$$

Donde:

- Q_e = Caudal de entrada al tanque [m^3/s]
- Q_a = Caudal almacenado en el tanque [m^3/s]
- Q_s = Caudal de salida del tanque [m^3/s]

Vamos a suponer que el flujo de entrada Q_e es proporcional a la abertura de la válvula considerando un suministro constante, entonces las ecuaciones son:

$$Q_e = K_e \cdot a_e \quad (4.10)$$

$$Q_s = K_s \cdot a_s \cdot \sqrt{2gh_1} \quad (4.11)$$

$$Q_a = A \cdot \frac{dh_1}{dt} \quad (4.12)$$

Donde:

- K_e = Constante
- a_s = Abertura de la válvula
- g = Fuerza de la gravedad
- h_1 = Altura del nivel
- A = Área del tanque

El área del tanque por la variación de la altura en el tiempo es igual **a todo lo que entra menos todo lo que sale**, matemáticamente esto es:

$$Q_a = A \cdot \frac{dh_1}{dt} = Q_e - Q_s \quad (4.13)$$

$$A \cdot \frac{dh_1}{dt} = K_e \cdot a_e - K_s \cdot a_s \cdot \sqrt{2gh_1} \quad (4.14)$$

El segundo termino de la ecuación (4.14) no es lineal, entonces se debe buscar un punto de equilibrio. Este punto se encuentra cuando no existe variación en el volumen del tanque, esto es:

$$A \cdot \frac{dh_1}{dt} = Q_e - Q_s = 0 \quad (4.15)$$

Para linealizar el sistema lo hacemos a través de las series de Taylor.

$$\begin{aligned} f(a_s, h_1) &= A \frac{dh_1}{dt} \\ f(a_s, h_1) &\approx f(\bar{a}_s, \bar{h}_1) + \frac{\delta f}{\delta a_s} \Delta a_s \Big|_{(\bar{a}_s, \bar{h}_1)} + \frac{\delta f}{\delta h_1} \Delta h_1 \Big|_{(\bar{a}_s, \bar{h}_1)} \\ \Delta a_s &= a_s - \bar{a}_s \\ \Delta h_1 &= h_1 - \bar{h}_1 \\ f(a_s, h_1) &\approx A \frac{\delta h_1}{\delta t} \Big|_{(\bar{a}_s, \bar{h}_1)} - K_s \sqrt{2g\bar{h}_1} \Delta a_s - \frac{K_s \bar{a}_s \sqrt{2g}}{2\sqrt{\bar{h}_1}} \Delta h_1 \end{aligned}$$

La ecuación linealizada queda de la siguiente manera:

$$A \frac{d\Delta h_1}{dt} \approx -K_s \sqrt{2g\bar{h}_1} \Delta a_s - \frac{K_s \bar{a}_s \sqrt{2g}}{2\sqrt{\bar{h}_1}} \Delta h_1$$

Aplicando transformada de Laplace obtenemos:

$$A s h(s) = -K_s \sqrt{2g\bar{h}_1} a_s(s) - \frac{K_s \bar{a}_s \sqrt{2g}}{2\sqrt{\bar{h}_1}} h(s)$$

$$\frac{h_1(s)}{a_s(s)} = \frac{-K_s \sqrt{2g\bar{h}_1}}{A s + \frac{K_s \bar{a}_s \sqrt{2g}}{2\sqrt{\bar{h}_1}}}$$

Relacionando con las ecuaciones (4.6) y (4.7), conseguimos la función de transferencia del sistema para un tanque controlado por la válvula de salida como:

$$\frac{H_1(s)}{Q_s(s)} = \frac{-1}{A s + \frac{K_v}{2\sqrt{H_1}}} \quad (4.16)$$

Donde :

- H_1 = Altura de trabajo.
- A = Sección transversal del tanque.
- K_v = Coeficiente de la válvula.

Para facilitar los cálculos podemos expresar como:

$$\frac{H_1(s)}{Q_s(s)} = \frac{-k_2}{s + k_1} \quad (4.17)$$

Donde:

$$k_1 = \frac{K_v}{2A\sqrt{H}}$$

$$k_2 = \frac{1}{A}$$

Como se mencionó anteriormente, el sistema puede ser controlado por diferentes maneras, la ecuación (4.16) cita a un control por la válvula de salida. Para encontrar la relación del nivel con el caudal de entrada se realizó el análisis de la misma manera, con lo que se obtuvo la siguiente función de transferencia:

$$\frac{H_1(s)}{Q_e(s)} = \frac{1}{As + \frac{K_v}{2\sqrt{H_1}}} \quad (4.18)$$

$$\frac{H_1(s)}{Q_e(s)} = \frac{k_2}{s + k_1} \quad (4.19)$$

Si se considera ahora el sistema de segundo orden (dos tanques en cascada) como muestra la figura 4.5, las ecuaciones dinámicas que describen al sistema son:

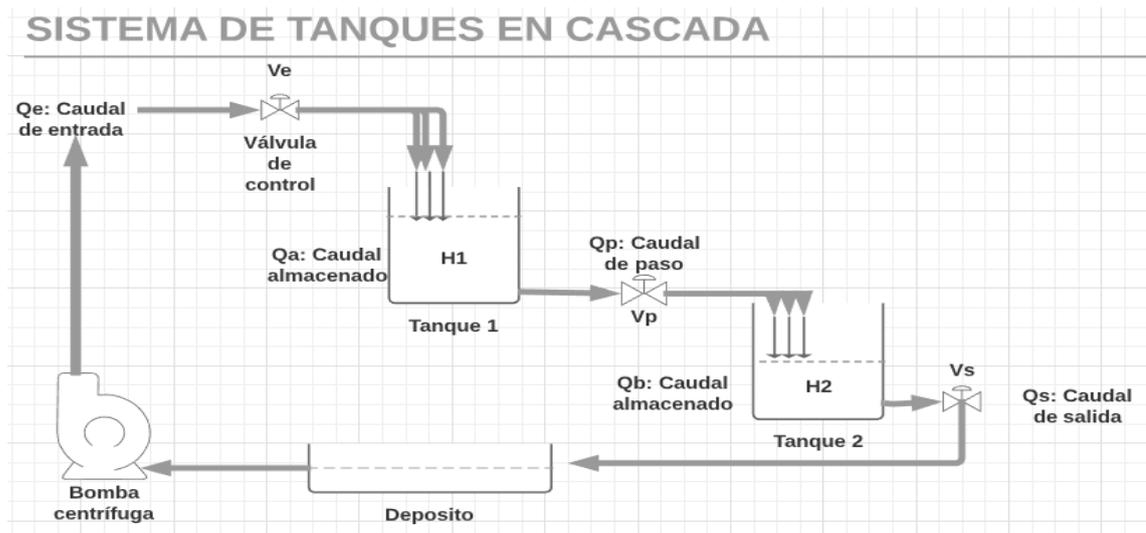


Figura 4.5: Esquema básico de un sistema de segundo orden . [32]

$$Q_a = Q_e - Q_p \quad (4.20)$$

$$Q_b = Q_p - Q_s \quad (4.21)$$

Haciendo las mismas consideraciones que en el caso anterior, vamos a suponer que el flujo de entrada es proporcional a la abertura de la válvula considerando un suministro constante, que todas las válvulas de control son de idénticas características y que los tanques tiene la misma sección transversal, entonces las ecuaciones que representan al sistema son:

$$Q_a = A_1 \frac{dh_1}{dt} = K_e a_e - K_p a_p \sqrt{2g(h_1 - h_2)} \quad (4.22)$$

$$Q_b = A_2 \frac{dh_2}{dt} = K_p a_p \sqrt{2g(h_1 - h_2)} - K_s a_s \sqrt{2gh_2} \quad (4.23)$$

Donde:

- K_e = Constante de caudal de entrada
- K_p = Constante de caudal de paso
- K_s = Constante de caudal de salida
- a_s = Abertura de la válvula de salida
- a_e = Abertura de la válvula de entrada
- a_p = Abertura de la válvula de paso
- g = Fuerza de la gravedad
- h_1 = Altura del nivel tanque 1
- h_2 = Altura del nivel tanque 2
- A = Área de los tanques

Para linealizar las ecuaciones anteriores, se procede de igual manera como se realizó para el modelo de un tanque, luego se aplica la transformada de Laplace para encontrar la función de transferencia.

Ecuación tanque n° 1:

$$Ash_1(s) = -K_p\sqrt{2g(h_1 - h_2)}a_p(s) - \frac{K_p\bar{a}_p\sqrt{2g}}{2\sqrt{(h_1 - h_2)}}h_1(s)$$

$$\frac{H_1(s)}{Q_p(s)} = \frac{-1}{As + \frac{K_v}{2\sqrt{(H_1 - H_2)}}} \quad (4.24)$$

Ecuación tanque n° 2:

$$Ash_2(s) = -K_p a_p \sqrt{2g(h_1 - h_2)}h_1(s) - \frac{K_s a_s \sqrt{2g}}{2\sqrt{(h_1 - h_2)}}h_2(s)$$

$$\frac{H_2(s)}{Q_p(s)} = \frac{\frac{K_v}{2\sqrt{(H_1 - H_2)}}}{A^2s^2 + A\left(\frac{K_v}{2\sqrt{(H_1 - H_2)}} + \frac{K_v}{2\sqrt{H_2}}\right)s + \frac{K_v}{2\sqrt{(H_1 - H_2)}}\frac{K_v}{2\sqrt{H_2}}} \quad (4.25)$$

Para facilitar los cálculos tenemos:

$$\frac{H_2(s)}{Q_p(s)} = \frac{k_1k_2}{s^2 + s(k_1 + k_3) + k_1k_3} \quad (4.26)$$

Donde:

$$k_1 = \frac{K_v}{2A\sqrt{H_1 - H_2}}$$

$$k_2 = \frac{1}{A}$$

$$k_3 = \frac{K_v}{2A\sqrt{H_2}}$$

Como se mencionó anteriormente, el sistema puede ser controlado de diferentes maneras, la ecuación (4.25) cita a un control por la válvula de paso. Para encontrar la relación del nivel con el caudal de entrada se realizó el análisis de la misma manera, con lo que se obtuvo la siguiente función de transferencia:

$$\frac{H_2(s)}{Q_e(s)} = \frac{\frac{K_v}{2\sqrt{(H_1 - H_2)}}}{A^2s^2 + A(K_v\sqrt{H_1 - H_2} + 2K_v\sqrt{H_2})s + \frac{K_v}{2\sqrt{(H_1 - H_2)}}\frac{K_v}{2\sqrt{H_2}}} \quad (4.27)$$

$$\frac{H_2(s)}{Q_e(s)} = \frac{k_1 k_2}{s^2 + s(2k_1 + k_3) + k_1 k_3} \quad (4.28)$$

4.1.5. Simulación de las funciones de transferencia

Se realizaron diversas simulaciones con las ecuaciones (4.16) y (4.25) aplicando una entrada de tipo escalón y se determinó que las dimensiones más adecuadas que debe tener cada tanque para que la unidad trabaje adecuadamente, es decir, no presente una constante de tiempo muy grande, son:

- A = Sección transversal del tanque = 10 cm x 17.5 cm = 175 cm²
- H = Altura de cada tanque = 30 cm
- Kv = Válvula de control = 5,091 $[\frac{m}{h}]$. $3,1416 \times 10^{-4} [m^2] \cdot 1 = 0,0024 [\frac{m^3}{h}] = 0.694 [\frac{cm^3}{s}]$

Para obtener la respuesta del modelo de un tanque para una altura de trabajo de 15 cm, se efectuó un script en Matlab, la siguiente gráfica muestra la respuesta en el tiempo en lazo abierto ante una entrada escalón unitario.

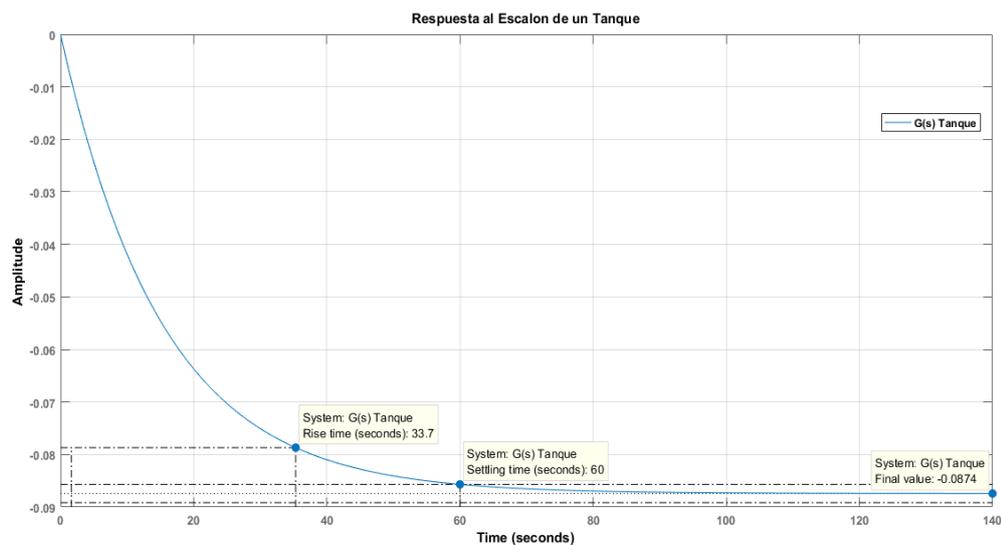


Figura 4.6: Respuesta a lazo abierto al modelo de un tanque . [32]

Podemos comentar de la respuesta obtenida en la fig. 4.6, que presenta un tiempo de establecimiento de 60 s, lo que hace una buena respuesta con esa medida seleccionada del tanque. Además al tratarse de un valor negativo de la ganancia es el motivo de la forma de la respuesta, esto sera compensada a través del controlador que se aplique.

La respuesta ante una entrada escalón para el modelo de dos tanques en cascada, citando las alturas de trabajo de $H_1= 15$ cm y $H_2=10$ cm, se muestra en la figura 4.7.

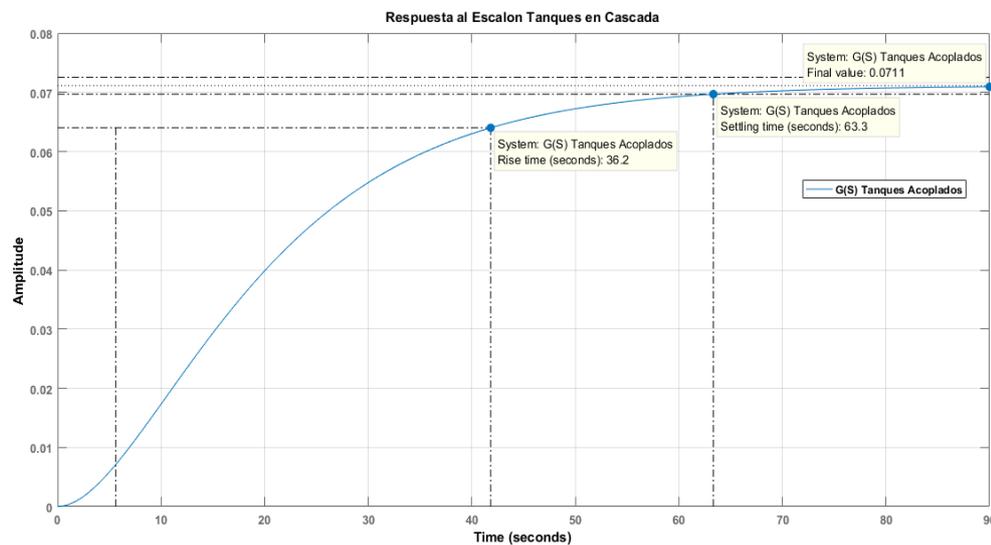


Figura 4.7: Respuesta lazo abierto al modelo de dos tanques . [32]

Para este sistema en lazo abierto, se obtuvo un tiempo de subida de 36.2 segundos, no presenta sobreimpulso, y con un tiempo de establecimiento de 63.3 s, lo que confirma que las medidas de los tanques son apropiadas para realizar el control.

Es útil mencionar que las aproximaciones que se realizaron para obtener este modelo lineal, **son validas en la cercanía de los puntos de operación**. Es por ello se las utiliza para determinar las constantes del tiempo del sistema y así determinar las dimensiones apropiadas, además esto nos permite tener perspectiva hacia el diseño del control.

4.2. Sensor de Presión

La medición de una variable es la parte más delicada en todo proceso de control, ya que la precisión del elemento encargado de tomar muestras interviene en todo el proceso de control del sistema, es decir, una mala lectura producirá un mal control del sistema. Debido a esto se buscó un sensor que presente ciertas características, como gran precisión, buena sensibilidad, mayor inmunidad al ruido, linealidad en el rango de trabajo y sin tantas variaciones a causa de agentes externos como la temperatura y humedad.

En el mercado existe una amplia variedad de sensores, en diferentes clasificación, están los de tipos inductivos, capacitivos, fotoeléctricos, magnéticos, ultrasonidos, como para citar algunos.

Esta gran oferta hizo que se eligiera uno acorde a la relación disponibilidad-precio, buscando que las características antes mencionadas se encuentren presentes. Por lo tanto se optó por un sensor, además estudiado para otro proyecto, de presión diferencial piezoresistivo de estado solido Mpx5010DP.



Figura 4.8: Sensor de presión diferencial Mpx5010DP . [34]

Este sensor mide indirectamente el nivel de líquido en el tanque, dado que en realidad lo que mide es la presión hidrostática en el fondo del tanque, sin embargo como la presión hidrostática es directamente proporcional a la altura del líquido que la produce se puede utilizar este sensor para medir el nivel del líquido.

Los sensores de presión diferencial actúan con la medida de presión entre dos puntos, en este caso, la presión hidrostática dentro del tanque es un punto a medir

y el otro punto, considerado como referencia, es la presión atmosférica. Estas dos medidas se comparan y brindan una salida eléctrica, positiva o negativa dependiendo de una muestra con respecto a la otra.

Este tipo de sensor es útil además para aplicaciones en instrumentación y medida, electrónica de consumo, dispositivos portátiles, automatización y control de procesos, etc.

4.2.1. Característica del sensor

El Mpx5010DP es un sensor de presión de silicio integrado de doble puerto en un encapsulado SIP de 6 pines. Este transductor piezorresistivo es un sensor de presión de silicio monolítico de última generación, diseñado para una amplia gama de aplicaciones. Es ideal para sistemas basados en microprocesador o microcontrolador.

Este transductor combina técnicas avanzadas de micro mecanizado, metalización de película delgada y procesamiento bipolar para ofrecer una señal de salida analógica precisa de alto nivel que es proporcional a la presión aplicada. El puerto axial ha sido modificado para acomodar tubos de grado industrial.¹ [34]

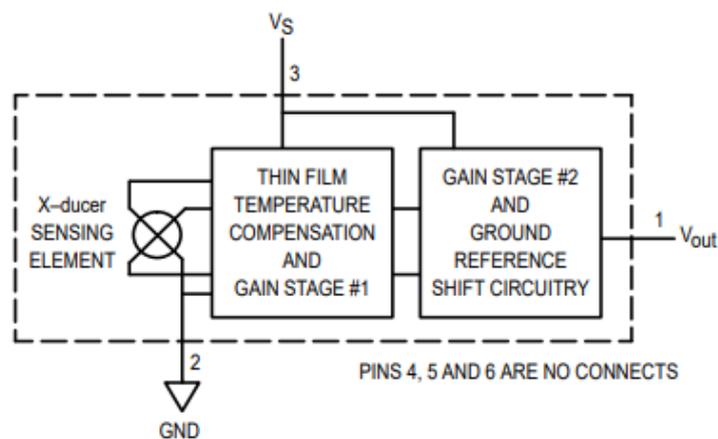


Figura 4.9: Esquemático sensor de presión. [34]

- Calibración, compensación de temperatura y acondicionamiento de señal en chip.
- Error máximo de 5,0 % entre 0 °C y 85 °C.

¹En el apéndice Datasheet de componentes se encuentra mayor información acerca de este sensor.

- Sensibilidad de 450mV/mm.
- Rango de presión de 0 KPa a 10 KPa.
- Rango de alimentación 4.75VDC a 5.25VDC
- Salida 0.2 a 4.7 V .

4.2.2. Acondicionamiento de la señal

El acondicionamiento de la señal es sumamente importante a la hora de programar el sensor en una aplicación, debido a que existen variados microcontroladores con diferentes características, como por ejemplo la resolución en el conversor A/D, conlleva a realizar un acondicionamiento de la señal que entrega el sensor para que la aplicación funcione de manera óptima.

Este tipo de sensor tiene su salida compensada en temperatura y amplificada la cual hace que sea compatible con los microcontroladores que poseen conversores A/D de bajo costo. Los mas comunes son aquellos que presentan un conversor A/D de 8-bit, siendo de alta resolución. Esta alta resolución hace que el ruido sea tenido en cuenta.

Existen dos tipos de ruido dominantes en el sensor piezoresistivo, el ruido blanco (Shot White) y el ruido debido a la frecuencia. El ruido blanco es el resultado del flujo no uniforme de los portadores a través de la junta y es independiente de la temperatura. El segundo ruido, resulta también de los defectos del cristal, es inversamente proporcional a la frecuencia y se considera más dominante en bajas frecuencias. El ruido también puede venir de circuitos externos, como fuentes de alimentación.

El sensor produce un pequeño voltaje diferencial del orden de los milivoltio. El amplificador del sensor, amplifica esta señal y la traduce a una salida de típica de 0.2 v a 4.7 v. A pesar de que el transductor tiene una respuesta mecánica alrededor de los 500 Hz, el ruido de salida se extiende en el rango de 500 Hz a 1 MHz. Este ruido se puede observar en la figura 4.10 .

Para contrarrestar los efectos del ruido en este sensor existen 2 métodos efectivos, puede ser realizando un filtro paso bajo por hardware y un filtro paso bajo por software.

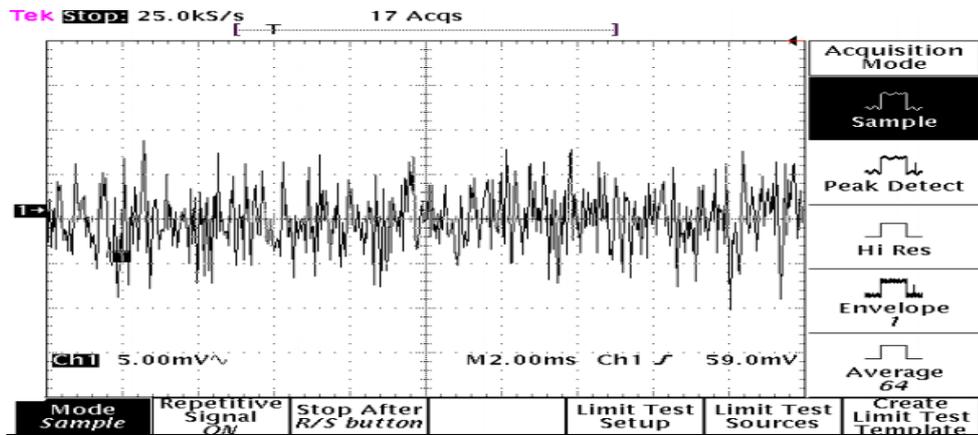


Figura 4.10: Ruido amplificado en sensor de presión. [41]

Para este proyecto se realizó un filtro pasa bajo por hardware debido a que, un filtro pasa bajo por software requiere memoria en el microcontrolador y llegó a ser un recurso limitado para todo el proyecto.

Se efectuó el filtro paso bajo que recomienda el fabricante, en conjunto con un amplificador operacional en configuración en seguidor de tensión (buffer) para la impedancia de la entrada al convertor.

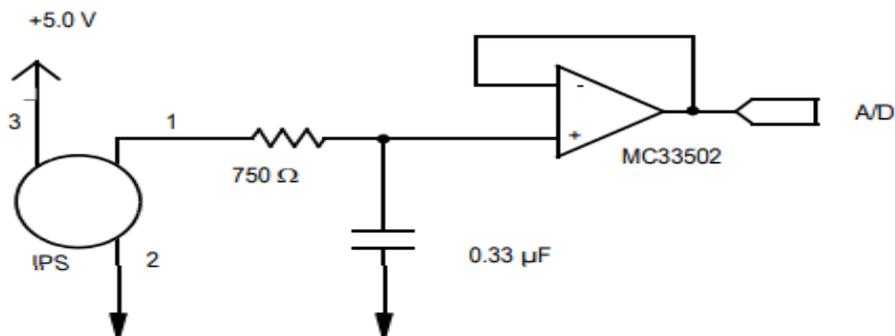


Figura 4.11: Filtro pasa bajo. [41]

Este filtro presenta una frecuencia de corte de 650 Hz, pero debido a los valores comerciales de resistencias disponible, se diseñó un filtro pasa bajo con una frecuencia de corte de 709.247 Hz como muestra la figura 4.13.

Por otro lado, la salida del sensor también depende de la tensión de alimentación, así una variación en la tensión de entrada produce variación en la salida. Para neutralizar estas variaciones debido al ruido proveniente de la fuente de alimentación, se

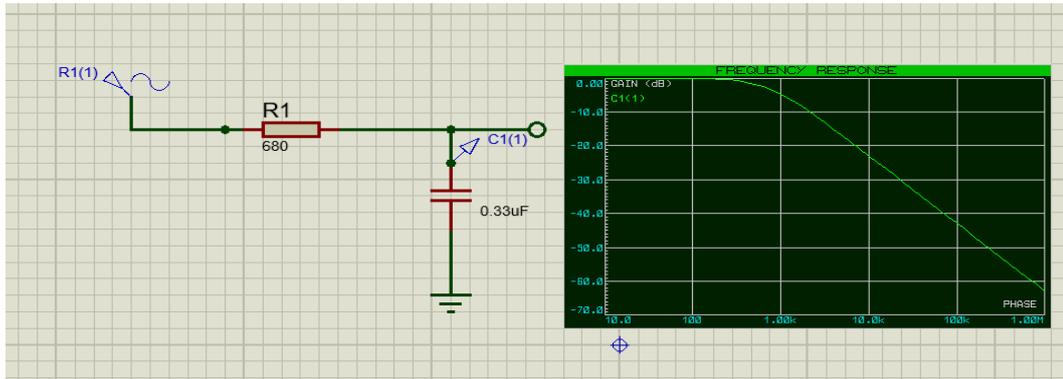


Figura 4.12: Simulación filtro pasa bajo. [32]

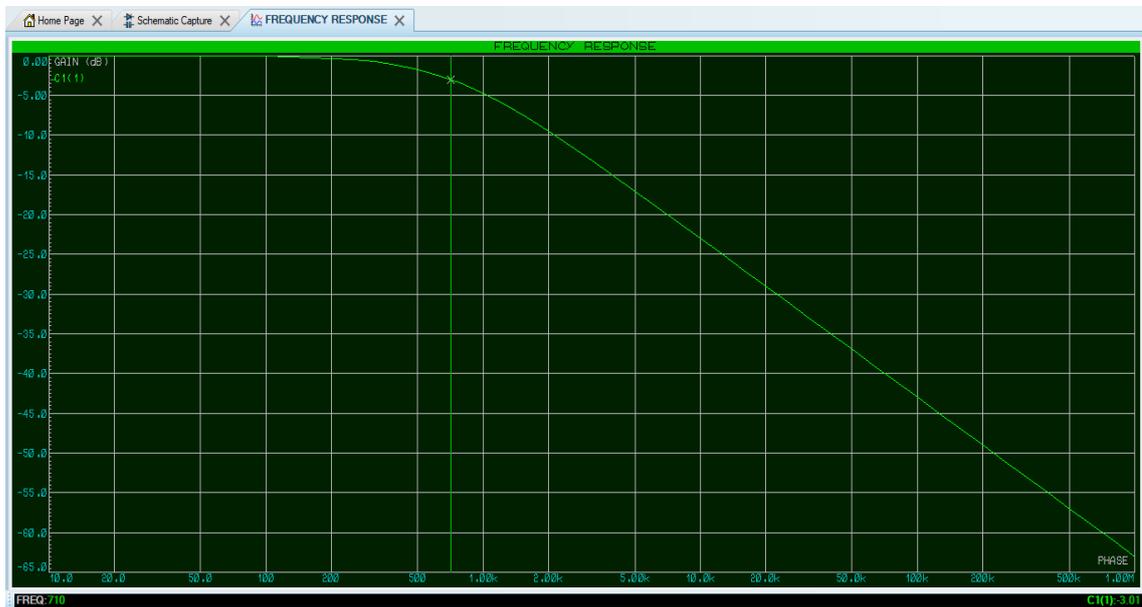


Figura 4.13: Respuesta en frecuencia de filtro pasa bajo . [32]

agrega un capacitor de 0.33 uF a 1.0 uF en paralelo con un capacitor de 0.01 uF.

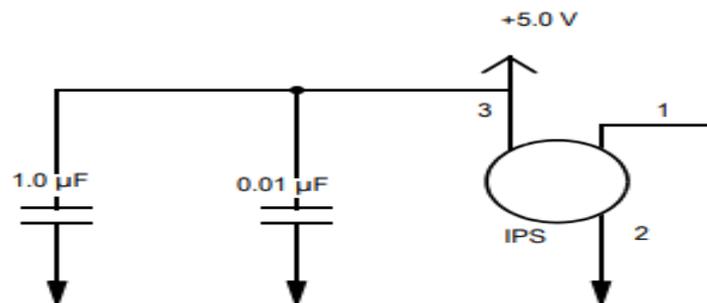


Figura 4.14: Filtro para fuente de alimentación. [41]

4.2.3. Acondicionamiento de señal para conversores de 0 a 10 V

Debido a que esta unidad presenta un funcionamiento libre, es decir, permite el uso de cualquier microcontrolador. Existen controladores, más precisamente los industriales, que presentan un voltaje de operación en su conversor A/D diferente a la que brinda este sensor (0.2 a 4.7 V).

Es por ello que se diseñó una configuración electrónica. Se utilizó un amplificador operacional en configuración **no inversor**, como se muestra en la figura 4.15.

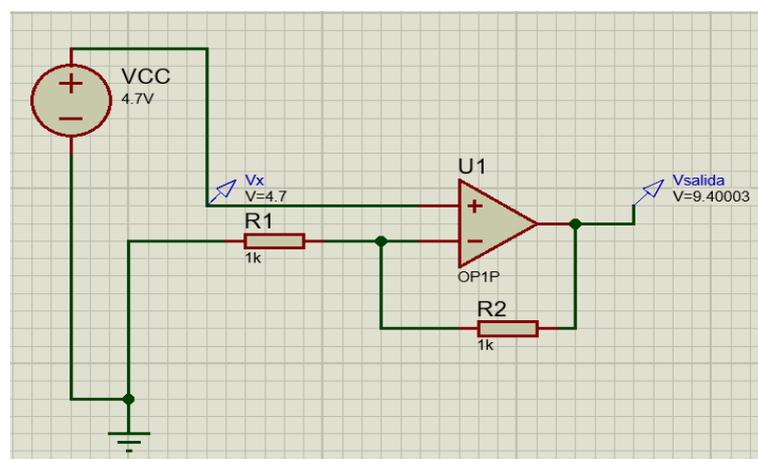


Figura 4.15: Acondicionamiento señal para A/D. [32]

$$V_o = \left(1 + \frac{R_2}{R_1}\right) \cdot V_{cc} \quad (4.29)$$

Esta configuración en **no inversor** resulta una solución adecuada debido a las características de ella. Como se observa en la ec. (4.29) la ganancia de esta configuración, se obtiene simplemente conectando dos resistencias del mismo valor, y la salida no se encuentra invertida. Otra de las ventajas de esta configuración es que actúa contra el efecto carga. Es decir, al conectar una carga disminuye la tensión de salida, es por ello que se utiliza esta configuración debido a su alta impedancia de entrada proporcionando un efecto de aislamiento de la salida con respecto a la entrada.

4.2.4. Función de Transferencia del Sensor

Este tipo de sensor se caracteriza por su linealidad en todo el rango de trabajo, este comportamiento también se aplica cuando se utiliza con un conversor A/D de 0

a 10 v . Resumiendo, la salida de esta configuración es **directamente proporcional a la presión aplicada al sensor, que a su vez es directamente proporcional a la altura del liquido en el tanque de la unidad**, esto hace que la característica de la planta en esta parte sea lineal.

El fabricante nos brinda información con respecto a esta linealidad y nos entrega la función de transferencia inicial para llegar a la función de transferencia necesaria para representar al sensor en este proyecto.

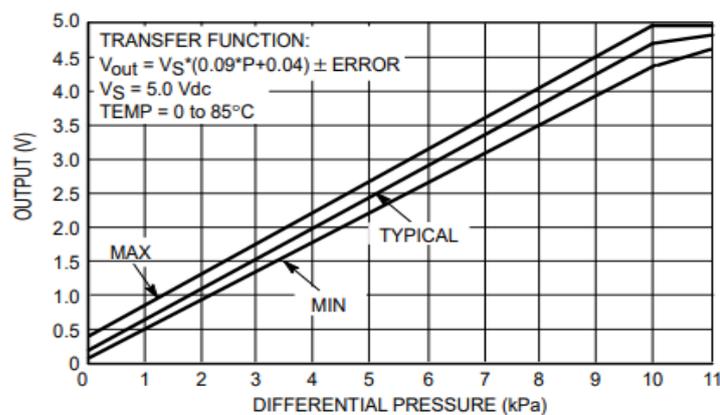


Figura 4.16: Salida vs presión diferencial. [34]

$$V_{out} = V_S * (0.09 * P + 0.04) + \text{ERROR} \quad (4.30)$$

Donde:

- V_{out} = Voltaje de salida. [V]
- V_S = Voltaje de alimentación. [5 V]
- P = Presión diferencial. [KPa]

Recordando que 10KPa = 102 cm de agua. La función que describe el comportamiento del sensor en este proyecto es la siguiente:

$$V_{out} = K_{out} * h \quad (4.31)$$

Donde:

- V_{out} = Voltaje de salida. [V]

- K_{out} = Pendiente de la recta = $5 \times 0.09 = 0.45 \frac{V}{KPa} \times \frac{1KPa}{10.2cm} = 0.044 \frac{V}{cm}$.
- h = Altura del líquido. [cm]

Aplicando la transformada de Laplace a la ecuación (4.31) obtenemos la función de transferencia de nuestro sensor.

$$G_{sensor} = \frac{V_{out}(s)}{H(s)} = 0.044$$

Como sabemos que la parte teórica con la practica difiere un poco a partir de ciertos factores externos. Se caracterizó el sensor para este proyecto de la siguiente manera.

El procedimiento consistió en llenar el tanque en ciertos niveles de altura y tomar el valor de tensión que brinda el sensor, para así relacionar las variables, altura en cm y voltaje.

Altura [cm]	Voltaje de salida [v]
0	0.27
2	0.31
5	0.45
8	0.58
10	0.67
12	0.75
15	0.88
18	1.01
20	1.09
22	1.18
23	1.22
24	1.26
25	1.31

Tabla 4.1: Caracterización del sensor MPX5010.

Una vez completada la tabla anterior, hizo usó de una operación matemática para encontrar la relación entre ambas variables. Esta operación matemática recibe el nombre de **Regresión Lineal**, obteniéndose la siguiente función de transferencia.

$$G_{sensor} = \frac{V_{out}(s)}{H(s)} = 0.044 + \frac{0.2174}{H(s)} \quad (4.32)$$

4.3. Sensor de ultrasonido HC-SR04

Otro tipo de sensor utilizado en este trabajo es el sensor de ultrasonido HC-SR04. Se eligió este tipo de sensor (digital) para contrastar con el anterior (analógicos) y así estudiar ambos tipos.



Figura 4.17: Sensor ultrasónico HC- SR04. [35]

4.3.1. Características del sensor

EL HC-SR04 es un sensor digital ultrasónico útil para la medición de distancias y detección de obstáculos.

- Salida Digital.
- Rango de distancia: 2 cm a 400 cm aprox.
- Resolución: 3 mm
- Frecuencia central: 40 kHz
- Ángulo efectivo: $< 15^\circ$
- Ángulo de medición: 30°
- Pulso de disparo: Nivel alto, $> 10 \mu s$, TTL
- Tiempo recomendado para dar un nuevo disparo: $\geq 60 \text{ ms}$.
- Voltaje de alimentación: 5 V DC

- Corriente en reposo: < 2 mA
- Corriente en operación: 15 mA
- Dimensiones: Largo 45 mm, ancho 20 mm, grosor 15 mm.

4.3.2. Función de transferencia

Este sensor se encuentra ubicado en el tanque n°1, por lo que es útil para realizar un control de primer orden. Para ello, es necesario encontrar la función de transferencia que relacione el volumen depositado de agua con la distancia (o centímetros) que brinda el sensor.

Volumen [cm^3]	Altura [cm]
0	30
350	28
700	26
1050	24
1400	22
1750	20
2100	18
2450	16
2800	14
3150	12
3500	10
3850	8
4200	6
4550	4
4900	2

Tabla 4.2: Caracterización del sensor HC-SR04.

Como en el caso anterior, se usó la operación matemática regresión lineal para encontrar la relación entre ambas variables. Se obtuvo la siguiente función de transferencia.

$$G_{sensorHC-SR04} = \frac{H(s)}{Volumen(s)} = -0.005714 + \frac{30}{Volumen(s)} \quad (4.33)$$

4.4. Electroválvula

Una de las dificultades que presentó este trabajo, fue el armado de una electroválvula con materiales de bajo costo. En el mercado existen una amplia gama de éstas, pero su costo es muy elevado para adquirirlo, por lo que se diseñó una a partir de componentes que se encuentra en el mercado local.

4.4.1. Diseño y Construcción

El diseño resultó bastante complejo con respecto al armado de la parte mecánica de los componentes que se involucran en esta electroválvula.

En su primera instancia se utilizó un servo motor MG995, pero el torque no fue suficiente para hacer girar la válvula de agua. Por lo tanto, surgió el primer inconveniente a tener en cuenta, **el torque del motor**. Para poder solucionar este problema se utilizó un sistema de engranajes, es decir una relación matemática, para obtener mayor torque con respecto a la cantidad de dientes de un engranaje.

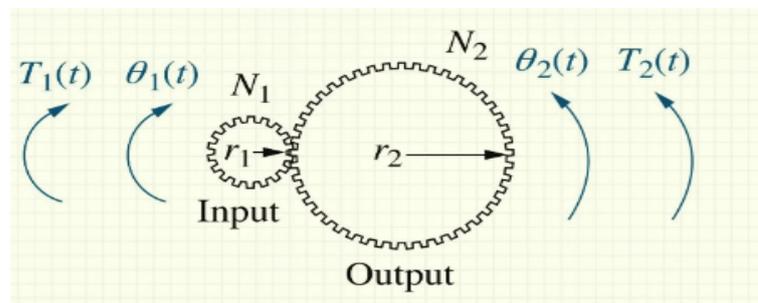


Figura 4.18: Variables en un sistema de engranajes. [32]

La relación del ángulo de giro con respecto al radio de cada engranaje está dado por:

$$r_1\theta_1 = r_2\theta_2 \quad (4.34)$$

$$\frac{\theta_2}{\theta_1} = \frac{r_1}{r_2} = \frac{N_1}{N_2} \quad (4.35)$$

Donde :

- θ = Ángulo de giro
- r = Radio

- N = Número de dientes

Una vez estudiado esto, se relaciona el torque de la siguiente manera:

$$\frac{T_2}{T_1} = \frac{\theta_1}{\theta_2} = \frac{N_2}{N_1} \quad (4.36)$$

En donde concluimos a partir de la ecuación anterior que, para obtener mayor torque en el engranaje de salida, es decir donde se quiere posicionar la válvula, es necesario tener un mayor número de dientes en su engranaje¹.

Una vez montado este sistema de engranajes en el servo motor, se observó que **no** se obtuvo el movimiento deseado de cierre y apertura de la válvula. Por lo tanto, se procedió a cambiar de motor y realizar una etapa de control interna para la electroválvula. Previamente se realizó pruebas con el sistema de engranajes acoplado al nuevo motor pero tampoco alcanzó las expectativas deseadas.

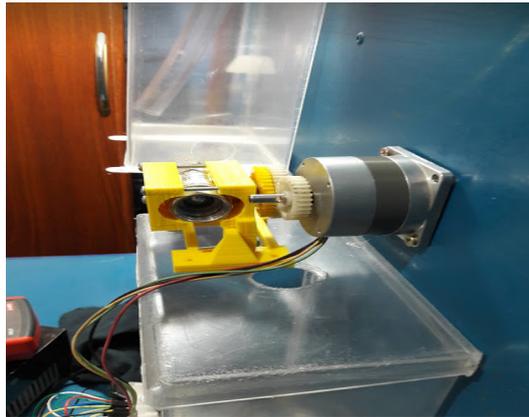


Figura 4.19: Motor acoplado al sistema de engranajes. [32]

¹Los engranajes fueron realizados a través de una impresora 3D.

Por esta razón, se consideró acoplar directamente el eje del motor con la parte móvil de la válvula, esto se observa en la figura 4.20.



Figura 4.20: Eje del motor acoplado a válvula. [32]

Con esta disposición, se hizo necesario realizar un sistema de control (controlador secundario) propio de la válvula para controlar el cierre y apertura de la misma. Este procedimiento se describe en subsecciones posteriores.

4.4.2. Características del motor

Motor Stepper STH - 55D121

Este motor pertenece a la familia SKN Shinano Kenshi Corporation de motores paso a paso de precisión. Es un motor híbrido¹ de 4 hilos, cuenta con un ángulo de paso de 1.8°, lo que permite una mayor precisión a la hora de realizar el control.

Las características más importante que se tuvieron en cuenta para este proyecto fueron, su torque de 2.8 kg-cm, su tamaño de 2.3”, y su peso de 360 g.

¹Combinación de un motor de reluctancia variable y el motor de magnetización permanente.

4.4.3. Etapa de potencia

Como el motor anterior necesita un mayor voltaje que el proveniente del microcontrolador, es necesario una etapa de potencia para que funcione adecuadamente y sin que produzca fallas por sobrecargas en la etapa de control del sistema.

Para ello, se hizo uso del integrado L293D, conocido también como doble puente H. Este dispositivo permite altas corrientes para el manejo de 4 canales, diseñado para aceptar tensiones estandar TTL y el manejo de cargas inductivas, como son relays y motores paso a paso.

Cuenta con 2 entradas de habilitación, encargadas de activar o desactivar el dispositivo independiente de las señales de entrada. Además contiene 2 fuentes de alimentación, siendo una para los niveles lógicos, de este modo la parte lógica trabaja a un bajo voltaje sin verse afectada por la alimentación principal de la carga. [38]

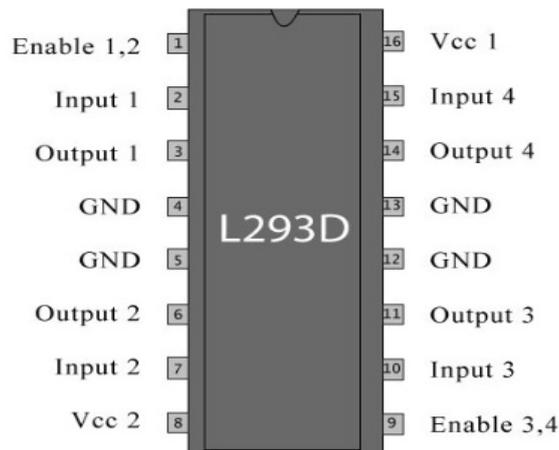


Figura 4.21: Conexiones de pines. [38]

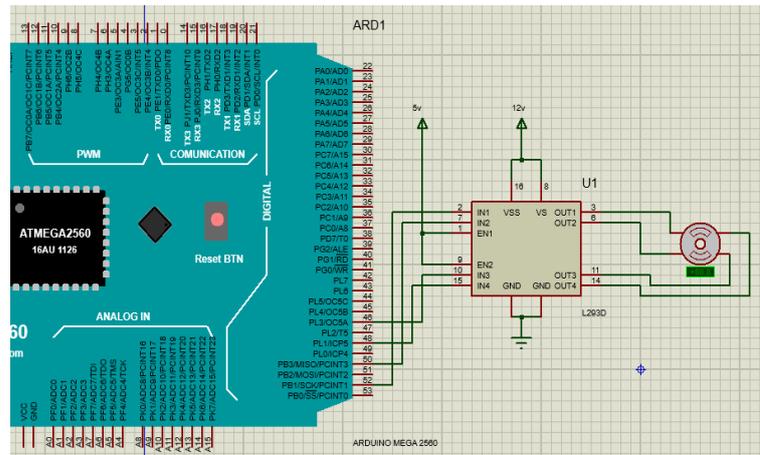


Figura 4.22: Simulación L293D. [32]

4.4.4. Diagrama de bloques

La fig. 4.23, muestra el diagrama de bloques de la electroválvula realizada. Podemos observar que en el lazo principal se encuentra el controlador junto con la planta, mientras que en el lazo realimentado se encuentra el bloque del sensor, a continuación se detallan cada uno.

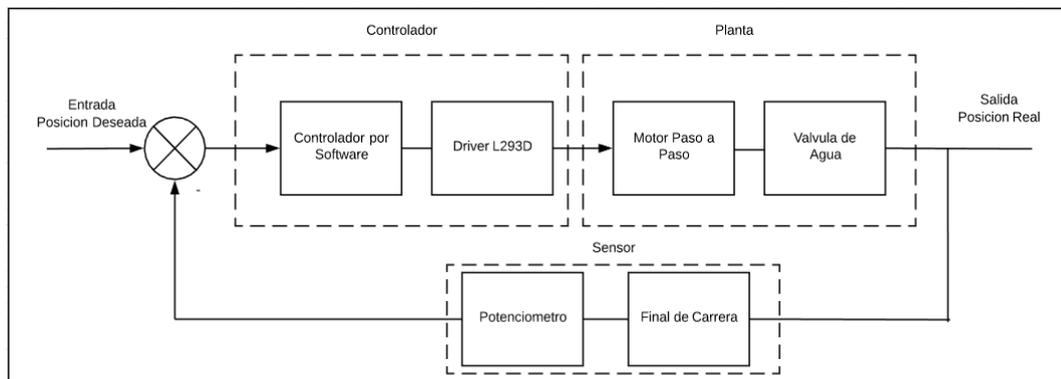


Figura 4.23: Diagrama de bloques de la electroválvula . [32]

Controlador

En este bloque se encuentran **el controlador realizado a través del software** (programación mediante una librería que controla la velocidad y el número de pasos aplicado al motor), y **el integrado L293D**, si bien podemos pensarlo como el actuador del sistema, lo ponemos en la categoría de controlador, ya que posee señales de activación y/o desactivación hacia los terminales del motor.

Planta

La planta de este sistema esta conformado por la electroválvula en si, es decir, el motor stepper acoplado a la válvula de agua.

Sensor

En el lazo de realimentación, se encuentra el bloque del sensor. Al principio estaba formado por un **Switch Final de Carrera**, el cual nos permitía setear la posición '0', es decir, donde la electroválvula se encuentra totalmente cerrada y establecerla como el valor de referencia. Debido a darle un mayor uso al **Potenciómetro**, se decidió que sea éste elemento el que tome la referencia del '0'. así mismo, otra función que presenta es la de obtener la posición real donde se encuentra la electroválvula, donde el valor (posición) es ingresado a través del conversor A/D del controlador para su procesamiento¹

4.4.5. Función de transferencia

Para caracterizar la electroválvula, se relacionaron las variables, tensión de entrada versus el caudal de salida. El proceso consiste en aplicar un voltaje inicial 1V, variando en 0.5V hasta llegar a un voltaje de 5V. En este punto cabe hacer la aclaración, que a tratarse de una electroválvula de fabricación propia el voltaje aplicado influye directamente en el grado de apertura de la válvula, controlando esto mismo a través del microcontrolador principal.

Como no se contaba con un caudalímetro o flujómetro, se utilizaron los tanques propios de la maqueta como volúmenes de control para realizar las mediciones.

El volumen de control del tanque n° 1 fue:

$$V_1 = 10 \text{ cm} \times 25 \text{ cm} \times 17.5 \text{ cm} = 4.375 \text{ L.}$$

El tanque n° 2 presentó un volumen de control de:

$$V_2 = 10 \text{ cm} \times 10 \text{ cm} \times 17.5 \text{ cm} = 1.75 \text{ L.}$$

El procedimiento se inició cargando el tanque n° 1 con agua hasta que alcanzaba un nivel de 25 cm. Luego se variaba la tensión de entrada a la electroválvula y se registraba el **tiempo** que tardaba en llenarse el tanque de control n° 2 a un nivel de 10 cm. Para cada tensión aplicada se registraron 3 mediciones con tiempos diferentes,

¹Para mayor detalles ir a la sección Código de Programación.

las cuales fueron promediadas con el fin de buscar una medida de tiempo más precisa. Esto se observa en la tabla 4.3.

Voltaje de entrada [V]	t ₁ [s]	t ₂ [s]	t ₃ [s]	t _{prom}	Q= $\frac{Volumen[cm^3]}{t[s]}$
1	79	81	78	79.3	22.05
1.5	24.09	24.36	24.32	24.25	72.16
2	19.61	20.96	20.20	20.25	86.41
2.5	17.9	17.29	17.32	17.53	99.82
3	16.63	16.55	16.89	16.69	104.85
3.5	16.10	16.14	15.82	16.08	108.83
4	15.97	16.01	16.04	16.00	109.37
4.5	16.03	15.52	16.04	15.86	110.34
5	15.52	15.66	15.80	15.79	110.82

Tabla 4.3: Caracterización de electroválvula.

Una de las formas que se utilizó en este trabajo para encontrar una función de transferencia que mejor describa el comportamiento de la electroválvula fue transcribiendo los datos obtenidos en 2 vectores haciendo uso de la herramienta **PID Tuner** de Matlab. Esta herramienta interactiva resulta muy útil a la hora de identificar sistemas, sin embargo no sustituye a los conocimientos adquiridos por un ingeniero pero sumados a estos, hace que los conocimientos sean afianzados más fácilmente.

Se analizaron diferentes modelos matemáticos, los cuales pueden visualizarse en las siguientes gráficas. Si bien esta herramienta resulta de buena utilidad para encontrar funciones de transferencia, finalmente se utilizó el método de mínimos cuadrados para obtener una respuesta puramente lineal.

Este método se utiliza para encontrar la mejor recta que se aproxime a un conjunto de datos dados y permite estudiar la naturaleza entre dos variables. Conceptualmente el proceso de ajustar los datos experimentales a una función lineal se denomina **Regresión Lineal**.

$$m = \frac{\Sigma XY - \frac{(\Sigma X)(\Sigma Y)}{n}}{\Sigma X^2 - \frac{(\Sigma X)^2}{n}}$$

Donde:

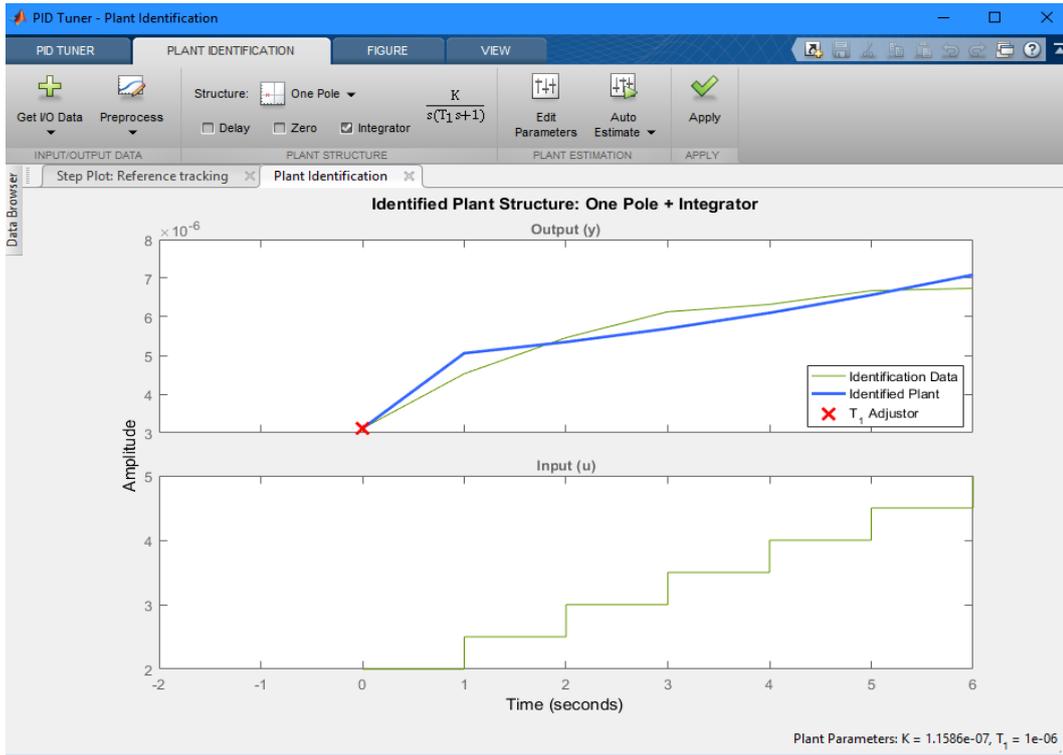


Figura 4.24: Modelo con un polo real más integrador. [32]

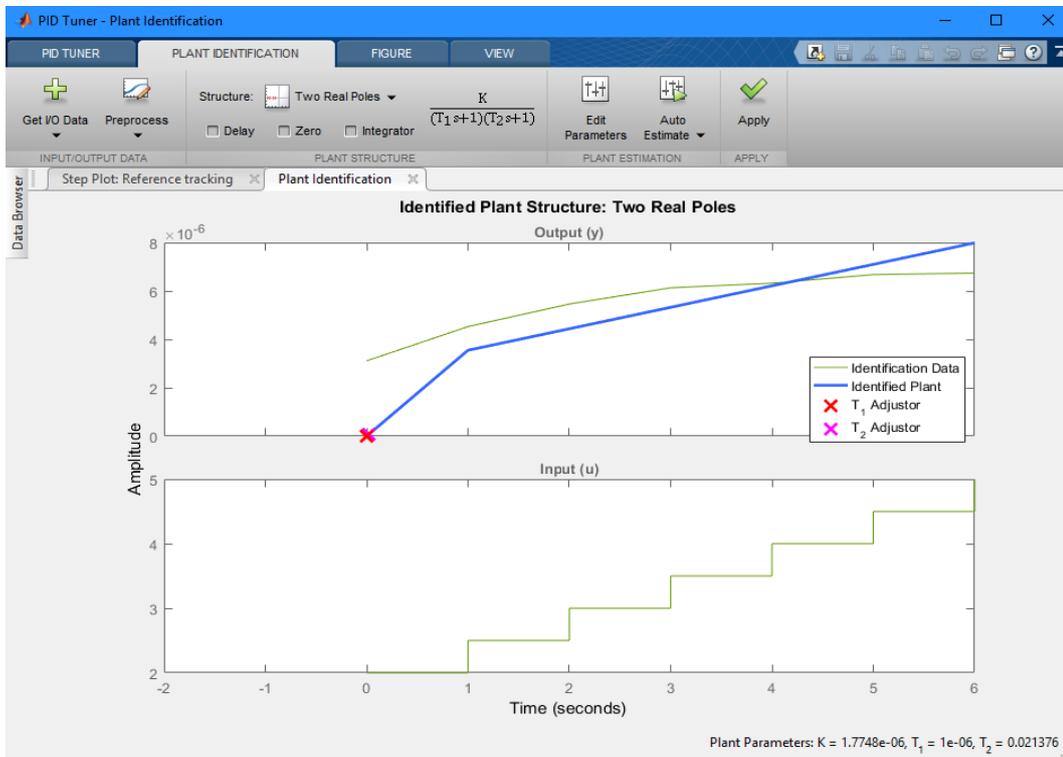


Figura 4.25: Modelo con dos polos reales. [32]

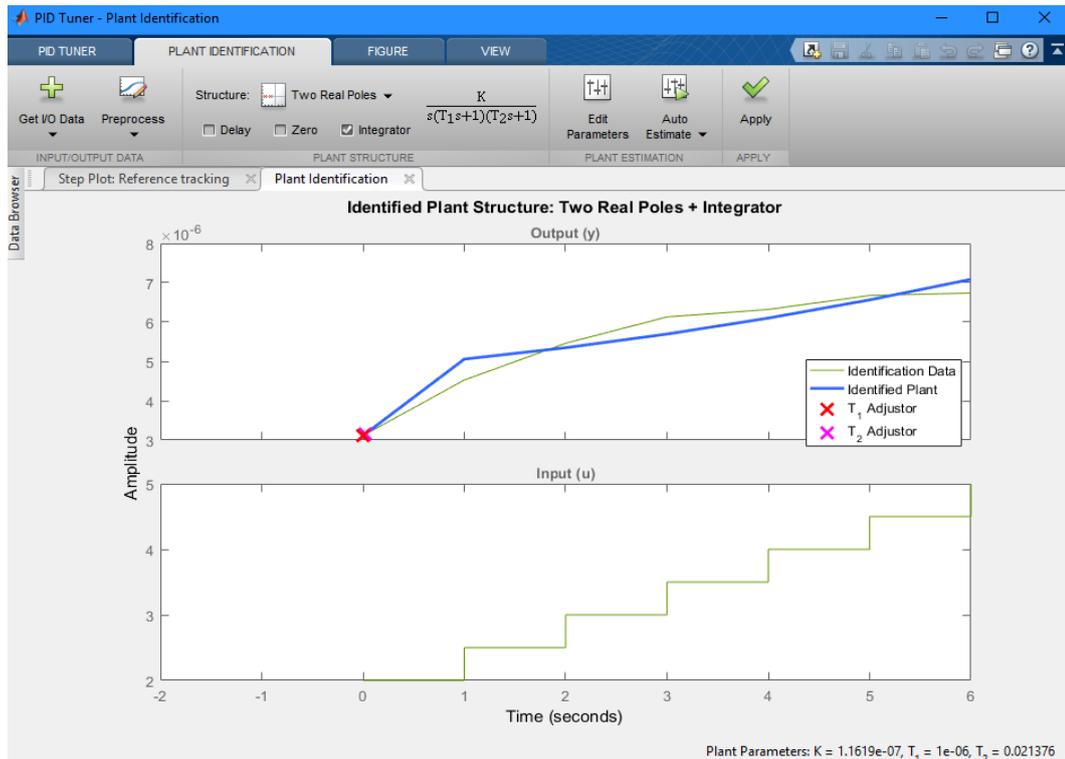


Figura 4.26: Modelo con dos polos reales más integrador. [32]

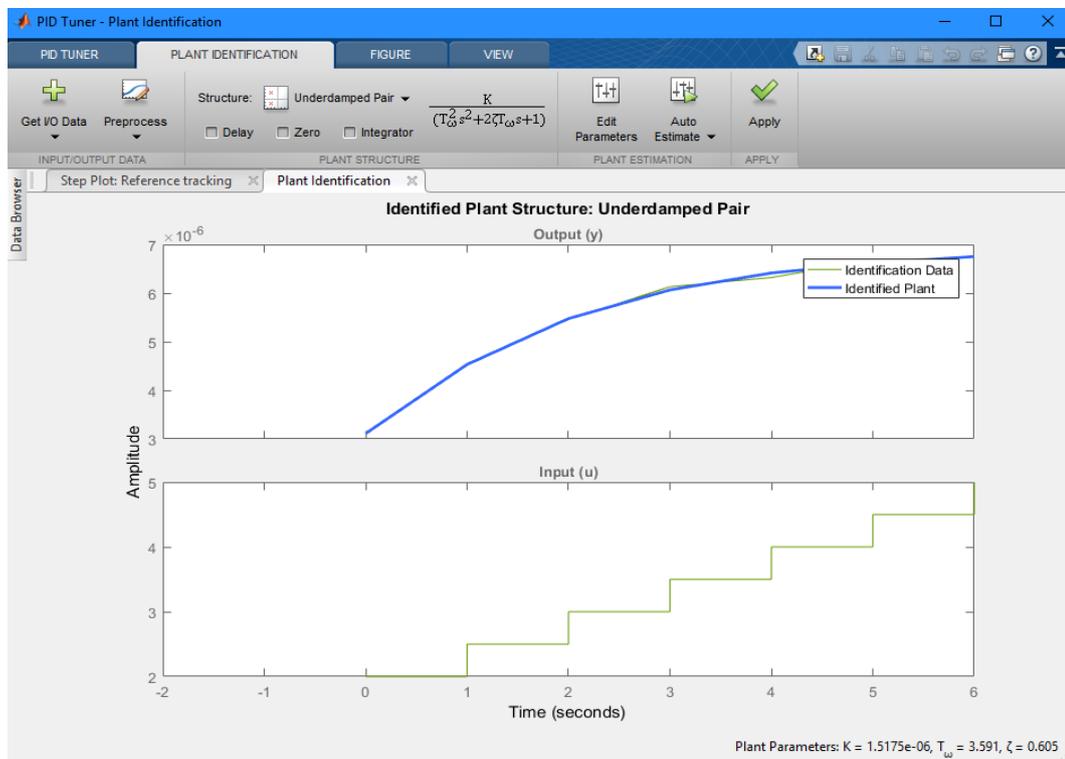


Figura 4.27: Modelo con un par de polos complejos conjugados. [32]

- $X=V=$ Voltaje de entrada [V]
- $Y=Q=$ Caudal de salida [m^3/s]
- $n=$ Número de muestra.

De la tabla 4.3 se obtienen los siguientes valores:

$$\begin{aligned}\Sigma V &= 24.5 \\ \Sigma Q &= 824.65 \\ \Sigma VQ &= 2736.22 \\ \Sigma V^2 &= 92.75\end{aligned}$$

Entonces la pendiente m de la curva es:

$$m = 18.85 \tag{4.37}$$

Para conocer la intersección en Q se realizan los siguientes cálculos:

$$\bar{x} = \frac{\Sigma x}{n} \quad \bar{y} = \frac{\Sigma y}{n}$$

$$b = \bar{y} - m\bar{x} = 40.34 \tag{4.38}$$

Resultando entonces, la recta que más se aproxima a la curva generada por los datos experimentales de la electroválvula, como:

$$Q = 18.85.V + 40.34 \tag{4.39}$$

En la figura 4.28 se observa la respuesta de la electroválvula obtenida a partir de las mediciones y su linealización por regresión lineal. Como manera de corroborar estos datos y que la obtención de la recta sea la que más se aproxima a la curva, se comparó con la herramienta **Curve Fitting Tool**, donde se obtuvo una recta de valores similares, como lo muestra la figura 4.29.

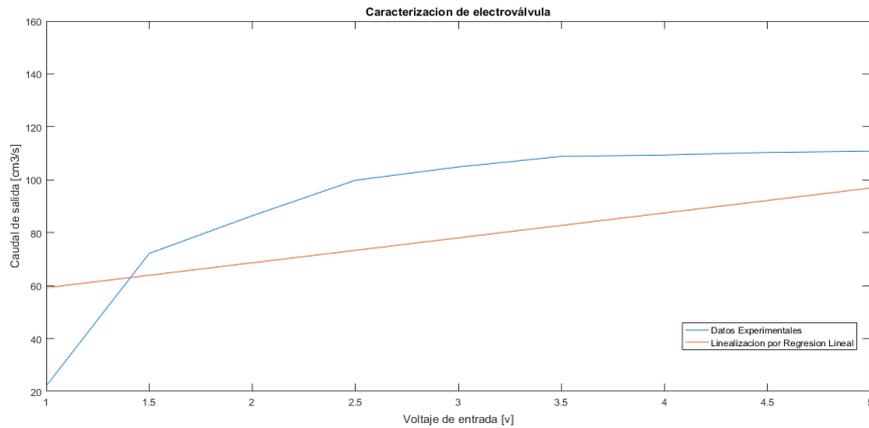


Figura 4.28: Caracterización electroválvula. [32]

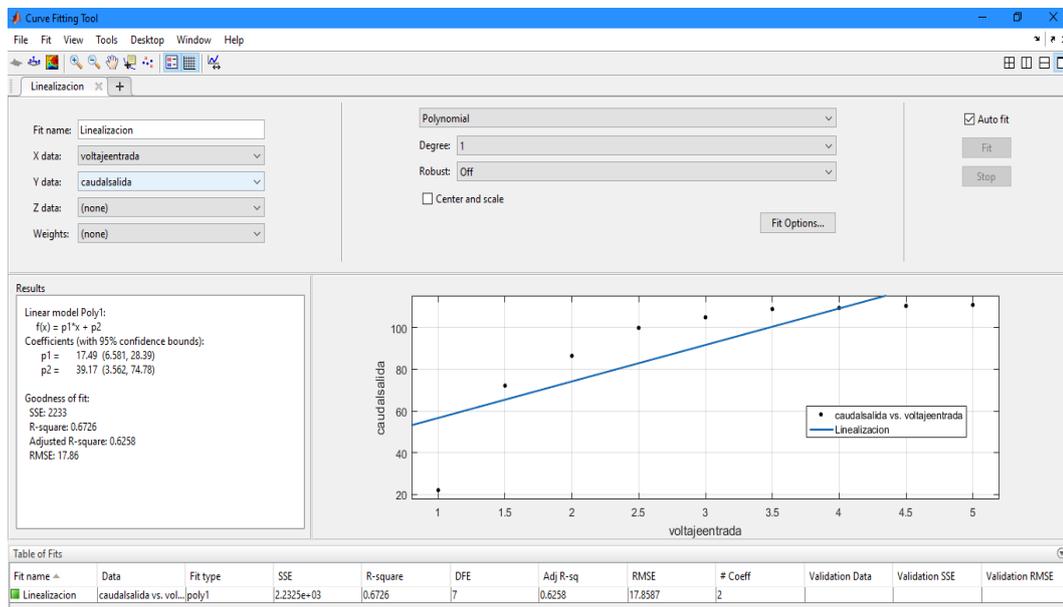


Figura 4.29: Herramienta Curve Fitting Tool . [32]

Función de transferencia

Finalmente si se aplica la transformada de Laplace a la ecuación (4.39) y se convierte a cm^3/s para simplificar los cálculos, obtenemos la siguiente función de transferencia para la electroválvula:

$$G_{elec}(s) = \frac{Q(s)}{V(s)} = 18.85 + \frac{40.34}{V(s)} \quad (4.40)$$

4.5. Bomba Periférica

Para el ingreso del caudal de agua en los tanques donde se realiza el control, se utilizó una bomba de 1/2 hp que toma el agua de un tanque reservorio, ubicado en la parte inferior de la unidad, la cual transporta el agua a través de una cañería propia del sistema, de manera que el tanque reservorio se encuentra realimentado con los desagotes de los tanques acoplados para el control.



Figura 4.30: Bomba Marca Plumita . [39]

4.5.1. Características del motor

- Potencia: $1/2 \text{ HP} = 0.37 \text{ KW}$.
- Peso: 3.8 Kg.
- Temperatura ambiente máxima: $80 \text{ }^\circ\text{C}$.
- Velocidad n_o : 2850/min.
- Caudal : 35 L/min.
- Altura manométrica : 20 m .
- Temperatura de agua máxima: 40°C

4.5.2. Etapa de potencia y acondicionamiento de señal

Al tratarse de un equipamiento que trabaja con una tensión de 220 AC, para tener un control indirectamente del caudal de entrada hacia los tanques fue necesario realizar el controlador de un motor AC descrito en la sección 3.6, juntamente con el relé de estado solido, diseñado para poder aislar la etapa de potencia de corriente alterna con la etapa de control brindada por el microcontrolador¹.

Este tipo de controlador (disparo de un triac), nos permite variar con un potenciómetro indirectamente el caudal de agua que ingresa a los tanques. Lo que hace útil para poder realizar perturbaciones y analizar diferentes formas de control del sistema en general. Además tiene la función de actuar como medida de seguridad, ya que permite controlar las posibles fugas y reboses de agua en los tanques, debido a la fuerza con que sale ésta de la bomba.

4.6. Fuente de Alimentación

Para este trabajo, se diseñó una fuente de alimentación regulable propia a partir de componentes pasivos y activos encontrados en el mercado local. Cabe mencionar que se realizó este diseño para optimizar el espacio y con el fin de tener una fuente de alimentación propia para la electroválvula realizada, y así los actuadores y módulos presentes en la unidad funcionen de manera independiente alimentados a través de una fuente ATX.

4.6.1. Características Fuente:

- Regulada variable.
- Tensión de entrada: 220v AC / 24v AC.
- Tensión de salida regulable: 1.2 a 30v DC.
- Tensión de salida fija: 5v DC.
- Corriente máxima de salida : 3 A.

Para esta fuente se utilizó el regulador variable LM317. Este regulador entrega una corriente de hasta 1.5 amperios y voltajes entre 1.2 voltios hasta 32 voltios.

¹El diseño se encuentra en el apéndice B - Diseño del PCB.

La línea de carga y la regulación es de mejor calidad que la de los reguladores fijos por ser un integrado más moderno. Está protegido contra las limitaciones de corriente, exceso de temperatura y por sobrecarga, así el terminal de regulación esté desconectado.

Como la fuente rectificadora está en el mismo integrado, no necesita muchos capacitores. A este tipo de reguladores se los llama flotante, esto quiere decir que el regulador solo ve la diferencia entre la entrada y la salida del voltaje. Esto permite utilizarlos para regular alto voltaje, siempre y cuando no supere más de 30 voltios la diferencia entre entrada/salida.

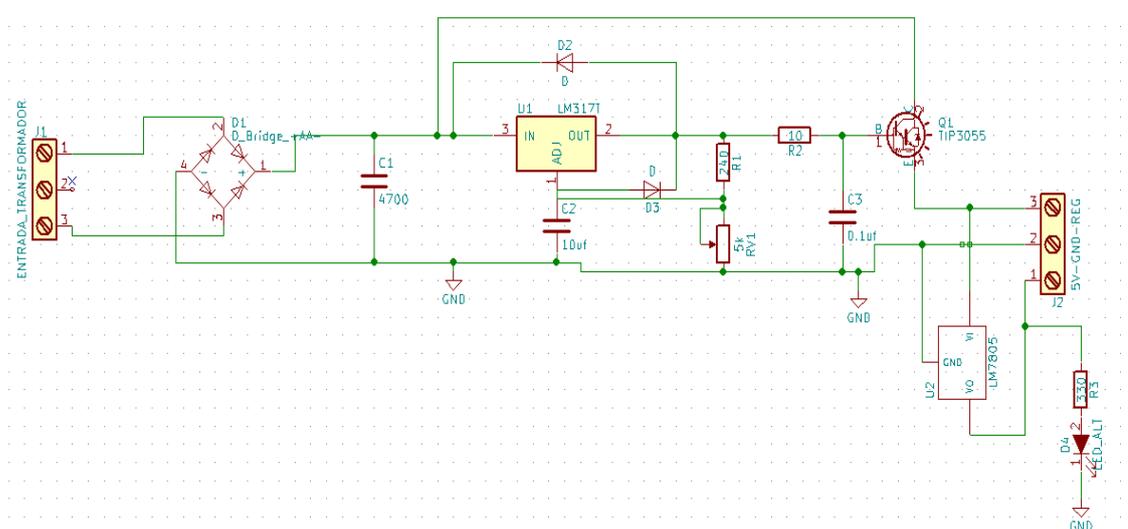


Figura 4.31: Diseño esquemático Fuente Regulada. [32]

La función del puente de diodos es la de invertir todos los ciclos negativos a ciclos positivos, desapareciendo la tensión negativa. El puente entrega los ciclos positivos a través de la pata marcada con un (+), y la pata de la esquina contraria, será tierra (GND) o negativo. El puente de diodos va acompañado de un capacitor que rectifica la corriente, que el puente de diodos entrega, pero que aún no ha convertido totalmente en corriente continua. El capacitor se carga y no deja que la corriente baje a cero, manteniéndola en un mismo voltaje. El valor de este capacitor es de 4700 microfaradios.

Como el regulador LM317 entrega 1 amperio en condiciones normales de trabajo y hasta 1.5 amperios forzando el regulador, se implementó un transistor de potencia TIP3055, para aumentar el manejo de corriente a 3 amperios.

Esta fuente nos permite obtener 2 tensiones, una regulada, la cual es utilizada solo para alimentar el driver L293D, encargado de controlar el motor paso a paso acoplado

a la válvula de agua, con una tensión mayor a 15 v, y una tensión fija proveniente del regulador LM7805 para la alimentación de los módulos y actuadores extras a la unidad, tales como Bluetooth, caudalímetro, entre otros.

4.6.2. Diseño PCB

Los diagramas esquemáticos y el circuito impreso fueron diseñados mediante el software **Kicad Designer** versión 2015. Se eligió trabajar con este software libre porque es la herramienta que utilizan los diseñadores de PCB profesionales, además posee soluciones de diseño que son eficientes y fáciles de usar, como el visualizador 3D del modelo creado, y la verificación automática de las reglas de diseño. Esto permite que la probabilidad de éxito del diseño sea alta.

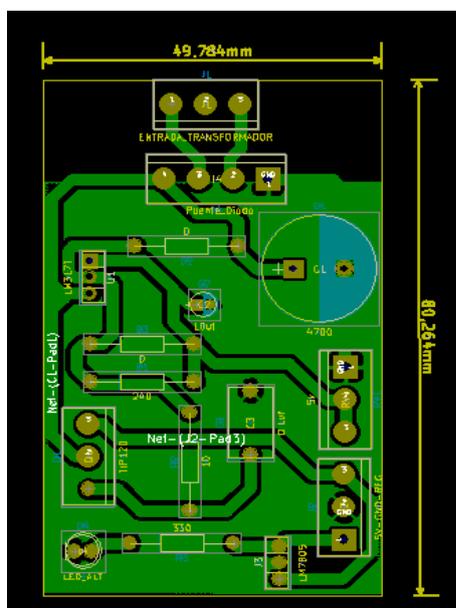


Figura 4.32: Fuente diseñada en Kicad, vista en 2D. [32]

4.6.3. Planos de Masa

Un plano de masa es un área grande de cobre conectada al punto de tierra del circuito (por lo general un terminal de la fuente de alimentación), y sirve como camino de retorno para la corriente de muchos de los componentes, pero también asegura que la conexión a tierra de todos los componentes estén en el mismo potencial de referencia. En lo general un plano de tierra se hace tan grande como sea posible, para que cubra la mayor parte del área del PCB que no este ocupado por las pistas de circuito, lo cual permite poner a tierra cualquier componente sin tener que ejecutar

pistas adicionales. La razón principal por la cual se usan grandes planos de tierra, es para reducir el ruido eléctrico y la interferencia a través de lazos de tierra, y también para evitar interferencias entre las pistas adyacentes del circuito. Cuando los circuitos digitales cambian de estado, grandes impulsos de corriente fluyen de los dispositivos activos (transistores o circuitos integrados) a través del plano de tierra. La amplia zona del plano de masa tiene una impedancia muy inferior a una pista de circuito, por lo que los impulsos de corriente causan menos perturbación.

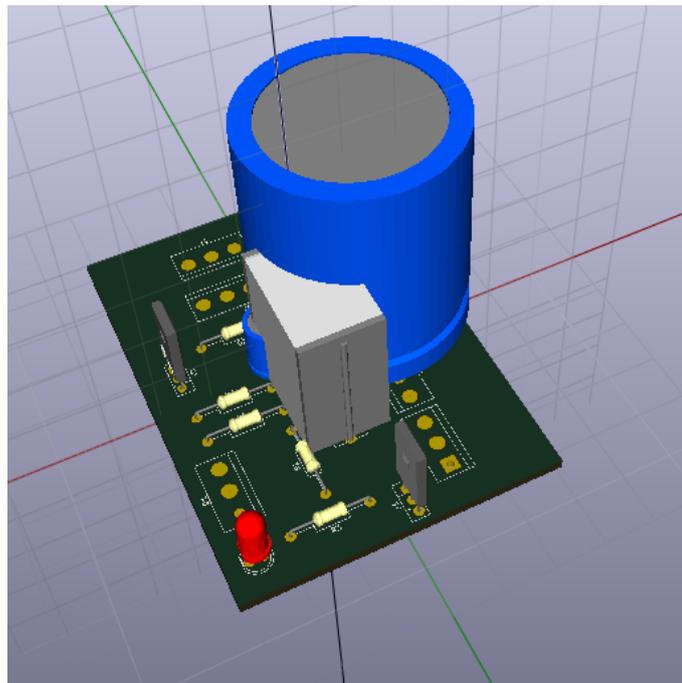


Figura 4.33: Fuente diseñada en Kicad, vista en 3D. [32]

4.7. Sistema de Control de Nivel

4.7.1. Diagrama de bloques

El esquema de bloques simplificado del sistema se muestra en la Fig. 4.34, donde podemos observar en el lazo principal el ingreso de la señal de referencia del nivel y la señal del nivel medido, cuya diferencia entre ambas nos brindan la señal de error del sistema, que es la señal trabajada. Seguidamente en cascada se encuentra el bloque del controlador PID, cuya función principal es la de compensar el sistema para obtener una buena respuesta tanto transitoria como en estado estable.

La señal proveniente del controlador ingresa al bloque actuador, en este caso al

drive (descrito en la sección anterior), que controla la apertura de la electroválvula diseñada, variando el caudal de entrada a la planta del sistema que corresponde al tanque.

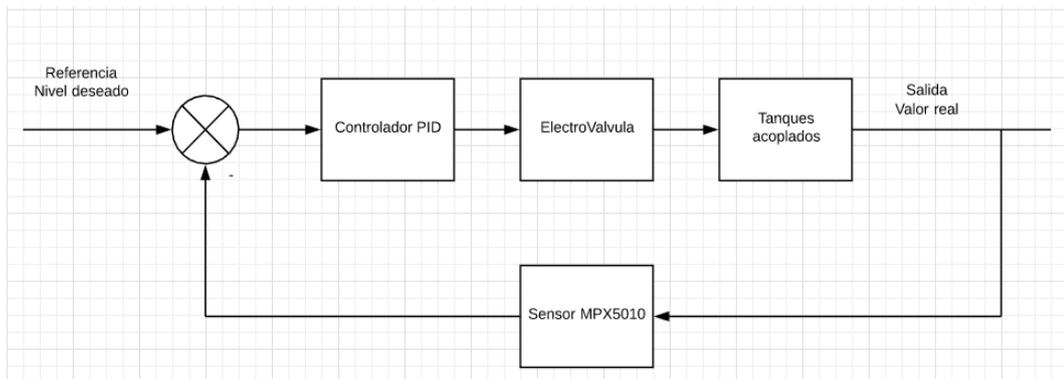


Figura 4.34: Diagrama de bloques del sistema. [32]

La señal de salida de la planta, que para este trabajo es el nivel o altura del líquido en el interior del tanque, es sensada, como se describió en secciones anteriores, de manera indirecta a través del sensor de presión, luego realimentada a través del lazo de realimentación al lazo principal para su comparación con el valor de referencia.

4.7.2. Diseño de control

Para realizar el control a la planta, es necesario obtener primeramente la función de transferencia completa. Esta se encuentra formada por los tres bloques principales correspondiente cada uno a:

- Electroválvula
- Sistema de tanques acoplados
- Sensor de nivel

En subsecciones anteriores se describieron particularmente cada función de transferencia que hacen a la planta, estas son:

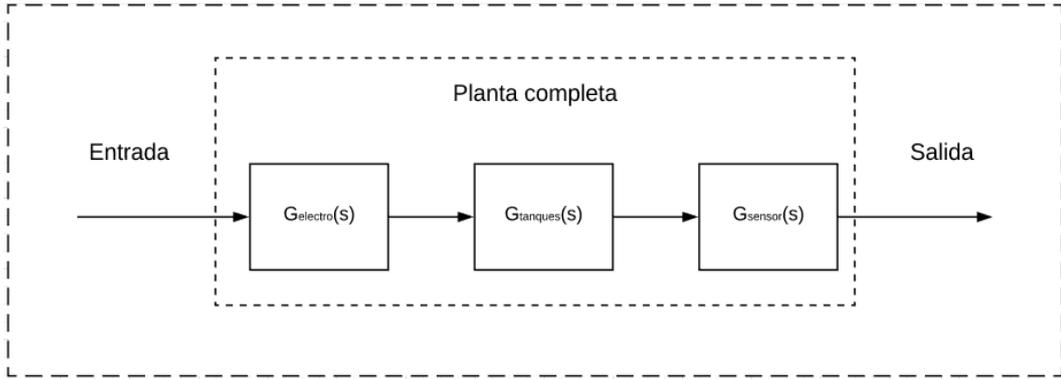


Figura 4.35: Diagrama de bloques de la planta. [32]

* **Función de transferencia de electroválvula:**

$$G_{elec}(s) = \frac{Q(s)}{V(s)} = 18.85 + \frac{40.34}{V(s)} \quad (4.41)$$

* **Función de transferencia de un tanque (1° orden):**

$$\frac{H_1(s)}{Q_s(s)} = \frac{-k_2}{s + k_1} \quad (4.42)$$

Donde:

$$k_1 = \frac{K_v}{2A\sqrt{H}} = \frac{0.69}{2 * 175 * \sqrt{15}} = 0.00051$$

$$k_2 = \frac{1}{A} = \frac{1}{175} = 0.0057$$

$$\frac{H_1(s)}{Q_s(s)} = \frac{-0.0057}{s + 0.00051} \quad (4.43)$$

* **Función de transferencia de tanques acoplados (2° orden):**

$$G_{tanques}(s) = \frac{H_2(s)}{Q_p(s)} = \frac{k_1 k_2}{s^2 + s(k_1 + k_3) + k_1 k_3} \quad (4.44)$$

Donde:

$$k_1 = \frac{K_v}{2A\sqrt{H_1 - H_2}} = \frac{0.69}{2 * 175 * \sqrt{15 - 10}} = 0.00088$$

$$k_2 = \frac{1}{A} = \frac{1}{175} = 0.00571$$

$$k_3 = \frac{K_v}{2A\sqrt{H_2}} = \frac{0.69}{2 * 175 * \sqrt{10}} = 0.00062$$

$$G_{tanques}(s) = \frac{H_2(s)}{Q_p(s)} = \frac{5.01x10^{-6}}{s^2 + 0.0015s + 0.54x10^{-6}} \quad (4.45)$$

*** Función de transferencia del sensor de ultrasonido:**

$$G_{sensor} = \frac{H(s)}{volumen(s)} = 0.005714 \quad (4.46)$$

*** Función de transferencia del sensor de presión:**

$$G_{sensor} = \frac{V_{out}(s)}{H(s)} = 0.044 \quad (4.47)$$

Aplicando el álgebra de diagramas de bloques al diagrama de la figura 4.35, se tiene:

$$G_{planta} = G_{electro}(s) * G_{tanques}(s) * G_{sensor}(s) \quad (4.48)$$

Sistema de primer orden

$$G_{planta} = \frac{-0.0005706}{s + 0.00052} \quad (4.49)$$

Sistema de segundo orden

$$G_{planta} = \frac{0.000003855}{s^2 + 0.0015s + 0.00000043} \quad (4.50)$$

4.7.3. Diseño del controlador PID analógico

En este apartado se expone el diseño de un controlador PID de tipo analógico, para cada uno de los sistemas con los que puede funcionar este trabajo final conforme al orden del mismo.

Sistema de primer orden

Para este sistema se considera un controlador tipo PI, cuya función de transferencia esta dada por:

$$G_{PI}(s) = K_p * [1 + \frac{K_i}{s}] \quad (4.51)$$

Para encontrar las constantes a este controlador se utilizó dos formas, las cuales se detallan a continuación:

- 1) Mediante la técnica de cancelación de polos y ceros.
- 2) Mediante la herramienta PID Tuner de Matlab.

Cancelación de polos y ceros

Partiendo de la ec. (4.49), la función de transferencia de la planta más el compensador estará dada por:

$$G_{planta}(s) = \frac{K_p * [s + K'_i]}{s} * \frac{-0.0004175}{s + 0.0652} \quad (4.52)$$

Se hace la consideración de colocar el polo de lazo cerrado en -0.2 y se aplica la técnica de cancelación de polos y ceros, obteniéndose los siguientes valores para las constantes:

$$s + K'_i = s + 0.00052$$

$$K'_i = 0.00052$$

$$K_p * (-0.0005706) = 0.2$$

$$K_p = -350.50$$

$$K_i = K'_i * K_p = -0.18$$

$$G_{PI}(s) = K_p + \frac{k_i}{s} \quad (4.53)$$

$$G_{PI}(s) = \frac{-350.5s - 0.18}{s} \quad (4.54)$$

Cabe hacer la aclaración que se eligió el polo de lazo cerrado en - 0.2, debido a que éste presenta una buena respuesta frente a otra ubicación de polos. Como motivo de

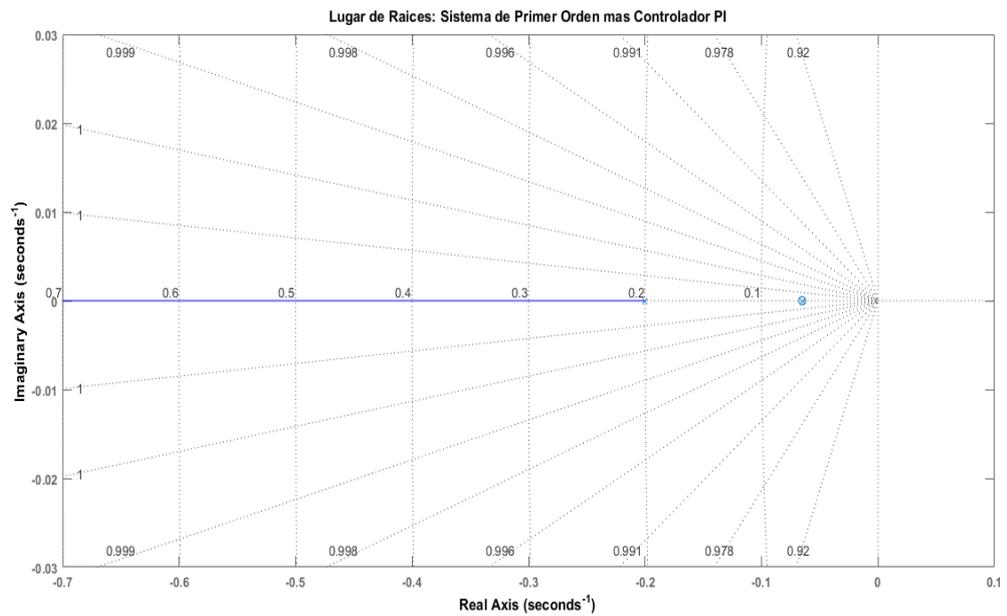


Figura 4.36: Lugar de raíces del sistema más controlador. [32]

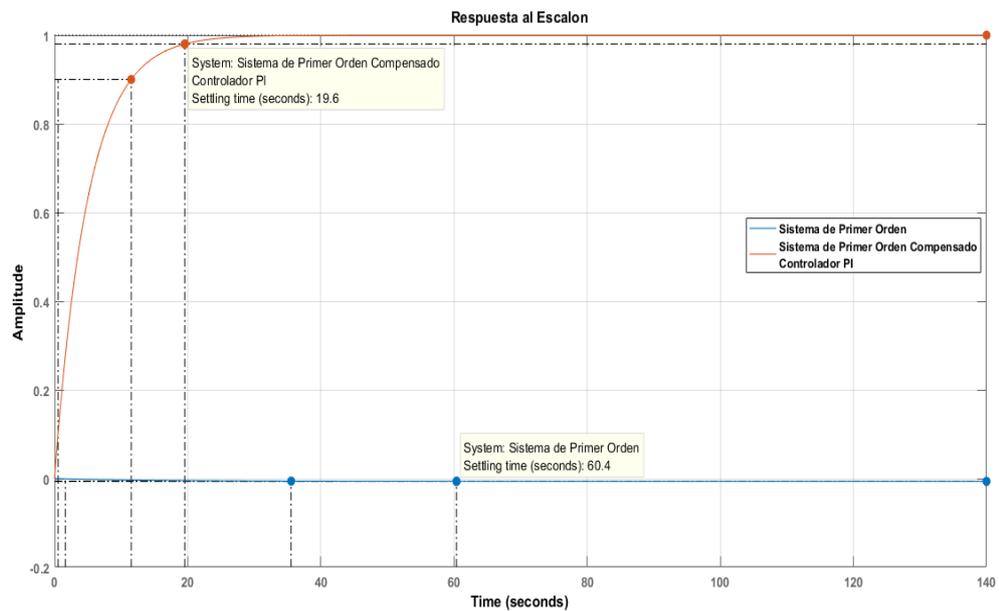


Figura 4.37: Respuesta al sistema de primer orden. [32]

mostrar lo anterior se tomo el polo en -0.1 , cuya diferencia en la respuesta se muestra en la figura 4.38.

El sistema presenta una respuesta sin sobreimpulso con un tiempo de subida de 11

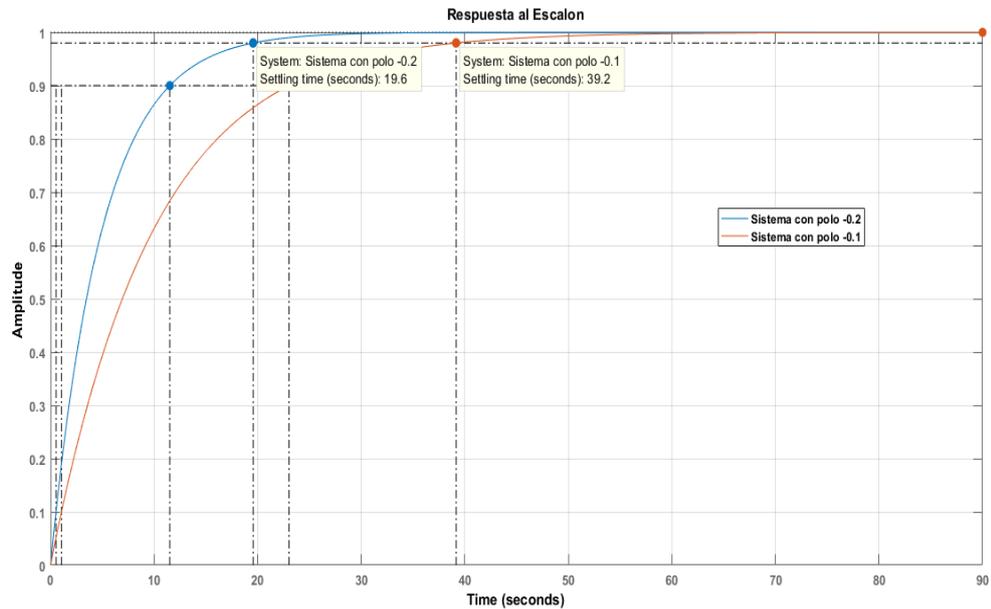


Figura 4.38: Diferencias en respuesta al sistema de primer orden. [32]

segundos, estableciéndose al valor final en 55 segundos, como características salientes del sistema.

Herramienta PID Tuner

Esta herramienta interactiva nos permite obtener un controlador PI a partir de las especificaciones requeridas en nuestro sistema, como por ejemplo tiempo de subida, tiempo de establecimiento, sobreimpulso, entre otras. Es decir, estableciendo parámetros entre rapidez y robustez del controlador. Para este caso se busca que el controlador responda con un tiempo de establecimiento menor a 20 s y sin tener sobreimpulso.

$$G_{PI}(s) = K_p + \frac{k_i}{s} \quad (4.55)$$

$$G_{PI}(s) = \frac{-1528.1s - 99.7}{s} \quad (4.56)$$

La Fig. 4.39, muestra la respuesta del sistema compensado con el controlador PI generado. Se puede observar que :

- Tiempo de subida = 3.44s
- Tiempo de establecimiento = 6.13s

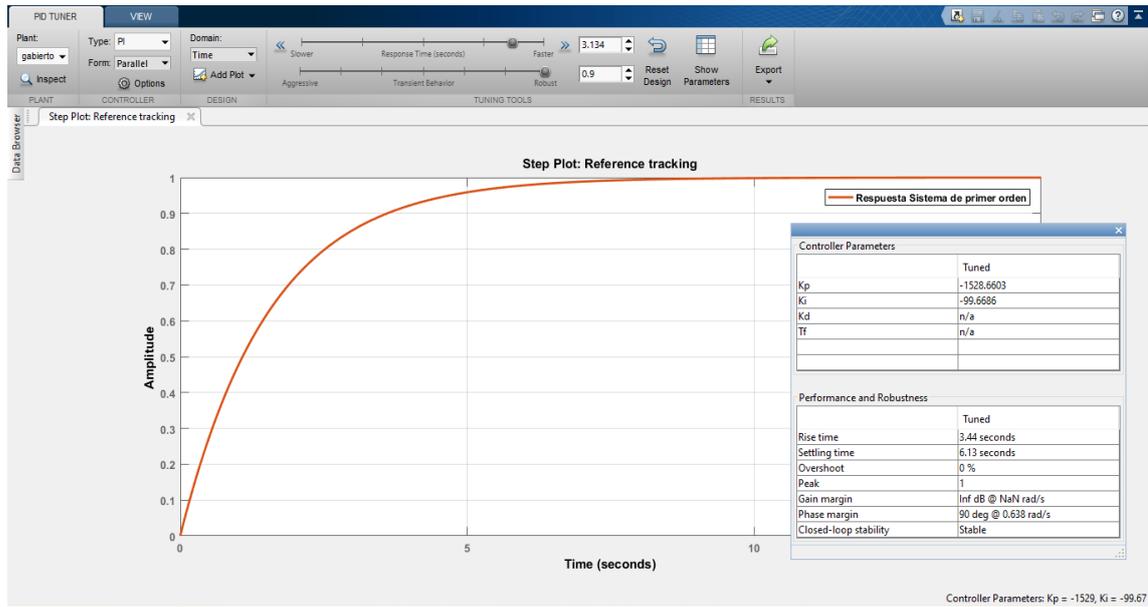


Figura 4.39: Diseño control PI y respuesta del sistema. [32]

- Sobreimpulso = 0 %

El controlador cumple ampliamente con las especificaciones requeridas. Cabe hacer mención que la herramienta utiliza **el mismo método** aplicado analíticamente líneas arribas (cancelación de polos y ceros), ubicando el polo de lazo cerrado en -0.638.

Sistema de segundo orden

Para este sistema se considera un controlador PID, cuya función de transferencia esta dada por la forma:

$$G_{PI}(s) = K_p * \left[1 + K_d s + \frac{K_i}{s} \right] \quad (4.57)$$

Al igual que en el caso del sistema de primer orden, se diseñó el controlador PID de 2 maneras.

Cancelación de polos y ceros

Partiendo de la ec. (4.50), la función de transferencia de la planta más el compen-sador estará dada por:

$$G_{planta}(s) = K_p * \left[\frac{K'_d s^2 + s + K'_i}{s} \right] * \frac{0.00000385}{s^2 + 0.0015s + 0.00000309} \quad (4.58)$$

$$K'_d = \frac{1}{0.0015} = 666.66$$

$$K'_i = \frac{0.00000309}{0.0015} = 0.00206$$

$$K_p * \frac{0.00000385}{0.0015} = 0.2 \Rightarrow K_p = 77.92$$

$$K_d = K'_d * K_p = 51946$$

$$K_i = K'_i * K_p = 0.1605$$

La función de transferencia del controlador PID queda conformada como:

$$G_{PID}(s) = K_p + \frac{k_i}{s} + K_d s \quad (4.59)$$

$$G_{PID}(s) = \frac{51946s^2 + 77.92s + 0.1605}{s} \quad (4.60)$$

En la figura 4.41, se observa la respuesta del sistema de segundo orden en contraste con el sistema compensado. Este ultimo, no presenta sobreimpulso, con un tiempo de subida de 11 s y asentamiento de 19.5 s .

Herramienta PID Tuner

Al igual que en el caso anterior, Matlab nos permite definir las especificaciones que respondan a nuestro diseño de sistema de control. Es por ello, que se diseño un controlador con la siguiente función de transferencia y constantes visualizadas en la figura 4.42.

$$G_{PID}(s) = \frac{16081.5s^2 + 3529s + 193.67}{s} \quad (4.61)$$

Se diseño este controlador con las siguientes especificaciones:

- Tiempo de establecimiento < 10 min.
- Sobre impulso < 10 %.
- Tiempo de subida < 120s.

Los parámetros seleccionados nos brindó la performance del sistema con:

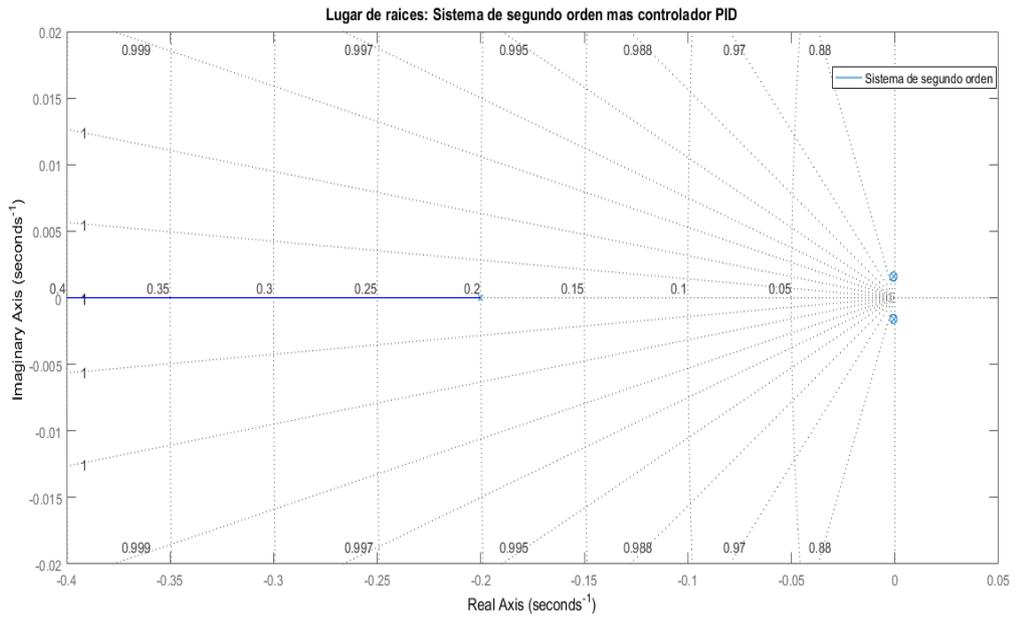


Figura 4.40: Lugar de raíces del sistema de segundo orden más controlador PID. [32]

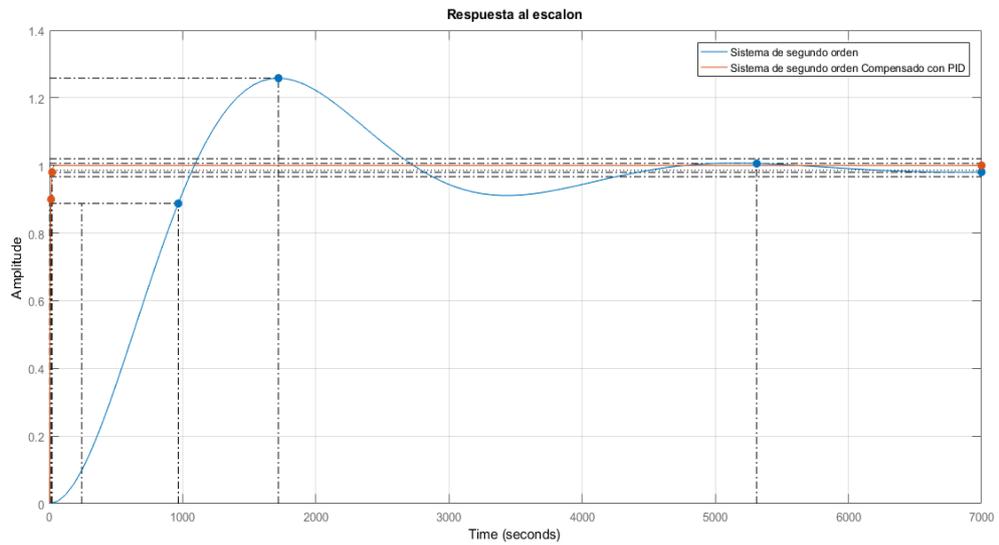


Figura 4.41: Respuesta al sistema de segundo orden. [32]

- Tiempo de establecimiento = 365 s.
- Sobre impulso = 4.22%.
- Tiempo de subida = 76.1 s.

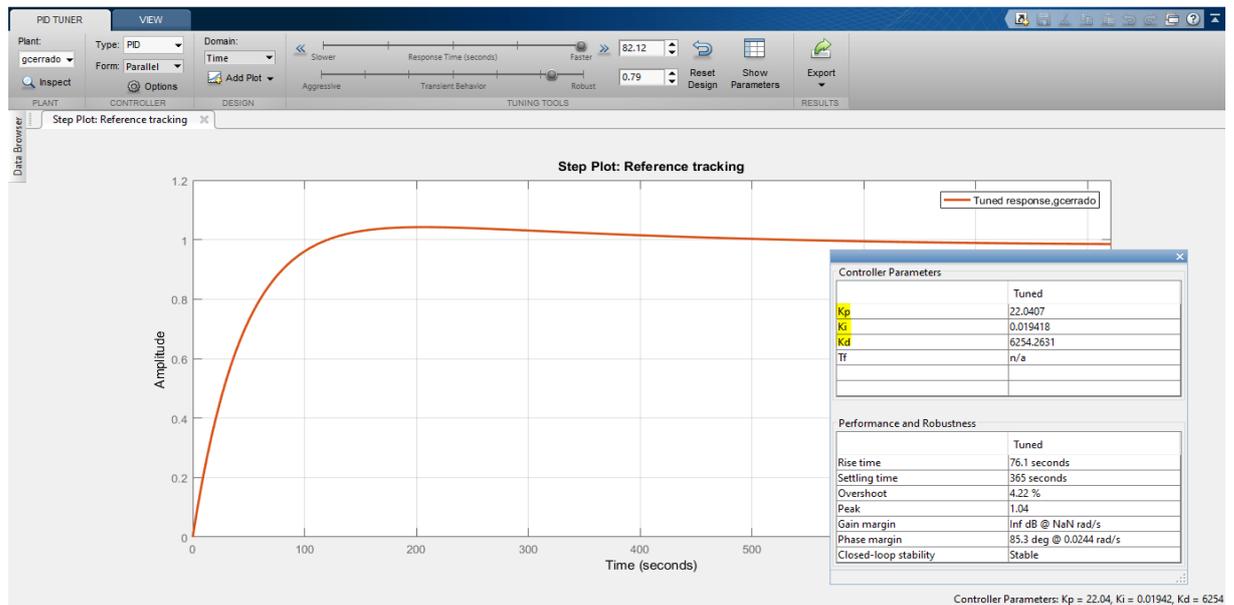


Figura 4.42: Diseño control PID y respuesta del sistema. [32]

Cumpliendo así, con los requerimientos solicitados.

4.7.4. Digitalización del Sistema

En este apartado se realizan la discretización de funciones de transferencias, tanto de las plantas anteriores como las funciones de los controladores que fueron implementados en el microcontrolador. Existen diferentes métodos para realizar discretización, a continuación se mencionan algunos:

1. Simulaciones invariantes y/o con Retenedor.

- Respuesta invariante al impulso.
- Respuesta a la entrada escalón o equivalencia ZOH. (Retenedor de orden cero).

2. Por aproximaciones o Transformaciones $s = F(z)$.

- Integración Rectangular (Euler).
- Integración Rectangular Retrasada (Euler hacia atrás).
- Integración Trapezoidal, conocida como Bilineal o Regla de Tustin.

Para representar las funciones de transferencia del sistema tanto de primer como de segundo orden, se utilizó el método de respuesta al escalón o más conocido como

retenedor de orden cero. La función de transferencia para tiempo discreto va a estar dada por:

$$G(z) = z[G_{zoh}(s)G(s)] = (1 - z^{-1})z\left[\frac{G(s)}{s}\right] \quad (4.62)$$

Sistema de primer orden

Tomando la función de transferencia dada por la ecu.(4.49), se obtiene su equivalente en z para un periodo de muestreo de $T= 1$ s como:

$$G_1(z) = \left(\frac{z-1}{z}\right)z\left[\frac{-0.0005706}{s(s+0.00052)}\right] \quad (4.63)$$

Multiplicando y dividiendo por 0.0652 tenemos que:

$$G_1(z) = \frac{-0.0005706}{0.00052}\left(\frac{z-1}{z}\right)z\left[\frac{0.00052}{s(s+0.00052)}\right] \quad (4.64)$$

$$G_1(z) = -1.0973\left(\frac{z-1}{z}\right)z\left[\frac{0.00052}{s(s+0.00052)}\right] \quad (4.65)$$

Haciendo uso de la Tabla de Transformada Z obtenemos:

$$G_1(z) = -1.0973\left(\frac{z-1}{z}\right)z\left[\frac{1 - e^{-0.00052T}}{(z-1)(z - e^{-0.00052T})}\right] \quad (4.66)$$

Simplificando y realizando los cálculos para $T=0.01$ s finalmente se obtiene:

$$G_1(z) = \frac{-0.0005704}{z - 0.9995} \quad (4.67)$$

De manera de corroborar los cálculos, se utilizó el comando c2d de Matlab. Esto se visualiza en la figura 4.43 .

Se comenta que el valor del numerador difiere un poco debido a que, en el momento de realizar los cálculos a mano no se tomaron en cuenta todos los decimales posibles como lo realiza Matlab.

```

gabierto =
    -0.0005706
    -----
    s + 0.00052

Continuous-time transfer function.

>> Planta_Digital=c2d(gabierto,1,'zoh')

Planta_Digital =
    -0.0005705
    -----
    z - 0.9995

Sample time: 1 seconds
Discrete-time transfer function.

```

Figura 4.43: Comando para discretizar con retenedor de orden cero. [32]

Sistema de segundo orden

En este caso, considerando la función de transferencia dada por la ecu.(4.51), se obtiene su equivalente en z para un periodo de muestreo de $T= 1$ s . Se aclara que el cálculo fue realizado a mano, sin embargo no se transcribió para no hacer extenso este documento. Finalmente se comparó el resultado obtenido a través del comando de Matlab, como se hizo anteriormente.

$$G_2(z) = \left(\frac{z-1}{z}\right)z \left[\frac{0.00000385}{s(s+0.0009)(s+0.0006)} \right] \quad (4.68)$$

$$G_2(z) = \frac{1.927 \times 10^{-6} z + 1.926 \times 10^{-6}}{(z-0.9994)(z-0.9991)} \quad (4.69)$$

4.7.5. Discretización del controlador PID

Para realizar el sistema de control en un microcontrolador, es necesario que la función de transferencia del compensador (controlador PID), se encuentre discretizada para poder ser implementado y así responder con las especificaciones requeridas del sistema.

Las propiedades del controlador digital dependen principalmente de:

- El periodo de muestreo T.

```

          3.855e-06
          -----
          s^2 + 0.0015 s + 5.4e-07

Continuous-time transfer function.

>> Planta_digital=c2d(sistema2orden,1,'zoh')

Planta_digital =

          1.927e-06 z + 1.926e-06
          -----
          z^2 - 1.999 z + 0.9985

Sample time: 1 seconds
Discrete-time transfer function.

>> zpk(Planta_digital)

ans =

          1.9265e-06 (z+1)
          -----
          (z-0.9994) (z-0.9991)

```

Figura 4.44: Comando para discretizar con retenedor de orden cero. [32]

- Método de digitalización utilizado.

Para encontrar la función del controlador tanto PI como PID se utilizó un periodo de muestreo de $T=1s$ y método de digitalización Euler hacia adelante.

Controlador PI

Tomando la ec.(4.54) obtenida a través del método cancelación de polos, aplicando el método de discretización Euler hacia adelante, se tiene que:

$$G_{PI}(z) = K_p + \frac{K_i T}{(1 - z^{-1})} \quad (4.70)$$

$$G_{PI}(z) = \frac{K_p - K_p z^{-1} + K_i T}{(1 - z^{-1})} \quad (4.71)$$

$$G_{PI}(z) = \frac{-350.68z + 350.5}{z - 1} \quad (4.72)$$

Por consiguiente aplicando el método de digitalización a la ec.(4.56) brindada por PID Tuner, tenemos:

$$G_{PI}(z) = \frac{-1529z + 1528}{z - 1} \quad (4.73)$$

Controlador PID

Considerando la ec.(4.58) y ec.(4.59), obtenidas por cancelación de polos y PID Tuner respectivamente, se encontraron las siguientes funciones discretizadas:

$$G_{PID}(s) = K_p + \frac{K_i}{s} + K_d s \quad (4.74)$$

$$G_{PID}(z) = K_p + \frac{K_i T}{(1 - z^{-1})} + \frac{K_d(1 - z^{-1})}{T} \quad (4.75)$$

$$G_{PID}(z) = \frac{\frac{K_d}{T} z^{-2} - (\frac{2K_d}{T} + K_p) z^{-1} + (K_p + K_i T + \frac{K_d}{T})}{(1 - z^{-1})} \quad (4.76)$$

$$G_{PID}(z) = \frac{1.04x10^5 z^2 - 2.078x10^5 z + 1.03810^5}{z^2 - 1} \quad (4.77)$$

$$G_{PID}(z) = \frac{3.57x10^4 z^2 - 6.413x10^4 z + 2.87x10^4}{z^2 - 1} \quad (4.78)$$

4.7.6. Ecuaciones en Diferencias

Lo que se programa en cualquier microcontrolador son aquellas ecuaciones en diferencias del controlador a través de arreglos (vectores). Consideremos el diagrama de bloques de la figura 4.45 :

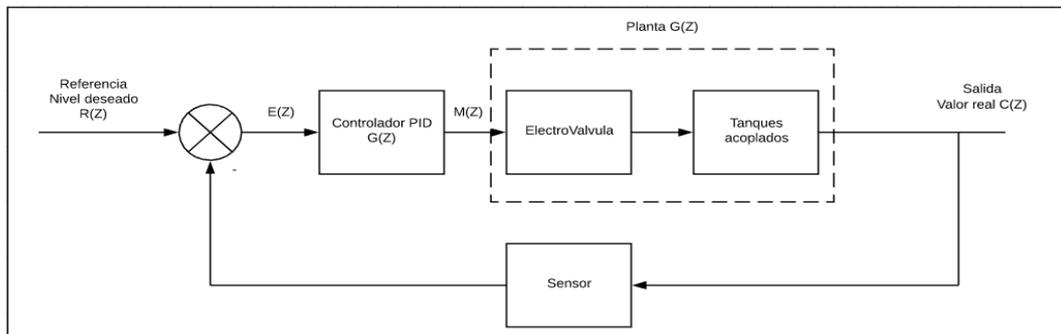


Figura 4.45: Diagrama de bloques Discreto. [32]

Se observa que la señal $M(z)$ es la encargada de generar la acción de manipulación del controlador que actúa directamente en el actuador. Podemos representar al controlador como:

$$G_c(z) = \frac{M(z)}{E(z)} \quad (4.79)$$

Donde:

- $E(z)$ = Señal Error (proveniente de la diferencia del Set Point con el valor real de la salida).
- $M(z)$ = Señal de salida del controlador.

Ecuación en diferencia controlador PI

Se encontró la ecuación en diferencias de la ec.(4.72), de la siguiente manera:

$$G_{PI}(z) = K_p + \frac{K_i T}{(1 - z^{-1})} \quad (4.80)$$

$$G_{PI}(z) = \frac{-K_p z^{-1} + (K_p + K_i T)}{(1 - z^{-1})} \quad (4.81)$$

Consideremos $K_a = K_i T + K_p$

$$G_{PI}(z) = \frac{-K_p z^{-1} + K_a}{(1 - z^{-1})} \quad (4.82)$$

$$\frac{M(z)}{E(z)} \cdot (1 - z^{-1}) = -K_p z^{-1} + K_a$$

$$M(z) - M(z)z^{-1} = [K_p z^{-1} + K_a]E(z)$$

$$M(z) = M(z)z^{-1} - K_p z^{-1}E(z) + (K_p + K_i T)E(z)$$

$$M(k) = M(k-1) - K_p \cdot e(k-1) + (K_p + K_i T) \cdot e(k) \quad (4.83)$$

$$M(k) = M(k-1) + K_p \cdot [e(k) - e(k-1)] + K_i T \cdot e(k) \quad (4.84)$$

Reemplazando los valores de K_p , K_i y T_m , tenemos:

$$M(k) = M(k-1) - 350 \cdot [e(k) - e(k-1)] - 0.18 \cdot e(k) \quad (4.85)$$

Ecuación en diferencia controlador PID

Para encontrar la ecuación en diferencias de la ec.(4.76), se procedió de la misma manera anteriormente realizada. El controlador PID a implementar en el sistema de segundo orden (2 tanques) esta dado por:

$$\frac{M(z)}{E(z)} = \frac{\frac{K_d}{T}z^{-2} - (\frac{2K_d}{T} + K_p)z^{-1} + (K_p + K_iT + \frac{K_d}{T})}{(1 - z^{-1})} \quad (4.86)$$

Siguiendo pasos algebraicos llegamos a:

$$M(k) = M(k - 1) + Q_2 \cdot e(k - 2) - Q_1 \cdot e(k - 1) + Q_0 \cdot e(k) \quad (4.87)$$

Donde:

- $Q_2 = \frac{K_d}{T} =$
- $Q_1 = [\frac{2K_d}{T} + K_p]$
- $Q_0 = [K_p + K_iT + \frac{K_d}{T}]$

La ecuación (4.87), es la implementada en el controlador para el sistema de 2° orden. Es importante detallar el comportamiento del sistema debido a la acción derivativa (tiempo de filtro derivativo).

Sabemos que la acción de control integral, elimina el error estacionario (offset), de la acción proporcional. Ahora la acción derivativa, brinda una característica de anticipación del comportamiento futuro del error. Esta predicción es hecha simplemente calculando la derivada del error en el instante de tiempo actual, que en otras palabras calcula la tangente a ese punto y traza **una proyección** hacia el futuro del posible comportamiento que tendrá el error.

Podemos concluir que el tiempo derivativo es sumamente importante. Si el tiempo es muy grande, nos estaríamos anticipando mucho en el futuro al comportamiento del error y esto puede generar una predicción bastante errónea. Una acción derivativa pequeña, permite anticiparse poco tiempo en el futuro permitiendo una mejor precisión. Y una acción derivativa demasiado pequeña, no tendrá un efecto significativo en el sistema. Esto podemos observar en la siguiente figura.

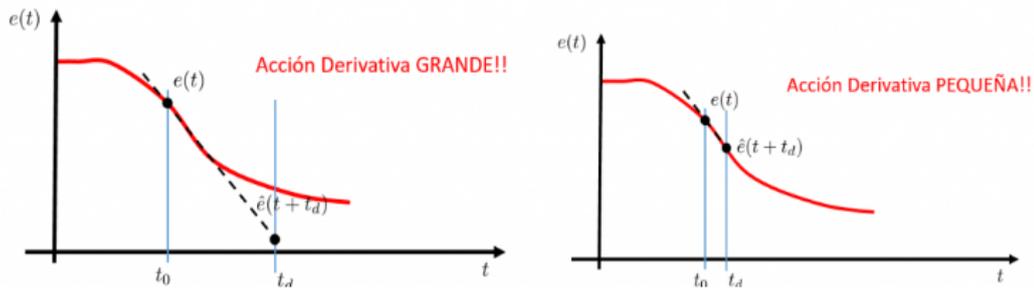


Figura 4.46: Comparación de T_d .

4.8. Consideraciones de Diseño - Software

Esta sección describe la parte concerniente a la programación en diferentes plataformas y entornos de desarrollos, como así también al Hardware utilizado. Es útil resaltar que la programación en diferentes lenguajes, ha incrementado el desarrollo de aplicaciones tanto a nivel prototipo como industrial, por lo que es importante tener en cuenta estos lenguajes y adquirir los conocimientos para su desarrollo.

4.8.1. Programación del Microcontrolador

La parte de programación del MCU, esta conformada por un grupo de librerías que resumen el código y las funciones necesarias para programar la placa Arduino. Estas librerías contienen los parámetros para poner en funcionamiento toda la unidad. Las rutinas del MCU fueron desarrolladas en lenguaje C++, haciendo uso del IDE de Arduino en su versión 1.8.3 .

Para obtener una mejor comprensión de la programación del Microcontrolador y así mismo de toda la unidad, se diseñó un diagrama de estados como lo muestra la siguiente figura.

La transición de estado a estado se detalla a continuación:

Estados **Inicialización - Bienvenida**: A través de un Timer de 7 segundos programado en la pantalla Touch.

Estados **Bienvenida - Contraseña**: Mediante el toque sobre cualquier lugar de la pantalla. Para regresar se aprieta la flecha en la parte superior de la pantalla Contraseña.

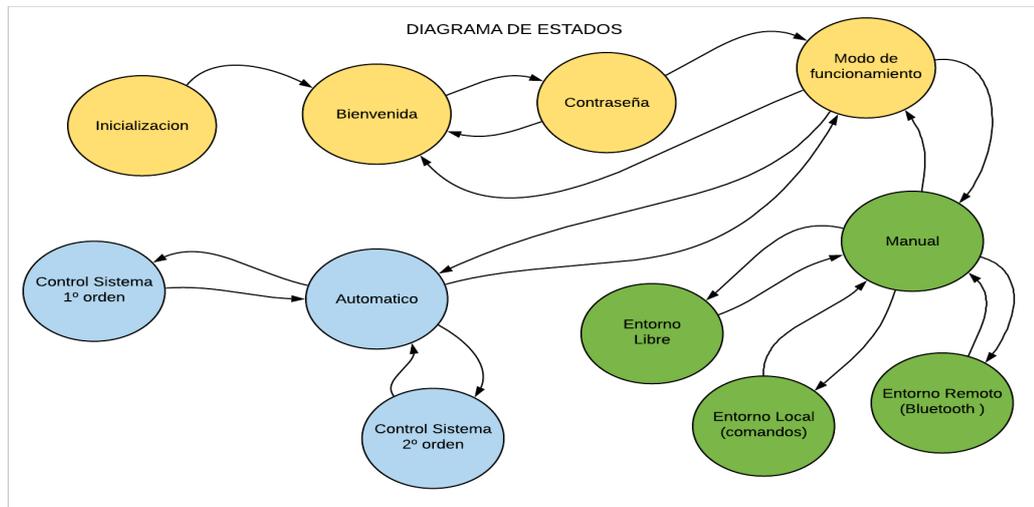


Figura 4.47: Diagrama de estados programación. [32]

Estados **Contraseña - Modo de Funcionamiento**: Ingresando el usuario **TEC-NOUNCA** (con mayúscula) y la contraseña **electro58**.

Estados **Modo de Funcionamiento - Automático - Manual**: A través del toque del icono correspondiente a los diferentes modo en la pantalla.

La programación de los estados hasta aquí descritos, corresponde a la programación directamente en la Pantalla Nextion Touch.

Estados **Automático - Control Sistema 1º orden**: A través del toque del icono correspondiente ingresamos al control de primer orden con el siguiente evento:

```

1 void boton_inicio_control_1PushCallback(void *ptr) //Funcion para ingresar al modo de
  control de un tanque.
2 {
3   variable_de_salida_libre = 1;
4   variable_de_salida_bluetooth = 1;
5   variable_de_salida_comando = 1;
6   variable_de_parada_control = 0;
7
8   while (variable_de_parada_control !=1)
9   {
10    digitalWrite(33,LOW);
11    digitalWrite(35,LOW);
12    digitalWrite(37,LOW);
13    digitalWrite(39,LOW);
14    digitalWrite(41,LOW);
15    digitalWrite(43,LOW);
16    digitalWrite(45,LOW);
17
18    digitalWrite(31,HIGH);
19    delay(100);
20    digitalWrite(31,LOW);

```

```

21   delay (100);
22   digitalWrite(31,HIGH);
23   delay (100);
24   digitalWrite(31,LOW);
25   nexLoop(nex_listen_list) ;
26   delay (1000);
27   Sensor_Mpx_control();
28   delay (1);
29   control_pi();
30 }
31 }

```

Listing 4.1: Evento para ingresar al control de primer orden.

La porción de código que efectúa el sistema de control es la siguiente función:

```

1 void control_pi()
2 {
3   actual = millis(); //Milisegundos desde que se inicio el programa
4
5   unsigned long dt = actual-pasado; //Tiempo de muestreo
6   float t; //Tiempo que demora en llegar el eco
7   float d; //distancia en cm
8   float Altura; // Variable que almacena la altura del tanque 1
9
10
11  digitalWrite(Triquer, LOW);
12  delayMicroseconds(1);
13  digitalWrite(Triquer, HIGH);
14  delayMicroseconds(10); //Enviamos un pulso de 10us
15  digitalWrite(Triquer, LOW);
16
17  t = pulseIn(Echo, HIGH); //Obtenemos el ancho del pulso
18
19  d = t * 10 / 292/ 2; //Escalamos el tiempo a una distancia en cm
20  Altura = 30-d;
21
22
23  if ( Altura < valor_maximo_nivel )
24  {
25    digitalWrite(Relay_Bomba, HIGH);
26  }
27  else
28  {
29    digitalWrite(Relay_Bomba, LOW);
30  }
31
32  if (( error >= -2) && (error <= 2))
33  {
34    digitalWrite(Relay_Bomba, LOW);
35  }
36
37
38  const float SensorOffset = 0; //40.5
39  float sensorValue;
40  float tension;
41  float altura;
42  float barra;
43
44  sensorValue = (analogRead(A0)-SensorOffset) ;
45  tension = sensorValue*(5.0/1024.0);
46  altura = (tension-0.25 )/0.005;
47  delay(1);
48
49  memset(buffer, 0, sizeof(buffer));

```

```

50  nsetpoint.getValue(&Sp); //Recibe setpoint de la pantalla
51  delay(10);
52  nP.getValue(&Kp); //Recibe valor Kp
53  delay(10);
54  nI.getValue(&Ki); //Recibe valor Ki
55
56  if(dt >= T) //Realiza el control dentro del periodo de muestreo.
57  {
58      pasado = actual;
59      Pv = map(altura,0,300,0,30);
60
61      error = (Sp-Pv); //Calculo del error
62      delay(5);
63
64      Cv = Cv + (Kp*(error-error_1))+(Ki*0.01*0.5)*(error+error_1);
65      delay(5);
66
67      error_1 = error; //Error anterior
68      delay(5);
69      Cv=constrain(Cv,0,30); //Restriccion de accion de control
70      PWM = map(Cv,0,30,-90,0); //Fija la posicion de la electrovalvula a traves de
71      los pasos.
72      stepper.runToNewPosition(PWM);
73      delay(5);
74
75      onda_control_1.addValue(0,Sp);
76      onda_control_1.addValue(1,Pv);
77  }
78 }

```

Listing 4.2: Función de control del sistema de primer orden.

Para detener el sistema de control y así poder regresar al menú anterior, se creó una función de parada del control. Es ejecutada en el icono correspondiente mostrado en la pantalla touch.

```

1 void boton_parada_control_1PushCallback(void *ptr) //Funcion que realiza cuando se
2   apreta el boton STOP
3 {
4   variable_de_parada_control = 1;
5   variable_de_salida_bluetooth = 1;
6   variable_de_salida_comando = 1;
7   variable_de_salida = 1;
8   digitalWrite(Relay_Bomba, LOW);
9   delay(1);
10  nexLoop(nex_listen_list);
11  delay(1);
12  nexLoop(nex_listen_list);

```

Listing 4.3: Función de parada del control de primer orden.

Estados **Automático - Control Sistema 2º orden**: A través del toque del icono correspondiente al sistema de segundo orden.

Estados **Manual - Libre**: A través del toque del icono correspondiente al modo libre.

La programación para acceder al entorno manual libre se realizó a través de la siguiente función:

```
1 void boton_entrada_librePushCallback(void *ptr) // Funcion que realiza cuando se
    apreta se apreta LA ENTRADA A MODO LIBRE
2 {
3 variable_de_salida_comando = 1;
4 variable_de_salida_libre = 0;
5 variable_de_salida_bluetooth = 1;
6
7 while (variable_de_salida_libre !=1)
8 {
9     nexLoop(nex_listen_list);
10    delay (10);
11 }
12 }
```

Listing 4.4: Evento para ingresar al modo Libre.

Dentro de este entorno se permite hacer uso de diferentes actuadores, bomba, electroválvula y dos tipos de sensores, ultrasonidos y de presión. Para activar o desactivar cada uno de ellos, se realizaron los siguientes eventos:

```
1 void cMPXPushCallback(void *ptr) //Evento para sensor MPX
2 {
3 uint32_t estadoMPX;
4 memset(buffer, 0, sizeof(buffer));
5 cMPX.getValue(&estadoMPX);
6
7 if(estadoMPX)
8 {
9     digitalWrite(Relay_sensor_mpx_manual, HIGH );
10    delay (1);
11 }
12 else {
13     digitalWrite(Relay_sensor_mpx_manual, LOW);
14     delay (1);
15 }
16 }
17
18 void cSONIDOPushCallback(void *ptr) //Evento para sensor Ultrasonido
19 {
20 uint32_t estadoSONIDO;
21 memset(buffer, 0, sizeof(buffer));
22 cSONIDO.getValue(&estadoSONIDO);
23 if(estadoSONIDO)
24 {
25     digitalWrite(Relay_sensor_ultrasonido, HIGH );
26     digitalWrite(Relay_sensor_ultrasonido_hecho, HIGH );
27     delay (1);
28 }
29 else {
30     digitalWrite(Relay_sensor_ultrasonido, LOW );
31     digitalWrite(Relay_sensor_ultrasonido_hecho, LOW );
32     delay (1);
33 }
```

```

33     }
34 }
35
36 void cBOMBAPushCallback(void *ptr) //Evento para bomba
37 {
38     uint32_t estadoBOMBA;
39     memset(buffer, 0, sizeof(buffer));
40     cBOMBA.getValue(&estadoBOMBA);
41     if(estadoBOMBA)
42     {
43         digitalWrite(Relay_Bomba_manual, HIGH );
44         delay (1);
45     }
46     else {
47         digitalWrite(Relay_Bomba_manual, LOW );
48         delay (1);
49     }
50 }
51
52 void cVALVULAPushCallback(void *ptr) //Evento para electrovalvula
53 {
54     uint32_t estadoVALVULA;
55     memset(buffer, 0, sizeof(buffer));
56     cVALVULA.getValue(&estadoVALVULA);
57     if(estadoVALVULA)
58     {
59         variable_de_salida_libre = 0;
60         while (variable_de_salida_libre !=1)
61         {
62             nexLoop(nex_listen_list) ;
63             delay (100);
64             digitalWrite(Valvula, HIGH );
65             delay (100);
66             digitalWrite(Valvula, LOW );
67             delay (100);
68             posicion_valvula();
69         }
70     }
71     else {
72         variable_de_salida_libre = 1;
73         digitalWrite(Valvula, LOW );
74         delay (1);
75     }
76 }

```

Listing 4.5: Programación de eventos para modo Libre.

Durante el evento para activar la electroválvula en el modo libre, se produce una llamada a una función diseñada para leer el valor de tensión aplicado al tablero y así tomar su correspondiente valor ADC y manipular los pasos con el cual, el motor debe girar y posicionar la electroválvula.

```

1 void posicion_valvula()
2 {
3     int giro = 0 ;
4     Posicion_valvula = (analogRead(A2)) ;
5     tension_posicion_valvula = Posicion_valvula*(5/1023);
6
7     giro = map(Posicion_valvula,0,1023,0,-98);
8     delay (10);
9
10    stepper.runToNewPosition(giro);

```

```

11 //Posicion de abierto (-98) . Valor del potenciometro en el conversor analogico
12 de 5V . 0- 1023
13 delay(100);
14 }

```

Listing 4.6: Función para electroválvula en modo libre.

Estados **Manual - Comandos**: A través del toque del icono correspondiente al Comando se ingresa a partir de la siguiente función:

```

1 void boton_entrada_comandoPopCallback(void *ptr)// Funcion que realiza cuando se
  apreta LA ENTRADA A MODO COMANDO
2 {
3   variable_de_salida_libre = 1;
4   variable_de_salida_bluetooth = 1;
5   variable_de_salida_comando = 0;
6
7   while (variable_de_salida_comando !=1)
8     {
9       nexLoop(nex_listen_list);
10      delay (2000);
11      Sensor_Mpx ();
12      delay (1);
13      Envio_Datos_Sensor_Ultrasonido();
14      delay (1);
15    }
16 }

```

Listing 4.7: Función para ingresar al modo comandos.

Dentro de este comando se envían los datos de los dos niveles para ser visualizados a través de la pantalla y permite activar y/o desactivar la bomba, como así también abrir y cerrar la electroválvula, a partir de las siguientes funciones desarrolladas.

```

1 void Envio_Datos_Sensor_Ultrasonido()
2 {
3   float t; //tiempo que demora en llegar el eco
4   float d; //distancia en centimetros
5   float e;
6   digitalWrite(Triquer, LOW);
7   delayMicroseconds(4);
8   digitalWrite(Triquer, HIGH);
9   delayMicroseconds(10); //Enviamos un pulso de 10us
10  digitalWrite(Triquer, LOW);
11
12  t = pulseIn(Echo, HIGH); //obtenemos el ancho del pulso
13  d =292-(t/5.9); //escalamos el tiempo a una distancia en mm
14
15
16  e =((d)*100)/300 ; //escalamos el tiempo porcentualmente para la
  barra de progreso en la pantalla Nextion.
17  delay(1);
18
19  nexLoop(nex_listen_list);
20  n0Bluetooth.setValue(d);
21
22  delay(1);
23  nexLoop(nex_listen_list);
24  j0Bluetooth.setValue(e);
25

```

```

26  delay(1);
27  nexLoop(nex_listen_list);
28  n0comando.setValue(d);
29
30  delay(1);
31  nexLoop(nex_listen_list);
32  j0comando.setValue(e);
33  delay(1);
34  }
35
36  void bt0PushCallback(void *ptr) //Funcion que realiza cuando se apreta BOMBA-MODO-
    COMANDOS de la pantalla
37  {
38      uint32_t dual_state1;
39      memset(buffer, 0, sizeof(buffer));
40      bt0.getValue(&dual_state1);
41  if(dual_state1)
42  {
43      digitalWrite (Relay_Bomba,HIGH);
44      delay (1);
45  }
46
47  else {
48      digitalWrite (Relay_Bomba,LOW);
49      delay (1);
50      }
51  }
52
53  void bt1PushCallback(void *ptr)//Funcion que realiza cuando se apreta VALVULA MODO-
    COMANDOS de la pantalla
54  {
55      uint32_t dual_state;
56      memset(buffer, 0, sizeof(buffer));
57      bt1.getValue(&dual_state);
58  if(dual_state)
59  {
60      Valvula_Abierta();
61      delay (1);
62  }
63  else {
64      Valvula_Cerrada();
65      delay (1);
66      }
67  }
68
69  void Valvula_Cerrada() //Funcion para cerrar valvula
70  {
71      stepper.runToNewPosition(0); // Posicion de cerrado (0) . Valor del potenciómetro
    en el conversor analogico de 0V- 0
72      delay(100);
73  }
74
75  void Valvula_Abierta() //Funcion para abrir valvula
76  {
77      stepper.runToNewPosition(-98); // Posicion de abierto (-98). Valor del
    potenciómetro en el conversor analogico de 5V- 1023
78      delay(100);
79  }

```

Listing 4.8: Funciones utilizadas en el modo comando.

Estados **Manual - Bluetooth**: A través del toque del icono correspondiente al modo Bluetooth, en donde se ejecuta la siguiente función.

```

1 void boton_entrada_bluetoothPushCallback(void *ptr) //Funcion que realiza cuando se
  apreta LA ENTRADA A MODO BLUETOOTH
2 {
3     variable_de_salida_comando = 1;
4     variable_de_salida_libre = 1;
5     variable_de_salida_bluetooth = 0;
6     int salida_bluetooth = 0 ;
7
8     while (variable_de_salida_bluetooth !=1)
9     {
10
11         nexLoop(nex_listen_list);
12         delay (2000);
13         Sensor_Mpx();
14         delay (10);
15         Envio_Datos_Sensor_Ultrasonido();
16         delay (100);
17
18     if (Serial1.available()>0)
19     {
20         char dato= Serial1.read();
21         texto_bluetooth.setText("Conectado");
22         if (dato=='a')
23         {
24             rBOMBA.Set_background_color_bco(1344);
25             digitalWrite (Relay_Bomba,HIGH);
26             delay (1);
27         }
28
29         else if (dato=='b')
30         {
31             rBOMBA.Set_background_color_bco(53248);
32             digitalWrite (Relay_Bomba,LOW);
33             delay (1);
34         }
35
36         else if (dato== 'c')
37         {
38             Valvula_Abierta();
39             delay (1);
40             rVALVULA.Set_background_color_bco(1344);
41         }
42
43         else if (dato== 'd')
44         {
45             Valvula_Cerrada();
46             rVALVULA.Set_background_color_bco(53248);
47             delay (1);
48         }
49
50         else if (dato== 's')
51         {
52             digitalWrite (Relay_Bomba,LOW);
53             Valvula_Cerrada();
54             rVALVULA.Set_background_color_bco(53248);
55             delay (1);
56             salida_bluetooth = 1;
57             variable_de_salida = 1;
58             pageManual.show(); //Funcion que me envia a la pagina anterior en la
  pantalla
59     }
60 }
61 }
62 }

```

Listing 4.9: Funciones utilizadas en el modo bluetooth.

Como se observa en cada evento, presentan un ciclo while donde realizan las acciones, siempre y cuando se apreté en el botón (icono) indicado. Ahora para salir y volver a la pantalla de Modo Manual, se debe apretar en la flecha ubicada en la parte superior de cada entorno. Lo que sucede es que llama a otra función desarrollada para que el sistema vuelva a ubicarse en modo espera desde el Modo Manual. Estas se describen a continuación:

```

1 void salida_comandosPopCallback(void *ptr) //Funcion que realiza cuando se apreta LA
  FLECHA DE SALIDA DE COMANDO
2 {
3     variable_de_salida = 1 ;
4     variable_de_salida_libre = 1;
5     variable_de_salida_bluetooth = 1;
6     variable_de_salida_comando = 1;
7
8     nexLoop(nex_listen_list);
9     delay (1);
10    nexLoop(nex_listen_list);
11    digitalWrite(Relay_Bomba, LOW);
12    Valvula_Cerrada();
13 }
14
15 void salida_librePushCallback(void *ptr) //Funcion que realiza cuando se apreta LA
  FLECHA DE SALIDA DE LIBRE
16 {
17     nexLoop(nex_listen_list) ;
18     variable_de_salida_libre = 1;
19     variable_de_salida_bluetooth = 1;
20     variable_de_salida_comando = 1;
21     variable_de_salida = 1 ;
22     delay (1);
23     nexLoop(nex_listen_list);
24     digitalWrite(Relay_Bomba, LOW);
25     Valvula_Cerrada();
26 }
27
28 void salida_bluetoothPushCallback(void *ptr)//Funcion que realiza cuando se apreta LA
  FLECHA DE SALIDA DE BLUETOOTH
29 {
30     variable_de_salida_libre = 1;
31     variable_de_salida_bluetooth = 1;
32     variable_de_salida_comando = 1;
33
34     nexLoop(nex_listen_list);
35     variable_de_salida = 1;
36     delay (1);
37     nexLoop(nex_listen_list);
38     digitalWrite(Relay_Bomba, LOW);
39     Valvula_Cerrada();
40 }

```

Listing 4.10: Funciones utilizadas para regresar al modo de espera Manual.

4.8.2. Pantalla Nextion

La programación de la pantalla interactiva touch Nextion, se realizó a través del entorno de programación brindada por el fabricante. Se desarrolló la programación

simulando una GUI (Interfaz Gráfica de Usuario), con la finalidad de automatizar las tareas y cálculos, ejerciendo un mayor control y supervisión de toda la unidad.



Figura 4.48: Programación de la pantalla Táctil. [32]

El editor Nextion tiene componentes masivos tales como botones, texto, barra de progreso, slider, panel de instrumentos, etc. para enriquecer el diseño de la interfaz. Por su parte la sintaxis de programación de cada evento, es un pseudo lenguaje C++ con instrucciones en código ASCII, es decir, el editor tiene asociado un set de instrucciones que hacen más accesible la programación.

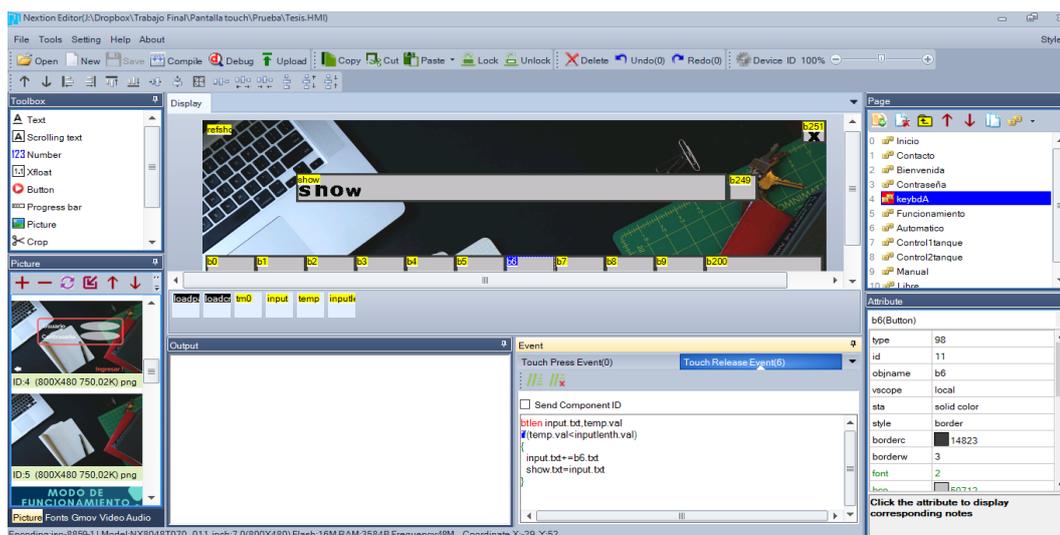


Figura 4.49: Instrucciones de programación. [32]

La pantalla Nextion se conecta al microcontrolador a través del Protocolo Serial TTL (5V, TX, RX, GND). Proporcionando las notificaciones de los eventos para que el microcontrolador pueda actuar y actualizar los eventos en la pantalla.

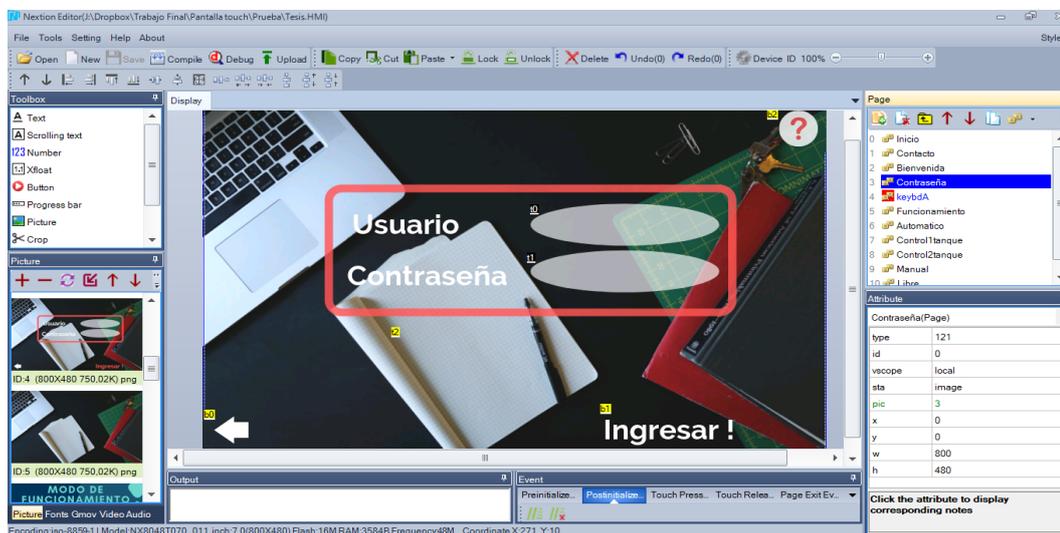


Figura 4.50: Programación pantalla. [32]

4.8.3. App Inventor- Aplicación para Android

Otra de las ventajas que presenta este trabajo, es su uso de manera remota. Para ello, se desarrolló una aplicación para dispositivos Android de manera que pueda ser utilizada para el desarrollo de diferentes trabajos prácticos sin necesidad de estar frente a la unidad.

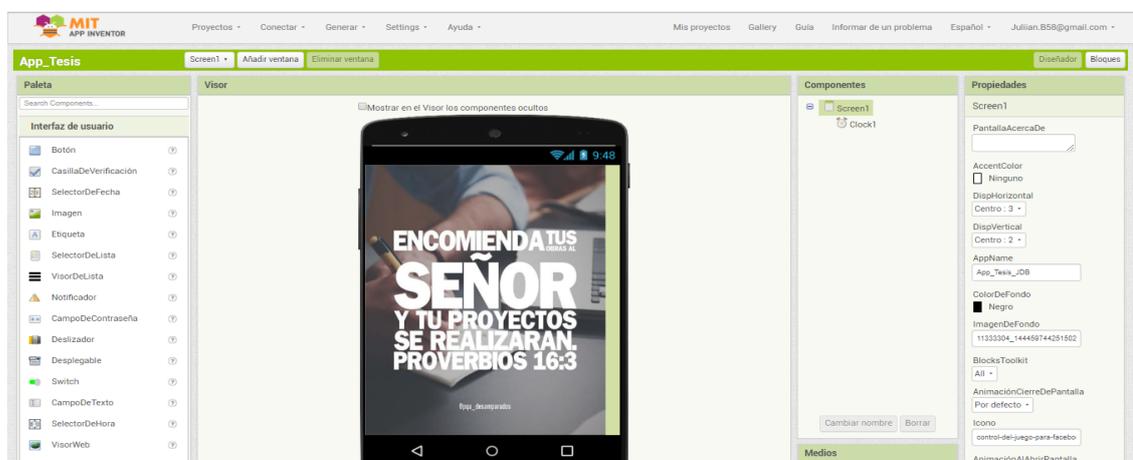


Figura 4.51: Entorno de desarrollo MIT App Inventor. [32]

El entorno de programación de la aplicación, es a través de diagramas de bloques que responde a un set de instrucciones pseudo lenguaje C.



Figura 4.52: Programación en bloques de la aplicación. [32]

Con esta aplicación resulta posible realizar las siguientes tareas y opciones:

- **Visualizar** el nivel de líquido en ambos tanques en modo Automático .
- **Activar/Desactivar** la bomba en modo Manual.
- **Abrir/Cerrar** la electroválvula en modo Manual.

4.9. Funcionamiento de la Unidad

En esta sección se describe las variantes de uso que presenta la unidad, permitiendo a cada usuario realizar practicas en los diferentes entornos. De manera general, la unidad presenta dos modos de funcionamiento: Automático y Manual.

4.9.1. Modo Automático

En este modo de operación, la unidad realiza un sistema de control sobre los tanques de manera automática a partir de parámetros ingresados por el usuario. Permite seleccionar trabajar entre un sistema de primer orden y/o un sistema de segundo orden.

Dentro de este modo de operación se permite realizar las siguiente funciones:

- Elegir el sistema de control.
- Fijar el Set Point a través de la pantalla.
- Fijar las constantes de los parámetros del PID.
- Visualizar la respuesta del sistema.
- Visualizar el nivel del líquido en los tanques.
- Visualizar el nivel a través de la aplicación para Android.

4.9.1.1. Control 1º Orden

Este entorno permite al usuario interactuar con un sistema de control de nivel de primer orden, es decir, realizando el control sobre un solo tanque de la unidad (tanque n°1).

El sistema de control, utiliza las ecuaciones previamente encontradas para el control sobre un tanque, junto con las constantes del controlador PID. Además brinda al usuario la posibilidad de realizar cálculos externos del controlador para estudiar su comportamiento y entender el funcionamiento del sistema.

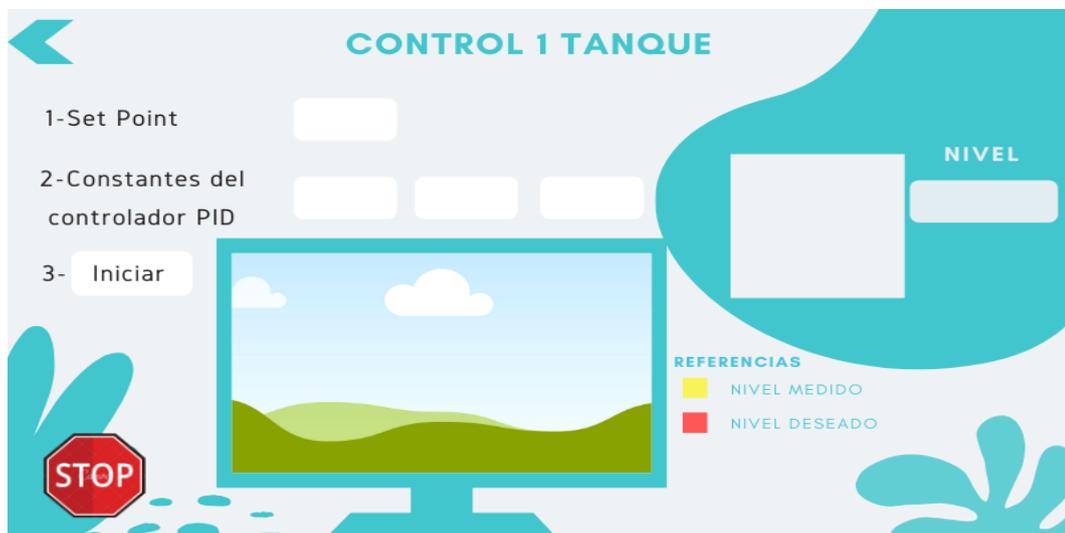


Figura 4.53: Display control de primer orden. [32]

Las constantes encontradas para el controlador PI en este trabajo:

$$G_{PI}(s) = K_p + \frac{k_i}{s} \quad (4.88)$$

$$G_{PI}(s) = \frac{-350s - 0.18}{s} \quad (4.89)$$

- $K_p = -350$
- $K_i = -0.18$
- $T_m = 1 \text{ s}$

Sistema de primer orden

$$G_{planta} = \frac{-0.0005706}{s + 0.00052} \quad (4.90)$$

El uso del entorno es intuitivo y consiste en una serie de pasos detallados en el display. Primeramente se selecciona el set point, nivel de liquido a controlar, seguido se constituyen las constantes del controlador, para este caso se usa el controlador PID, y finalmente se inicia el control a través del botón **iniciar**.

En esta pantalla, el usuario puede visualizar la respuesta del sistema (curva temporal) y el nivel del líquido en el tanque, con respecto a los parámetros ingresados. Además existe un led indicador (primer led de la tira) donde parpadea cada 2 segundos, indicando que el sistema de control se esta ejecutando.

Para detener el sistema, se debe apretar dos veces el botón **Stop**.

4.9.1.2. Control 2º Orden

Cuando se desea trabajar con un sistema más complejo, de segundo orden, este entorno dispone las herramientas para hacerlo. Cuenta con opciones para establecer el Set Point y los valores del controlador, para este caso PID. Se aclara que este tipo de control se realiza sobre el tanque N°2 dado que el análisis de este trabajo se basó en controlar el nivel de dicho tanque.

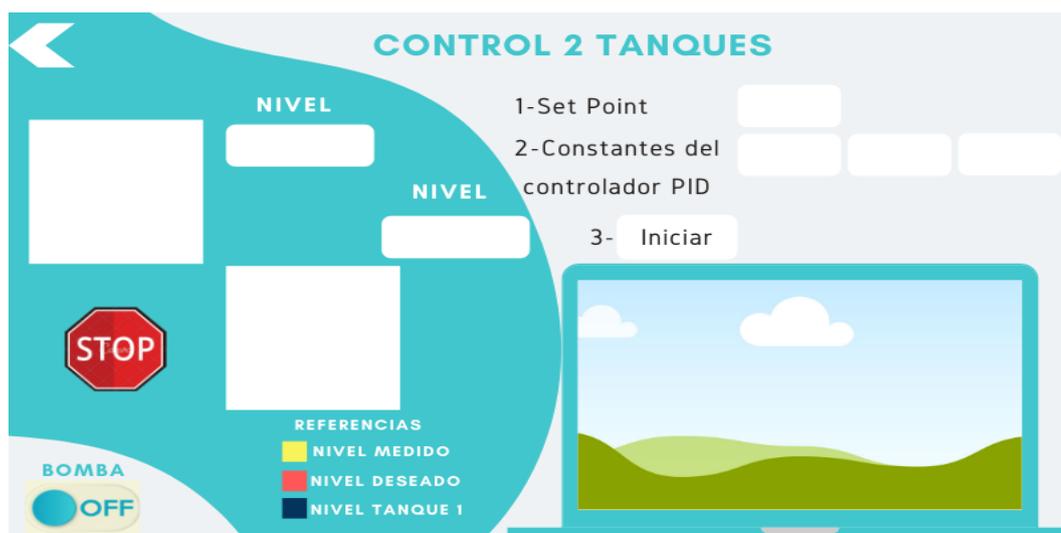


Figura 4.54: Display control de segundo orden. [32]

Las constantes del controlador PID encontradas son:

$$G_{PID}(s) = K_p + \frac{k_i}{s} + K_d s \quad (4.91)$$

$$G_{PID}(s) = \frac{51946s^2 + 77.92s + 0.1605}{s} \quad (4.92)$$

- $K_p = 77.92$
- $K_i = 0.16051$
- $K_d = 51946.1$

Sistema de segundo orden

$$G_{planta} = \frac{3.855 \times 10^{-6}}{s^2 + 0.0015s + 3.09 \times 10^{-6}} \quad (4.93)$$

Tanto la respuesta del sistema como el nivel en los tanques, es visualizado a través de la pantalla. Cabe aclarar que la interfaz solo permite poder un setpoint de hasta 28 cm para evitar reboses y un mal funcionamiento.

El funcionamiento es idéntico al caso anterior. Una vez que se selecciona trabajar con este sistema, la unidad carga el tanque N°1 hasta un nivel de 24 cm y bloquea los botones. Esto se indica a través de una leyenda en la pantalla. Luego el sistema esta apto para ser manipulado. Para detener es necesario apretar dos veces el botón STOP.

Otra característica que presenta este modo de funcionamiento, es que dispone de la posibilidad de activar o desactivar la bomba para que ingrese caudal al tanque N°1 y de esta manera el control sobre el tanque N°2 se prolongue por mas tiempo. Esto se ejecuta desde el botón ubicado en la parte inferior izquierda de la pantalla.

4.9.2. Modo Manual

En este modo de operación, la unidad permite ser manipulada por el usuario de acuerdo a sus especificaciones y diseño de sistemas de control sobre los procesos de nivel. Una de las particularidades y exclusividad que presenta esta unidad, es la de utilizar los actuadores independientes con cualquier Microcontrolador.

Dentro de este modo de operación, se permite realizar las siguientes funciones:



Figura 4.55: Display Modo Manual. [32]

- Activación de actuadores para diseño libre.
- Encender y/o apagar la bomba a través de la pantalla.
- Mover la Electroválvula a través de la pantalla.
- Visualizar el nivel del líquido a través de leds indicadoras.
- Visualizar el nivel del líquido en los tanques en la pantalla.
- Encender y/o apagar la bomba de forma remota a través de una aplicación móvil.
- Mover la Electroválvula de forma remota a través de una aplicación móvil.

4.9.2.1. Modo Libre

Como se menciono anteriormente, este trabajo permite el uso de las diferentes partes de la unidad con cualquier microcontrolador, como PLC, entre otros. Es decir, es ideal para realizar practicas en lo referido a los sistemas de control, tanto en tiempo continuo como discreto.

Para hacer uso de este entorno, solo basta con pinchar los actuadores que se desean utilizar desde la pantalla y subir el tablero para hacer la conexión al controlador secundario. Se aclara que cada actuador actúa de forma independiente.

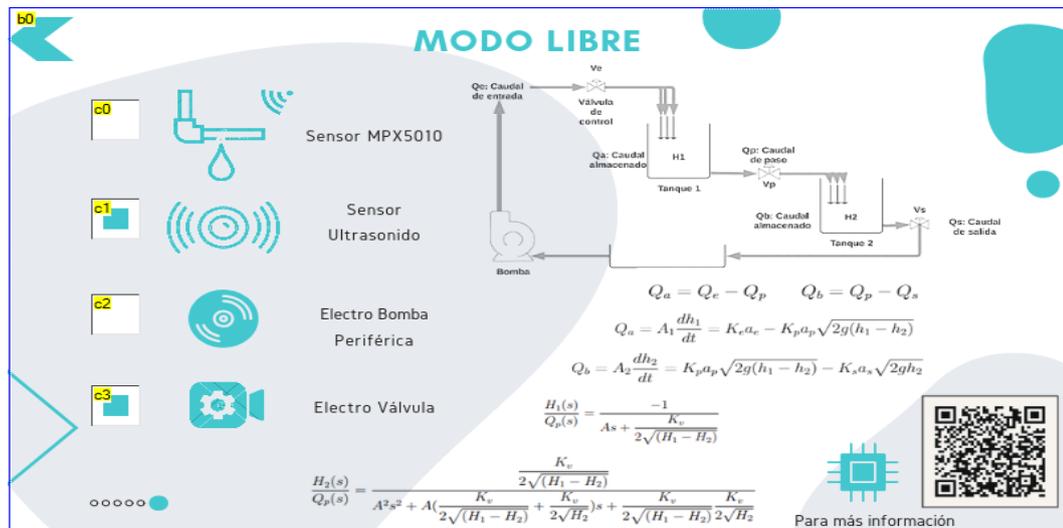


Figura 4.56: Display Modo Libre. [32]

Características de uso

- Salida analógica **Sensor de presión MPX**: 0.2 - 4.7V – 0.4 - 9.4V
- Entrada/Salida digital **Sensor de ultrasonido HC-SR04**.
- Entrada analógica **Electroválvula**: 0-5V - 0-10V.
- Entrada Digital Relé estado sólido **Bomba Periférica**.
- Potenciómetro Regulador de caudal **Bomba Periférica**.

En lo concerniente al uso de la electroválvula, internamente se realiza un control desde el Microcontrolador primario (Mega 2650), en donde, como parte de unos de los objetivos planteados, al aplicarle una tensión el controlador lee el conversor ADC (0 a 1023), y ubica la electroválvula de acuerdo al control interno del motor (0 a 1000 pasos).

Al tratarse de una unidad didáctica, se creó un tablero de conexión a través de un automatismo de subida y bajada. De manera que sea controlado exclusivamente de manera analógica por medio de pulsadores y así evitar el uso de memoria del MCU.

En la figura 4.56, se observa un código QR¹ en cual redirecciona a la pagina donde se encuentra toda la información del proyecto.

¹Un código QR es un código de barras bidimensional cuadrada que puede almacenar los datos codificados.



Figura 4.57: Tablero Didáctico. [32]

4.9.2.2. Modo Comandos

Este entorno fue pensado de manera que el usuario pueda manipular los actuadores de forma independiente, permitiendo establecer condiciones iniciales en diferentes trabajos prácticos.

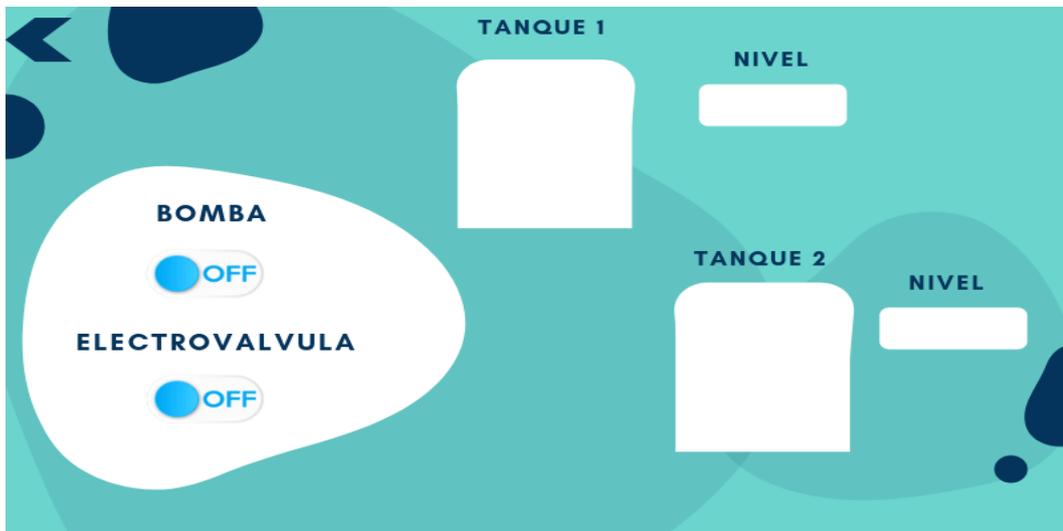


Figura 4.58: Display Modo Comandos. [32]

Además nos permite observar el nivel del líquido en los diferentes tanques, y así mismo interactuar con el sistema de manera local desde la pantalla touch.

4.9.2.3. Modo Bluetooth

Otra de las particularidades y exclusividades que presenta este proyecto, es el uso de la unidad en forma **remota**.

Esto se realiza a través del protocolo de comunicación IEEE 802.15. Estableciendo una comunicación de la unidad con una aplicación de diseño propio, para dispositivos portátiles con sistema operativo Android. Disponible en el enlace del código QR para su descarga.

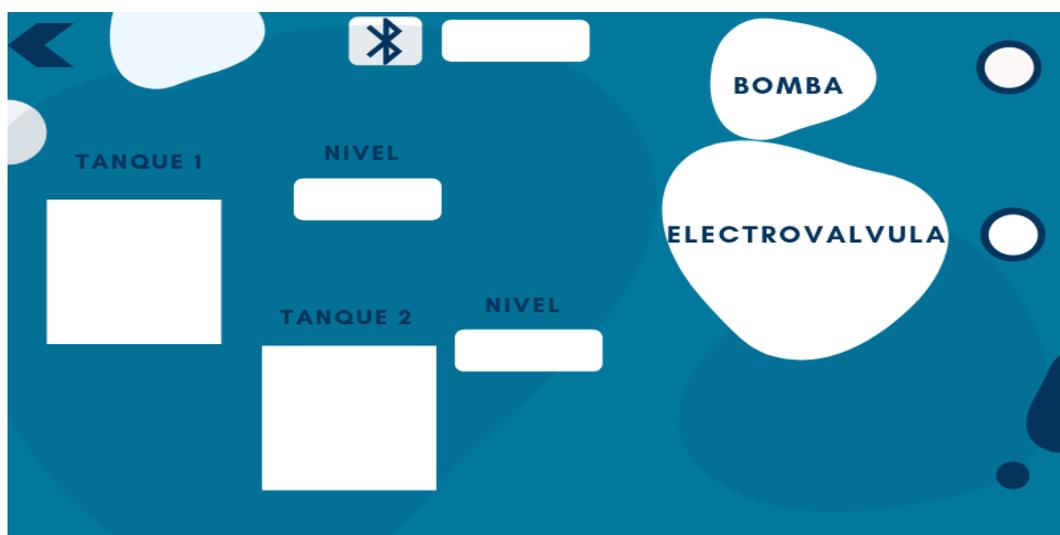


Figura 4.59: Display Modo Bluetooth. [32]

Las ventajas que esta comunicación dispone, es la de poder establecer condiciones iniciales para trabajos prácticos de manera remota y **monitorear** el comportamiento que el sistema presente sin necesidad de estar físicamente frente a la unidad.

Capítulo 5

Trabajos Prácticos

5.1. Caracterización del sistema

Ejercicio N° 1

-Determine la función de transferencia de la electroválvula del sistema.

Para caracterizar la electroválvula, relacione las variables, **tensión de entrada** versus **caudal de salida** de la siguiente manera:

1. Cargue el tanque N°1 con agua, usando el entorno manual, hasta que alcance un nivel de 20 cm.
2. Usando el entorno libre, aplique un voltaje de 1v y varíe en 0.5v hasta llegar a los 5v la tensión de entrada a la electroválvula.
3. Registre el tiempo, con cada voltaje aplicado, que tarda en llenarse el tanque N°2 a un nivel de 10 cm.
4. Repita este procedimiento al menos 3 veces, para así obtener un promedio del tiempo y una respuesta más precisa.
5. Obtenga el caudal de salida para cada tensión a través de: $Q = \frac{\text{Volumen}[m^3]}{\text{tiempo}[s]}$

El volumen del tanque N°2 es: $V_2 = 10 \text{ cm} \times 10 \text{ cm} \times 17.5 \text{ cm} = 1.75 \text{ L}$.

6. Complete el siguiente cuadro.

Voltaje de entrada [V]	t_1 [s]	t_2 [s]	t_3 [s]	t_{prom}	$Q = \frac{Volumen[m^3]}{t[s]}$
1					
1.5					
2					
2.5					
3					
3.5					
4					
4.5					
5					

7. Con los datos obtenidos puede proceder de 2 maneras para obtener la función de transferencia.
- Transcriba a Matlab y utilice el comando IDENT.
 - Analíticamente a través de Regresión Lineal. (Puede comprobar el resultado con Curve Fitting Tool de Matlab)

Ejercicio N°2

-Encuentre la función de transferencia del tanque N°2.

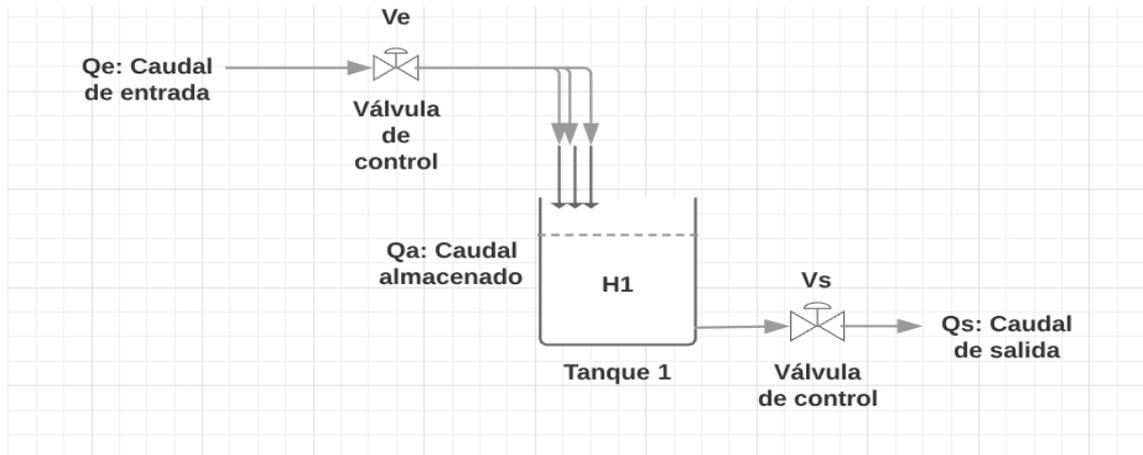


Figura 5.1: Modelo sistema de primer orden.

Partiendo de que, el caudal de entrada menos el caudal de salida, durante un intervalo pequeño de tiempo, es igual a la cantidad almacenada en el tanque. Matemáticamente esto es:

$$C.dh = (Q_e - Q_s).dt$$

$$Q_s = \frac{h}{R}$$

$$RC \frac{dh}{dt} + h = RQ_e$$

$$\frac{H_s}{Q_e} = \frac{R}{RCs + 1}$$

R: Representa la **resistencia hidráulica** a la salida del tanque.

C: Representa la **capacitancia** del tanque, o lo que es lo mismo, el área transversal del tanque.

El área del tanque es: 10 cm x 17.5 cm = 175 cm².

Para determinar la resistencia hidráulica, realice el siguiente método experimental. Este método, consiste en obtener el valor de oposición o resistencia que ofrece la válvula manual de drenaje. Esta válvula es una llave tipo globo de 1/2" y de 1/4 de vuelta.

1. Llene el tanque N°2 en su nivel de operación (por ejemplo 20-10 cm). Se recomienda poner un tapón en el codo de desagote para evitar el cierre de la válvula para el llenado del tanque en cada medición.
2. Tome el tiempo que se demora en evacuar el agua hasta el nivel deseado.
3. La válvula de drenaje se debe encontrar cerrada en un 75 % aproximadamente.
4. Se considera el flujo, como un flujo laminar, por lo que la resistencia viene dada por:

$$R = \frac{dH}{dQ} = \frac{\Delta H}{\Delta Q} = \frac{\Delta H}{\frac{\Delta H \cdot A}{t}} = \frac{t [s]}{A [m^2]}$$

5. Realice el procedimiento al menos 3 veces para cada nivel de operación y complete el siguiente cuadro.

Nivel [cm]	t ₁ [s]	t ₂ [s]	t ₃ [s]	t _{prom}	R = $\frac{t[s]}{A[m^2]}$
20-15					
20-10					
20-5					
15-10					
15-5					
10-5					
5-3					

6. De la tabla encontrada anteriormente, analice cada valor obtenido y decida sobre un valor de Resistencia Hidráulica. Explique el motivo de su elección. Tenga en cuenta que un sistema no lineal se aproxima a un sistema lineal en cercanías a un punto de equilibrio.
7. Obtenga la función de transferencia del sistema de primer orden.

Ejercicio N°3

- Hallar la función de transferencia que describa al sensor de presión MPX5010DP ubicado en el tanque N°2.

En este caso se relacionan las variables, **altura** en cm, y el **voltaje** de salida del sensor.

$$G_{sensor} = \frac{V_{out}(s)}{H(s)}$$

Puede utilizar la hoja de especificaciones del sensor para tener referencia. Recuerde que 10KPa = 102 cm de agua.

El proceso para la obtención de la función, es el siguiente:

1. Utilice el entorno manual y llene el tanque hasta un valor de operación deseado, por ejemplo a 20 cm.
2. Tome muestra del voltaje desde los terminales del sensor en el tablero en modo libre.
3. Repita el procedimiento en al menos 10 niveles de altura.
4. Tener en cuenta su curva de histeresis, haciendo un barrido de lecturas subiendo y bajando el nivel.
5. Realice una tabla similar a la siguiente:

Nivel bajando [cm]	Voltaje[v]	Nivel subiendo [cm]	Voltaje[v]
25		0	
20		5	
15		10	
10		15	
5		20	
0		25	

6. Al obtener 2 grupos de datos, utilice el comando IDENT de Matlab, compare las curvas y obtenga la función de transferencia del sensor.

5.2. Análisis de la respuesta transitoria y error en estado estacionario

La respuesta en el tiempo de un sistema de control consta de dos partes: la respuesta transitoria y la respuesta en estado estacionario. La respuesta transitoria se refiere a la que va del estado inicial al estado final.

Por respuesta en estado estacionario se entiende la manera como se comporta la salida del sistema conforme t tiende a infinito. Por tanto, la respuesta del sistema $c(t)$ se puede escribir como $c(t) = ctr + css(t)$ donde el primer término del miembro derecho de la ecuación es la respuesta transitoria y el segundo término es la respuesta en el estado estacionario.

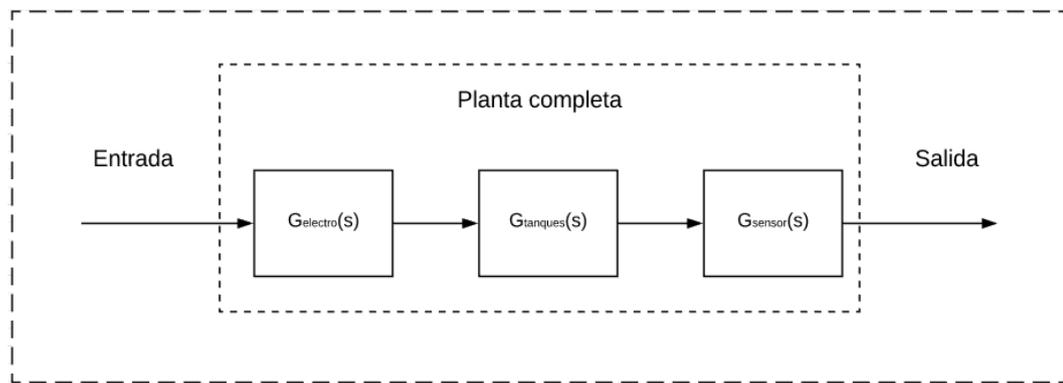


Figura 5.2: Diagrama de bloques de la planta.

Ejercicio N°1

Con las funciones de transferencias obtenidas en el practico anterior, encuentre la función de transferencia de lazo cerrado del sistema:

a) Explique el comportamiento del sistema a partir de las especificaciones de la respuesta temporal (tiempo de subida, sobre impulso, etc), frente a una entrada escalón.

b) Expresa la función de transferencia del sistema, en termino de Ganancia-Polos-Ceros y analice el comportamiento a partir de estos parámetros. ¿Se anulan? ¿Cual polo causa la atenuación?

c) Utilice el comando Rlocus en Matlab para visualizar el lugar de raíces y analizar lo expresado anteriormente.

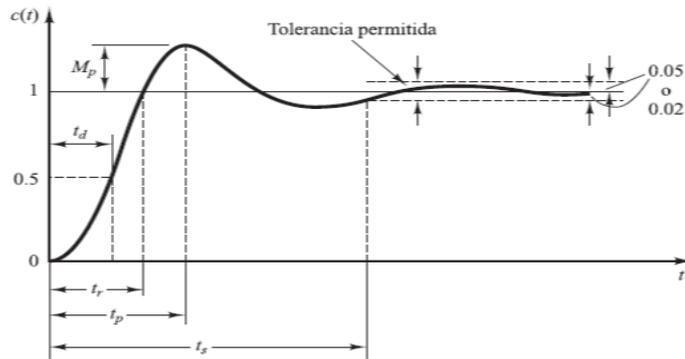


Figura 5.3: Especificaciones de respuesta transitoria.

Ejercicio N°2

a) Calcular el error estacionario, ante las señales típicas de pruebas (escalón, rampa, parábola), para la planta encontrada anteriormente.

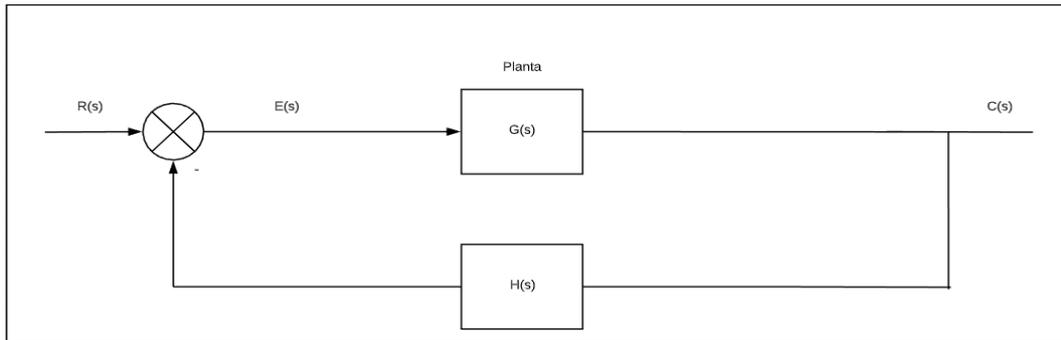


Figura 5.4: Diagrama de bloques del sistema.

$$H(s) = 1$$

$$E(s) = \frac{1}{1 + G(s)} R(s)$$

b) Encuentre el error del sistema anterior, para una entrada escalón, donde $H(s) =$ función de transferencia del sensor encontrada.

c) Observe la figura 5.6. Hallar el valor de K , frente a una entrada escalón, de manera que el error estacionario sea ≤ 0.1 .

$$H(s) = 1$$

$G(s) =$ Función de la planta del sistema.

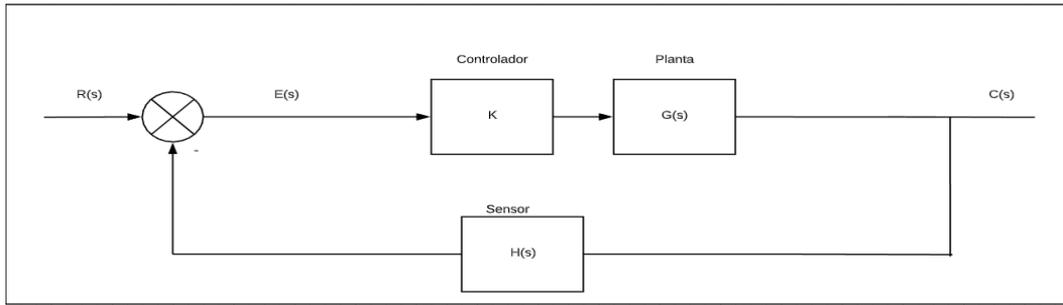


Figura 5.5: Diagrama de bloques del sistema con compensador K.

d) Encontrar el error del siguiente sistema.

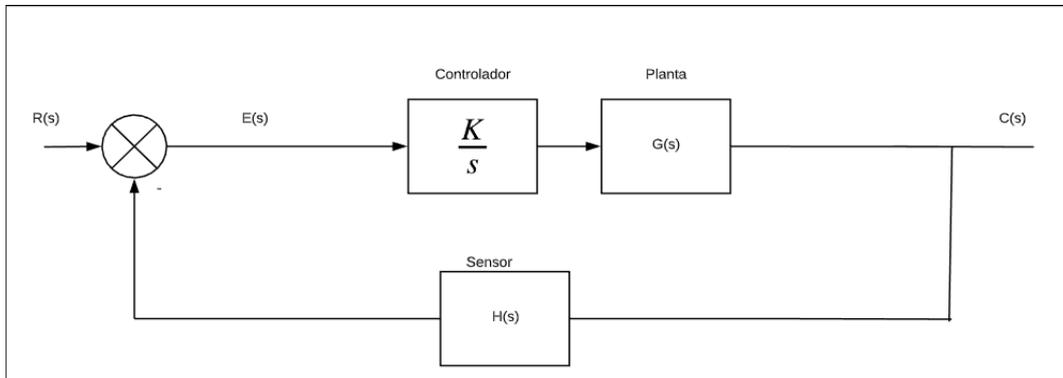


Figura 5.6: Diagrama de bloques del sistema con compensador $\frac{K}{s}$

$H(s)$ = Función del sensor de presión encontrada.

$G(s)$ = Función del sistema de primer orden encontrado.

5.3. Análisis de estabilidad

El problema más importante en los sistemas de control es, la estabilidad. Tanto absoluta como relativa, nos brindan información sobre en que condiciones el sistema se vuelve inestable, o siendo inestable, como estabilizarlo. De ahí la importancia de diferentes métodos para conocer la estabilidad de los sistemas.

Ejercicio N°1

- Aplique el criterio de estabilidad de Routh a los siguientes sistemas.

a) $G(s) = \frac{-0.0004175}{s + 0.0652}$ - Donde $H(s) = 1$

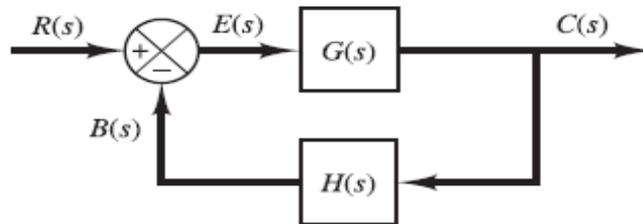


Figura 5.7: Sistema en lazo cerrado. [17]

b) $G(s) =$ Función encontrada en practico anterior. - $H(s) = 1$

c) $G(s) = \frac{4.687 \times 10^{-5}}{s^2 + 0.1927s + 0.009}$ - Donde $H(s) = 1$

Ejercicio N°2

El criterio de estabilidad de Routh tiene una utilidad limitada en el análisis de un sistema de control lineal, sobre todo porque no sugiere cómo mejorar la estabilidad relativa ni como estabilizar un sistema inestable. Sin embargo, es posible determinar los efectos de cambiar uno o dos parámetros de un sistema si se examinan los valores que producen inestabilidad.

- Considere el sistema de la figura 6.8 y determine el rango de K.

a) $G(s) = \frac{-0.0004175}{s + 0.0652}$ - Donde $H(s) = 1$

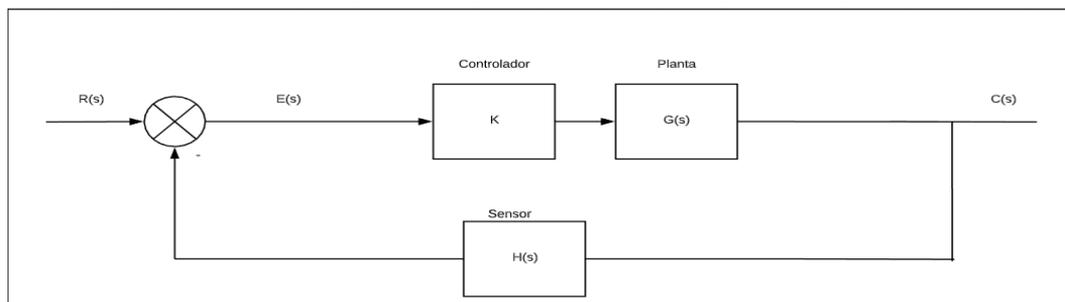


Figura 5.8: Diagrama de bloques del sistema con compensador K.

b) $G(s) =$ Función del sistema encontrada - Donde $H(s) =$ Función Sensor encontrada.

c) $G(s) = \frac{4.687x10^{-5}}{s^2 + 0.1927 + 0.009}$ - Donde $H(s) = 1$

Ejercicio N°3

Considérese el sistema de la Figura 6.9. Determinar el rango de valores de K para la estabilidad.

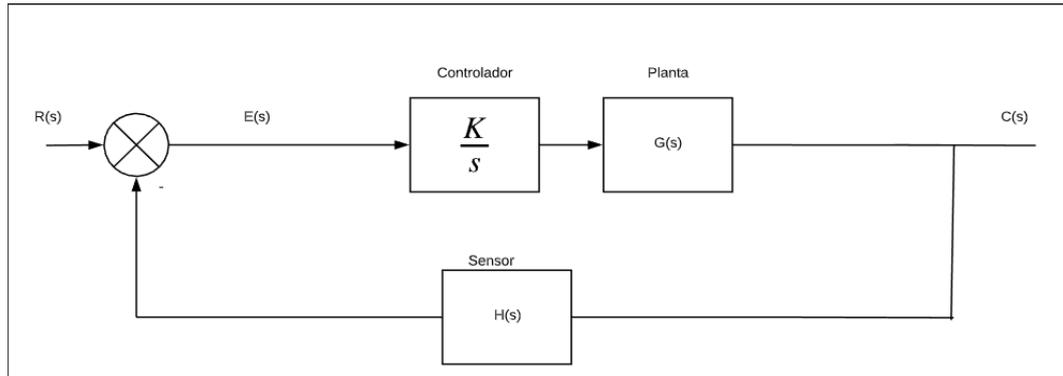


Figura 5.9: Diagrama de bloques del sistema con compensador $\frac{K}{s}$

a) $G(s) = \frac{-0.0004175}{s + 0.0652}$ - Donde $H(s) = 1$.

b) $G(s)$ = Función de transferencia encontrada. $H(s)$ = Función del sensor encontrada.

c) $G(s) = \frac{4.687x10^{-5}}{s^2 + 0.1927 + 0.009}$ - Donde $H(s) = 1$

5.4. Diseño de controlador tipo PI

Ejercicio N°1

A partir de la ecuación del sistema de control de nivel de un tanque a lazo cerrado encontrada anteriormente, diseñe un controlador tipo PI con las siguientes especificaciones:

- Tiempo de subida < 5 s .
- Tiempo de establecimiento < 10 s .
- Sobreimpulso < 10 %

Para encontrar las constantes de este controlador, se puede utilizar una de las siguientes formas:

- Por cancelación de polos y ceros.
- Por Reglas de Ziegler - Nichols.
- Por PID Tuner de Matlab.

Ejercicio N°2

Una vez encontrado el controlador tipo PI, obtenga la función de transferencia de lazo cerrado junto con el controlador, ejecute el comando STEP en Matlab y observe ahora su comportamiento. ¿ Responde a las especificaciones de diseño?

5.5. Sistema de control de 2º orden

Considere el sistema de control de tanques en cascada de la siguiente figura.

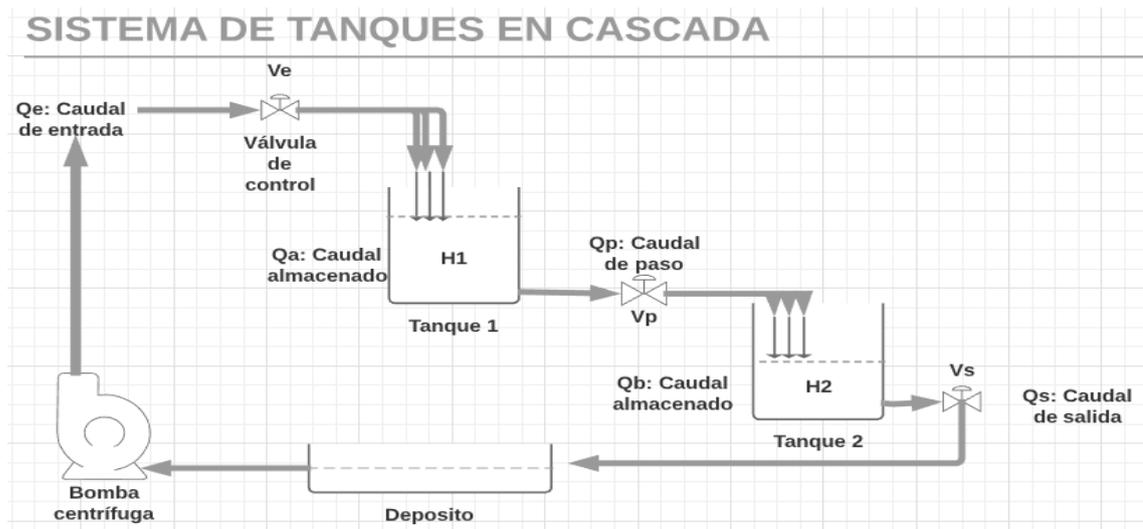


Figura 5.10: Esquema básico de un sistema de segundo orden . [32]

Un proceso de segundo orden, son aquellos procesos que constan de dos elementos capacitivos, y dos elementos resistivos. Podemos decir que los procesos de segundo orden, se clasifican como no interactivos e interactivos.

La figura 5.10, muestra un proceso **no interactivo** de segundo orden. Se puede observar, que en el proceso no interactivo, el nivel del tanque dos, no tiene impacto sobre la tasa de flujo que sale del tanque uno.

La función de transferencia entre las variables de salidas del proceso de segundo orden no interactivo, puede ser deducida utilizando la relación del balance de masa

en estado estacionario, primero utilizada para encontrar la ecuación diferencial, que describa el proceso en términos de la variable independiente t . Luego se utiliza la transformada de Laplace para encontrar la función de transferencia en función de la variable compleja s .

La función de transferencia que relaciona el nivel del líquido del tanque n°1 con respecto a la válvula de paso V_p , esta dada por:

$$\frac{H_1}{Q_p} = \frac{-R_e}{R_e C_1 s + 1}$$

Donde:

- R_e = Resistencia de paso de válvula de control V_e
- C_1 = Capacitancia del tanque n°1, o lo que es lo mismo el área transversal del tanque

La función de transferencia total que relaciona el nivel de líquido del tanque n°2 con el caudal de la válvula de paso V_p , es:

$$\frac{H(s)}{Q_p(s)} = \left[\frac{-R_e}{R_e C_1 s + 1} \right] \cdot \left[\frac{R_s}{R_s C_2 s + 1} \right]. \quad (5.1)$$

$$\frac{H(s)}{Q_p(s)} = \left[\frac{-R_e R_s}{R_e C_1 R_s C_2 s^2 + (R_e C_1 + R_s C_2) s + 1} \right]. \quad (5.2)$$

Ejercicio N°1

a) Con la resistencia hidráulica obtenida en prácticos anteriores, encuentre la función de transferencia a partir de la ec. (5.2). Considere a $C_1 = C_2$ y $R_e = R_s$

b) Aplique una entrada tipo escalón a la función de transferencia encontrada y responda. ¿ la planta es estable?

c) Utilice el comando Rlocus y visualice el lugar de raíces de la planta.

d) Una vez encontrada la función de transferencia de la planta y analizada, obtenga la función de transferencia a lazo cerrado del sistema. Para ello, utilice las funciones de transferencias del sensor y electroválvula encontrada anteriormente. ¿ El sistema es estable ? ¿ Si no lo es, como podría compensarlo?

5.6. Diseño de controlador tipo PID

Ejercicio N°1

A partir de la ecuación del sistema de control de nivel de dos tanques a lazo cerrado encontrada anteriormente, diseñe un controlador tipo PID con las siguientes especificaciones:

- Tiempo de subida < 4 s .
- Tiempo de establecimiento < 8 s .
- Sobreimpulso $< 15\%$

Para encontrar las constantes de este controlador, se puede utilizar una de las siguientes formas:

- Por cancelación de polos y ceros.
- Por Reglas de Ziegler - Nichols.
- Por PID Tuner de Matlab.

Ejercicio N°2

a) Una vez encontrado el controlador tipo PID, obtenga la función de transferencia de lazo cerrado junto con el controlador, ejecute el comando STEP en Matlab y observe ahora su comportamiento. ¿ Responde a las especificaciones de diseño?.

b) ¿Existen diferencias entre este tipo de controlador y un controlador tipo PI para este sistema?. Analice con respecto a la respuesta en estado estacionario.

c) Al momento de implementar el controlador en un Microcontrolador, ¿ cual cree que seria más eficiente?

5.7. Discretización de sistemas

Para realizar un sistema de control e implementarlo en cualquier MCU (microcontrolador), es necesario que las funciones de transferencias, tanto de las plantas como de controladores, se encuentre discretizada y así responder con las especificaciones requeridas del sistema.

Existen diferentes métodos para realizar la discretización, a continuación se mencionan algunos:

1. **Simulaciones invariantes y/o con Retenedor.**

- Respuesta invariante al impulso.
- Respuesta a la entrada escalón o equivalencia ZOH. (Retenedor de orden cero).

2. **Por aproximaciones o Transformaciones $s = F(z)$.**

- Integración Rectangular (Euler).
- Integración Rectangular Retrasada (Euler hacia atrás).
- Integración Trapezoidal, conocida como Bilineal o Regla de Tustin.

Por otro parte, las propiedades del controlador digital dependen principalmente de:

- El periodo de muestreo T .
- Método de Digitalización utilizado.

Ejercicio N°1

a) Represente al sistema de primer orden, utilizando el método de respuesta al escalón o más conocido como Retenedor de orden cero.

b) Represente al sistema de segundo orden, utilizando el método de respuesta al escalón o más conocido como Retenedor de orden cero. Utilice la siguiente expresión.

$$G(z) = z[G_{zoh}(s)G(s)] = (1 - z^{-1})z\left[\frac{G(s)}{s}\right] \quad (5.3)$$

c) Represente al sistema de primer orden, ejecutando el comando **c2d** de Matlab, con un periodo de muestreo de $T_m = 0.01s$ y el método de digitalización ‘tustin’.

Ejercicio N°2

a) Tome las funciones de transferencias encontradas en los incisos anteriores y visualice las respuesta ante una entrada escalón. Especifique su comportamiento a partir de su tiempo de subida y establecimiento, sobreimpulso, etc.

b) Con respecto a la estabilidad del sistema. ¿ Sigue siendo inestable o paso a ser estable luego de la discretizacion ?.

c) Dibuje el lugar de raíces de cada función encontrada con el comando **Rlocus** de Matlab.

5.8. Ecuaciones en Diferencias

Lo que se programa en cualquier microcontrolador son aquellas ecuaciones en diferencias del controlador a través de arreglos (vectores). Consideremos el diagrama de bloques de la figura 5.11 :

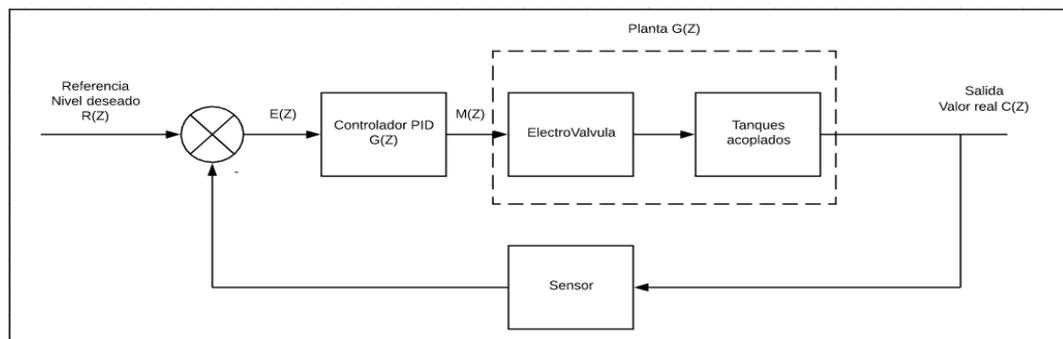


Figura 5.11: Diagrama de bloques Discreto. [32]

Se observa que la señal $M(z)$ es la encargada de generar la acción de manipulación del controlador que actúa directamente en el actuador. Podemos representar al controlador como:

$$G_c(z) = \frac{M(z)}{E(z)} \quad (5.4)$$

Donde:

- $E(z)$ = Señal Error (proveniente de la diferencia del Set Point con el valor real de la salida).
- $M(z)$ = Señal de salida del controlador.

Ejercicio N°1

Encuentre la ecuación en diferencias del controlador tipo PI encontrado en prácticos anteriores, a partir de:

$$\frac{M(z)}{E(z)} = K_p + K_i \frac{T_m}{z-1} \quad (5.5)$$

$$\frac{M(z)}{E(z)} \cdot (z-1) = K_p z - (K_p + K_i T_m)$$

Ejercicio N°2

Encuentre la ecuación en diferencias del controlador tipo PID encontrado en prácticos anteriores, a partir de:

El controlador PID a implementar en el sistema de segundo orden (2 tanques) esta dado por:

$$\frac{M(z)}{E(z)} = K_p + K_i \frac{T_m}{z-1} + \frac{1}{T_f + T_m/(z-1)} \quad (5.6)$$

donde:

- K_p = Constante Proporcional.
- K_i = Constante Integral .
- K_d = Constante Derivativo .
- T_f = Tiempo de filtro derivativo.
- T_m = Tiempo de Muestreo.

Capítulo 6

Resultados y Análisis

6.1. Pruebas de desempeño

Una vez construida la unidad didáctica para sistemas de control con enfoque en procesos de nivel, se efectuaron diversas pruebas para los diferentes entornos y actuadores que posee la unidad.

Los resultados que a continuación se muestran son el fiel reflejo de las características del equipo.

6.1.1. Electroválvula

Al tratarse de una unidad didáctica diseñada exclusivamente para el desarrollo de sistemas de control con enfoques en procesos de nivel, se efectuó pruebas a la electroválvula desde el modo libre.

Esta prueba consistió en aplicar tensión entre 0 y 5 v, e ir variando la tensión en 0.5 v , para obtener el caudal que entrega en cada valor de tensión aplicado. Es decir, al aplicar una tensión la válvula tiene asociada un área de apertura lo que permite el paso del liquido. Los valores obtenidos se muestran en tabla 6.1.

Se tomaron en cuenta algunas consideraciones a tomar las mediciones:

- Se llenó el tanque N°1 con el entorno automático (control 2 tanques) hasta un nivel de 25 cm.
- Se tomó el tiempo que tarda en llenarse el tanque N°2 a un nivel de 10 cm.
- Volumen tanque N°2 = $10 \text{ cm} * 10 \text{ cm} * 17.5 \text{ cm} = 1750 \text{ cm}^3$

- Se repitió 3 veces cada medición y se obtuvo un tiempo promedio para mitigar parte del error de lectura.

Voltaje de entrada [V]	t ₁ [s]	t ₂ [s]	t ₃ [s]	t _{prom}	Q= $\frac{Volumen[cm^3]}{t[s]}$
1	79:24	81:22	78:45	79:63	22.05
1.5	24:09	24:36	24:32	24:25	72:16
2	19:61	20:96	20:20	20:25	86.41
2.5	17:99	17:29	17:32	17:53	99.82
3	16:63	16:55	16:89	16:69	104.85
3.5	16:10	16:34	15:82	16:08	108.32
4	15:97	16:01	16:04	16:00	109.37
4.5	16:03	15:52	16:04	15:86	110.37
5	15:92	15:66	15:80	15:79	110.82

Tabla 6.1: Caracterización del caudal de la válvula.

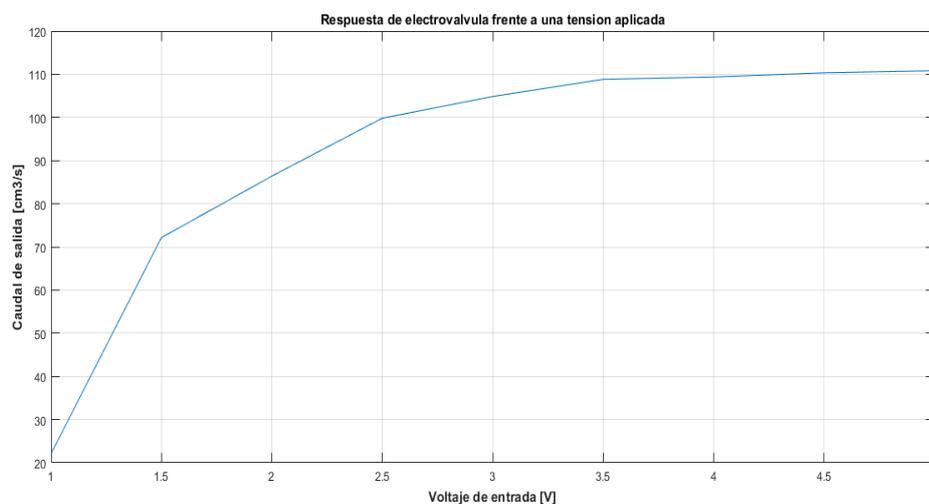


Figura 6.1: Caracterización del caudal de electroválvula. [32]

Podemos observar de la gráfica anterior que, hasta un voltaje de 3.5, el grado de abertura de la válvula es mas o menos lineal (70 % de abertura), por lo que resulta necesario trabajar en esa área.

Esta prueba nos permite contemplar que la electroválvula en modo libre funciona perfectamente ante una tensión aplicada. A su vez, las mediciones obtenidas nos brindaron la forma de respuesta que presenta, útil para los ensayos en trabajos futuros.

6.1.2. Bomba de agua

Como este actuador no se tuvo en cuenta a la hora de la caracterización del sistema. Se realizaron pruebas de desempeño desde el punto de vista del disparo del triac y activación del relé de estado sólido.

Para controlar el disparo del triac y así modificar el ángulo con que toma la onda sinusoidal (a menor ángulo mayor es el paso de la onda sinusoidal), se desplaza el potenciómetro ubicado en el tablero didáctico. Este accionar influye indirectamente en el caudal que ingresa a los tanques. Por lo que se relacionó el caudal que entrega frente al porcentaje de la ubicación del potenciómetro. Esto se observa en la tabla 6.2

Se tuvieron en cuenta algunas consideraciones a la hora de tomar las mediciones:

- Se tomó el tiempo que tarda en llenarse el tanque N°1 desde 5 cm a 20 cm.
- El tanque reservorio siempre estuvo con agua para evitar zonas muertas.
- Al cambiar de ubicación el potenciómetro, ejemplo de 75 % a 100 %, se lo realizó con la bomba en funcionamiento, en algunos casos.
- En los casos de que el caudal fue mayor al caudal de la válvula de desagote, se utilizó el pulsador golpe de puño de emergencia para detener la bomba.
- Volumen tanque N°1 = 10 cm * 15 cm * 17.5 cm = 2625 cm³
- Se repitió 3 veces cada medición y se obtuvo un tiempo promedio para mitigar parte del error de lectura.

Porcentaje de entrada [%]	t ₁ [s]	t ₂ [s]	t ₃ [s]	t _{prom}	Q= $\frac{Volumen[cm^3]}{t[s]}$
0	7:48	7:72	8:10	7:76	338.27
25	15:27	14:18	14:62	14:69	178.69
50	18:23	17:14	17:45	17:60	149.14
75	20:44	23:02	19:22	20:89	125.65
100	25:09	26:05	25:46	25:53	102.82

Tabla 6.2: Caracterización del caudal de Bomba periférica.

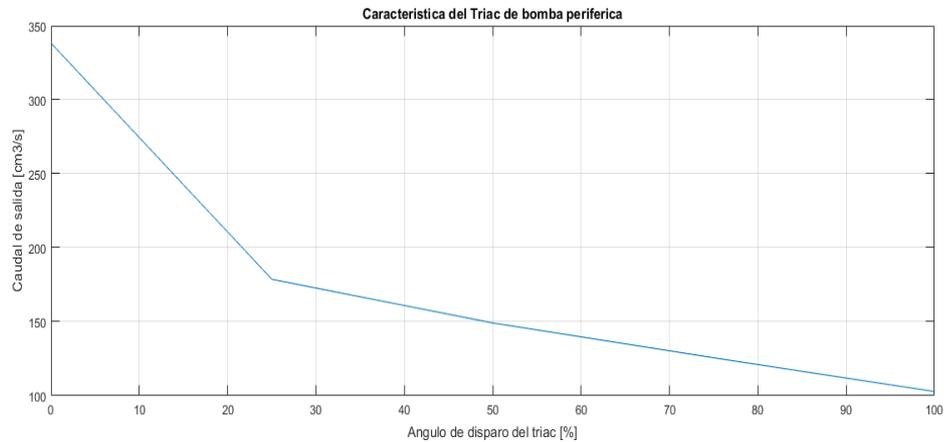


Figura 6.2: Caracterización del triac de bomba periférica. [32]

De la figura 6.2, podemos concluir que la bomba periférica al desplazar el potenciómetro presenta una respuesta lineal por tramos, es decir de 0% a 25% mantiene una pendiente y luego sigue su linealidad hasta 100% con menor caudal.

Por otro lado, este actuador puede ser accionado por un Microcontrolador secundario para realizar algún tipo de control de caudal. Para ello, se aplicó una tensión de 5 V (simulando un Pin HIGH de un controlador) a los bornes del tablero para encender este actuador.

El rango de operación del disparo del relé de estado solido es de:

Tensión Mínima = 3.3 V .

Tensión Máxima = 5.2 V .



Figura 6.3: Entrada externa para el uso de bomba . [32]

6.1.3. Modo Libre

Estas pruebas se realizaron para explorar el funcionamiento de los sensores, tanto de presión como de ultrasonido ubicado en los tanques, como así también el uso de la electroválvula y bomba como actuadores.

Para probar este modo, se realizó 2 sistemas de control utilizando cada actuador de la unidad con dos microcontroladores, uno conocido como EDU-CIAA y otro comercialmente como Arduino Nano.

El primer sistema de control, se basó utilizando el tanque N°1 como planta, el sensor de **ultrasonido** como elemento de medición y la bomba de agua y electroválvula como actuadores. Se realizó un controlador tipo ON-OFF controlando el caudal de entrada a través de la bomba. El controlador consistió en estabilizar el nivel del tanque N°1 entre 10 a 12 cm.

El segundo sistema, se utilizó el tanque N°2 como planta, el sensor de **presión** como elemento de medición y la electroválvula como actuador. Se aclara que el tanque N°1 se consideró como un tanque reservorio de agua para este caso. El controlador consistió en estabilizar el nivel del tanque a una altura de entre 14 y 15 cm. Esto se hizo a partir del Microcontrolador EDU-CIAA con su poncho correspondiente como puede verse en las siguientes figura.

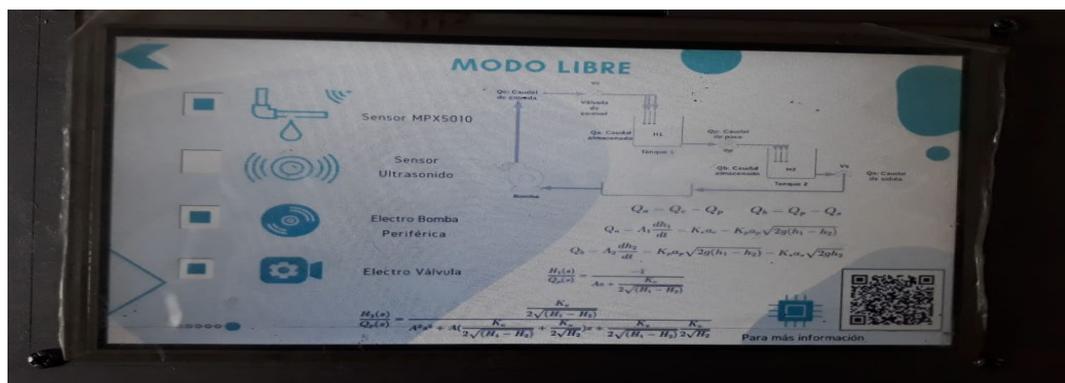


Figura 6.4: Modo libre realizando pruebas de desempeño. [32]

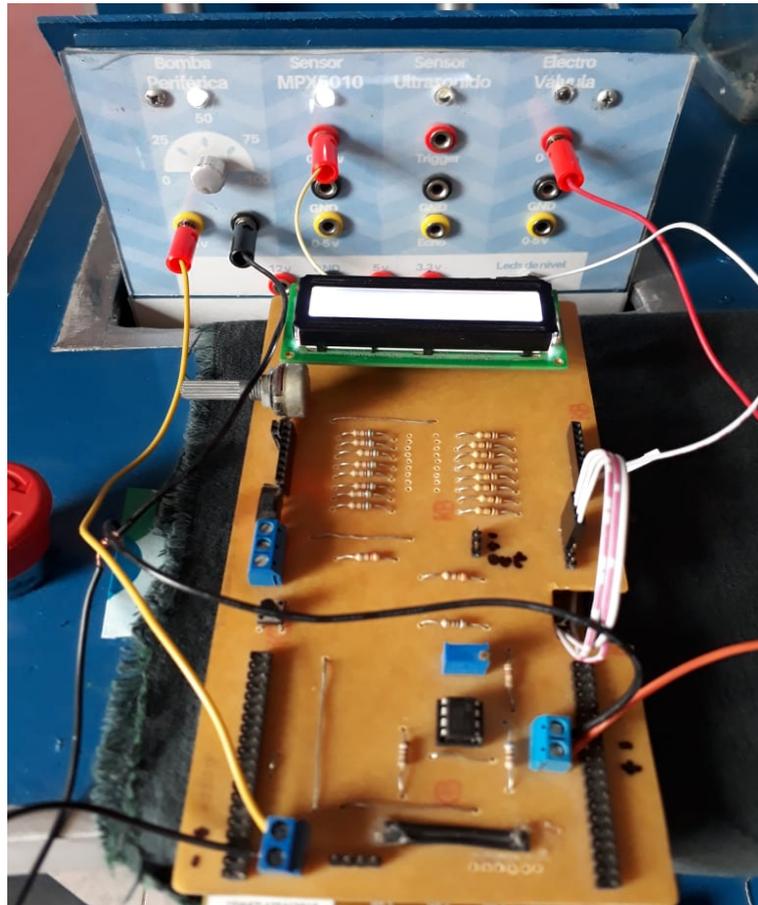


Figura 6.5: Microcontrolador secundario EDU-CIAA. [32]

6.1.4. Modo Bluetooth

En esta subsección referido al control de la unidad de forma remota, se hicieron pruebas para determinar los tiempos de respuesta de cada actuador pulsado desde la aplicación y los tiempos de muestreo de los datos enviados de la unidad hacia el dispositivo móvil.

Evento	t_1 [s]	t_2 [s]	t_3 [s]	t_{prom}
Bomba ON	2:24	2:04	1:17	1:81
Bomba OFF	2:17	1:32	2:18	1:89
Válvula ON	1:45	2:38	2:24	2:02
Válvula OFF	1:59	2:31	1:52	1:80

Tabla 6.3: Tiempo de respuesta de los actuadores.

Evento	t_1 [s]
Tanque N°1	2
Tanque N°2	2

Tabla 6.4: Tiempo de muestreo de los datos.

6.2. Sistemas de control

Debido a que este proyecto presenta una manera practica de poder caracterizar los subsistemas que hacen al sistema de control. Se realizaron diversas pruebas y simulaciones a las ecuaciones encontradas a partir del análisis teórico y a su vez a las ecuaciones encontradas a partir de mediciones tomadas a cada actuador junto a las brindadas por el libro **Ingeniería de Control Moderna 5ta Edición** Katsuhiko Ogata. Otra manera practica que se utilizó para determinar las constantes de los controladores PID fue la **segunda regla de sintonía de Ziegler-Nichols**. Esto nos permite poder contrastar si estos sistemas se asimilan y visualizar el comportamiento que estos presentan.

6.2.1. Sistema de primer orden

Para realizar el sistema de primer orden, se utilizó un solo tanque de la unidad, en este caso el tanque N°1 donde se encuentra el sensor de ultrasonido.

Pruebas del análisis teórico

- La función de transferencia del tanque es:

$$\frac{H_1(s)}{Q_p(s)} = -\frac{1}{As + \frac{K_v}{2\sqrt{H_1}}} \quad (6.1)$$

Donde tomando $H_1 = 10$ y $K_v = 55.59 \frac{cm^3}{s}$, tenemos:

$$\frac{H_1(s)}{Q_p(s)} = -\frac{5.71x10^{-3}}{s + 5.71x10^{-3}} \quad (6.2)$$

- Función de transferencia del sensor ultrasónico:

$$\frac{H_1(s)}{Volumen(s)} = -0.005714cm^3 \quad (6.3)$$

- Función de transferencia de electroválvula:

$$\frac{Q_1(s)}{V(s)} = 17.49 + \frac{39.17}{V(s)} \quad (6.4)$$

- Función de transferencia en lazo abierto:

$$G(s) = \frac{0.0005706}{s + 0.0401} \quad (6.5)$$

- Función de transferencia en lazo cerrado:

$$G(s) = \frac{0.0005706}{s + 0.04158} \quad (6.6)$$

- Función de transferencia del controlador PID :

$$G_{PID}(s) = K_p + \frac{k_i}{s} + k_d s \quad (6.7)$$

$$G_{PID}(s) = \frac{147.5s^2 + 1167.37s + 208.96}{s} \quad (6.8)$$

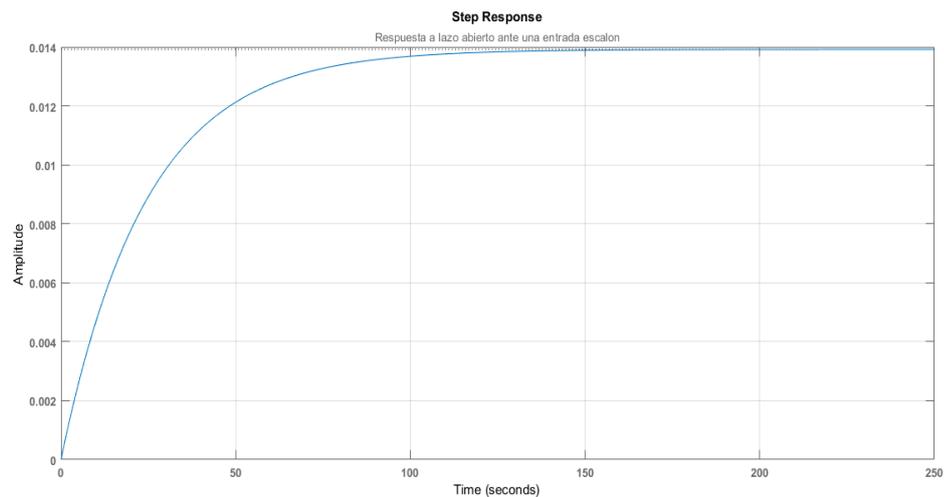


Figura 6.6: Respuesta a lazo abierto. [32]

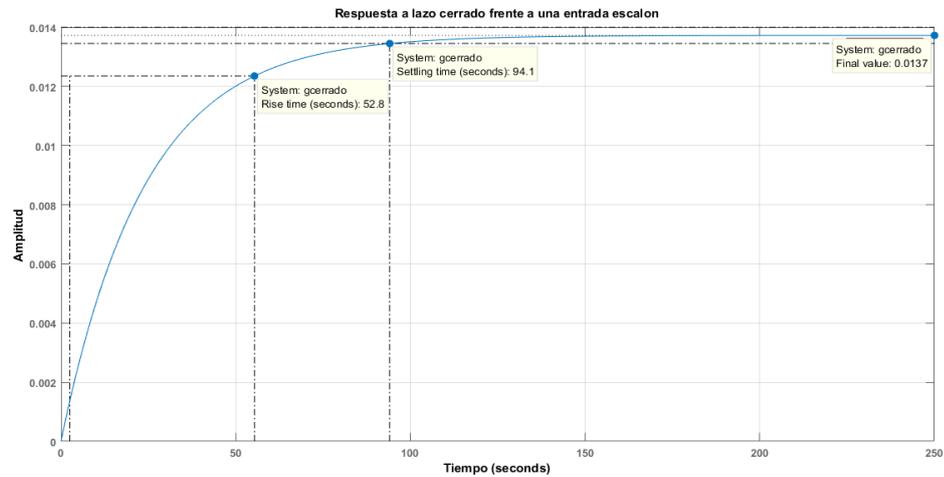


Figura 6.7: Respuesta a lazo cerrado. [32]

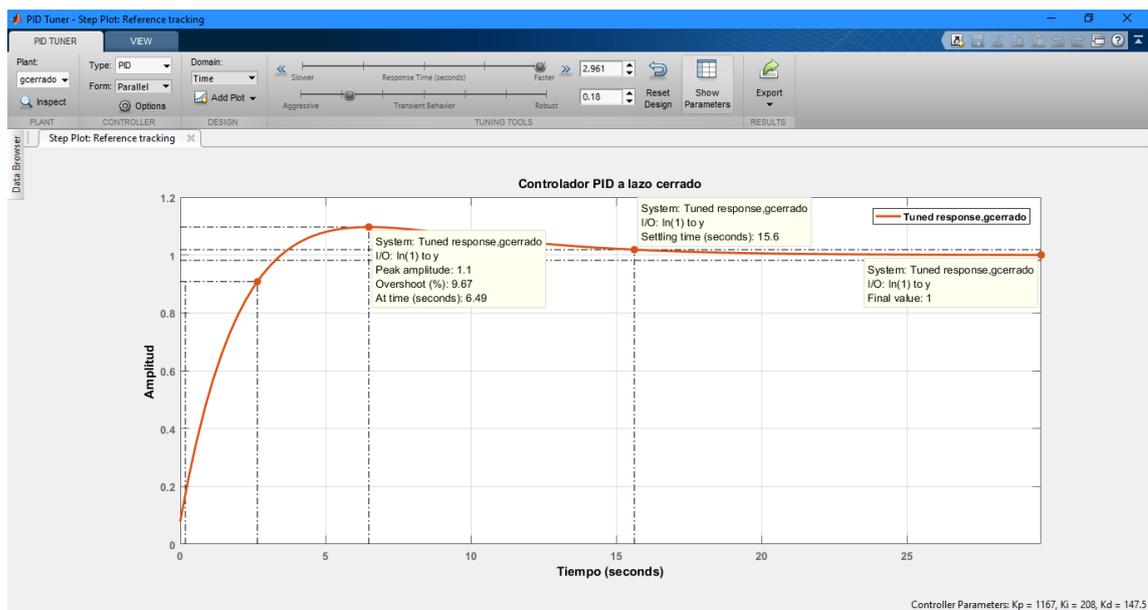


Figura 6.8: Respuesta a lazo cerrado con PID. [32]

Al aplicar estos valores a través de la pantalla obtenemos lo siguiente:

- Tiempo de establecimiento: No se estabilizó, si no que producía oscilaciones entre 13.8 y 12.1 cm .
- sobreimpulso: No presentó.

Cabe hacer la aclaración que la constante $K_i = 20800$ debido a que la pantalla no permite el ingreso de valores decimales, por lo que el controlador multiplica al valor

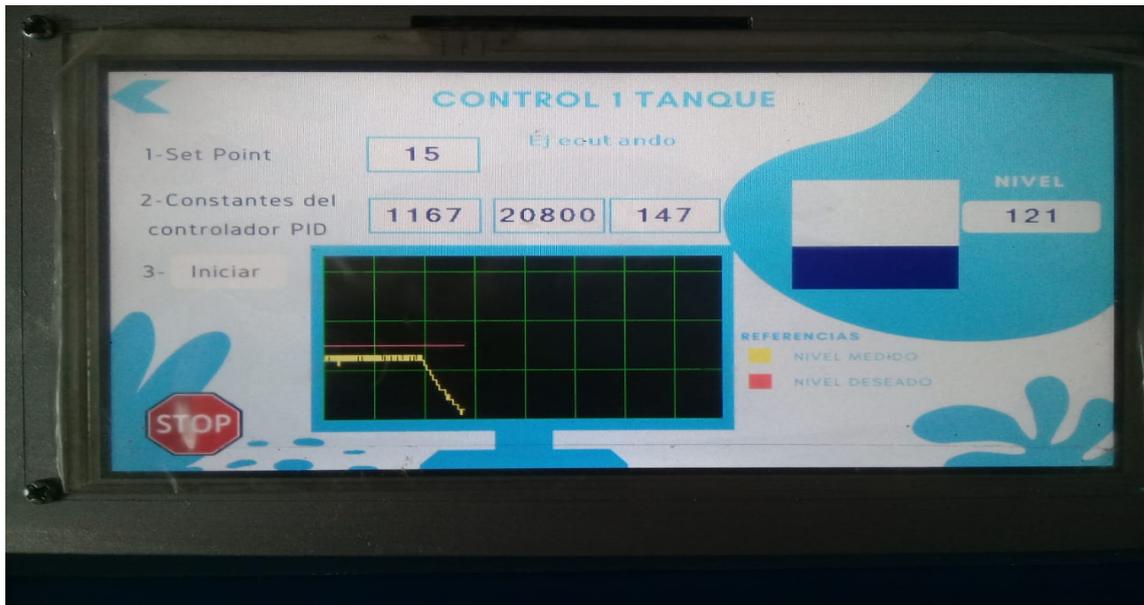


Figura 6.9: Respuesta del controlador PID teórico. [32]

de la constante por 0.01 . Dicho de otra manera, el controlador esta recibiendo la constante $K_i = 20800 \times 0.01 = 208$

Desde Ziegler-Nichols

Para realizar este apartado a través de la 2da regla de sintonía, es necesario encontrar un valor de la constante P que lleve al sistema al estado de oscilación. Es decir, este valor lleva a que los polos se ubican sobre el eje imaginario y el sistema se encuentre críticamente estable.

Este método fue realizado a partir del entorno modo automático control de 1 tanque con las siguientes condiciones:

- Válvula de entrada abierta 20 %.
- Válvula de desagote ubicada debajo del tanque N°1 totalmente cerrada.
- Válvula de desagote ubicada debajo del tanque N°2 totalmente abierta.
- Válvula de salida totalmente abierta.
- Potenciómetro del tablero didáctico en la posición '0'.

Experimentalmente se utilizó el valor de $P = 500$ y se obtuvo una oscilación sostenida entre 16.5 y 15.5 cm. A partir de esta oscilación sostenida se tomó el tiempo

que tardaba en volver a un valor de referencia, es decir, el periodo con el que se repetía. Este recibe el nombre de periodo crítico.



Figura 6.10: Periodo de oscilación vista desde la pantalla. [32]

Siguiendo la regla de Ziegler- Nichols se obtuvo los siguientes valores:

Tabla 8-2. Regla de sintonía de Ziegler-Nichols basada en la ganancia crítica K_{cr} y periodo crítico P_{cr} (segundo método).

Tipo de controlador	K_p	T_i	T_d
P	$0.5K_{cr}$	∞	0
PI	$0.45K_{cr}$	$\frac{1}{1.2}P_{cr}$	0
PID	$0.6K_{cr}$	$0.5P_{cr}$	$0.125P_{cr}$

Figura 6.11: Regla de sintonía de Ziegler-Nichols. [19]

- $K_{cr} = 500$
- $P_{cr} = 8.645$ s.
- $K_p = 0.6 K_{cr} = 300$
- $T_i = 0.5 P_{cr} = 4.32$

- $T_d = 0.125 P_{cr} = 1.08$

Así, el controlador PID queda formado como:

$$G_c(s) = 300\left(1 + \frac{1}{4.32s} + 1.08s^2\right) \quad (6.9)$$

$$G_c(s) = 324s^2 + 69.44s + 300 \quad (6.10)$$

Cabe hacer la aclaración que estos valores nos dieron el puntapié para realizar una sintonía fina del controlador para este sistema. Por otro lado, es útil decir que la ecuación con la que se trabajó internamente en el microcontrolador dispone los valores del controlador negativamente, debido a las ecuaciones encontradas de manera teórica. Esto es así, ya que la pantalla no permite que los valores negativos sean enviados.

```

1201      error = ( Sp - Pv );
1202      delay(1);
1203      Iterm += ((-ki) * error);
1204      if (Iterm > 1000)
1205      {
1206          Iterm = 1000;
1207      }
1208      else if (Iterm < 0)
1209      {
1210          Iterm = 0;
1211      }
1212      dInput1 = (Pv - Pv_1) ;
1213      Cv = ((-kp) * error) + Iterm - ((-kd) * dInput1);
1214      delay(1);
1215
1216      if (Cv > 1000)
1217      {
1218          Cv = 1000;
1219      }
1220      else if (Cv < 0)
1221      {
1222          Cv = 0;
1223      }
1224

```

Figura 6.12: Fragmento del código implementando la ecuación del controlador. [32]

Como primeras conclusiones para este tipo de control de primer orden (1 tanque), podemos resaltar que los análisis teóricos realizados, tanto desde el análisis a partir de las ecuaciones de la teoría de Bernoulli como las dispuestas por el libro **Ingeniería de Control Moderna 5ta Edición** Katsuhiko Ogata, nos dieron las bases y conceptos para aplicar de manera practica este sistema. Si bien es útil aclarar que los valores obtenidos de manera teórica difieren de los obtenidos de forma practica, aun así fue necesario todo el desarrollo para entender conceptos de la teoría de control, tanto el análisis y diseño como la puesta en marcha a partir de las ecuaciones.

CONTROLADOR CON $K_p = 250$, $K_i = 5000 \times 0.01 = 50$, $K_d = 10$.

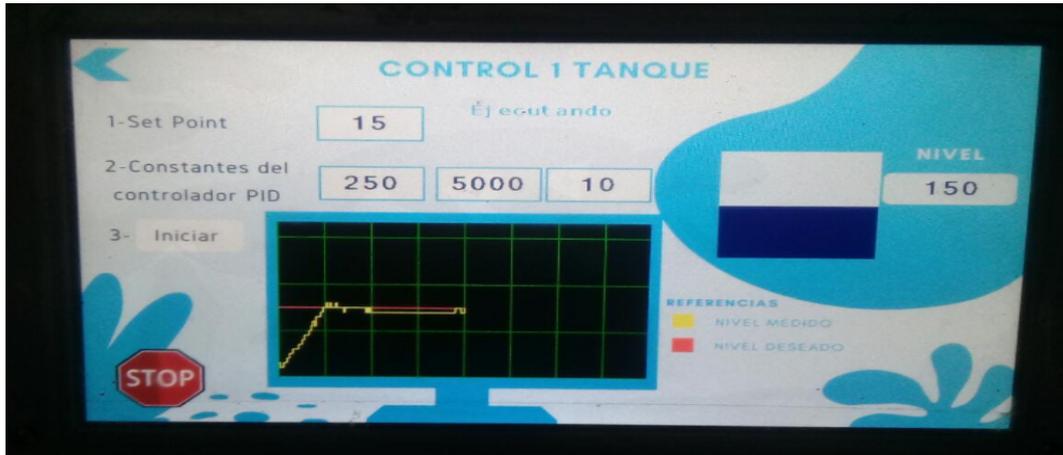


Figura 6.13: Respuesta real del sistema con el controlador PID. [32]

Como dato importante, el tiempo de cada celda en la figura es de 1:01 minuto, dando un total de 8:15 minutos que dura el sistema actuando el control.

6.2.2. Sistema de segundo orden

Para realizar el sistema de segundo orden, se utilizaron ambos tanques de la unidad, sin embargo el control fue realizado en el tanque N°2 donde se encuentra el sensor de presión.

Pruebas del análisis teórico

- La función de transferencia de los tanques en cascada es:

$$\frac{H_2(s)}{Q_p(s)} = \frac{\frac{K_v}{2\sqrt{(H_1 - H_2)}}}{A^2s^2 + A\left(\frac{K_v}{2\sqrt{(H_1 - H_2)}} + \frac{K_v}{2\sqrt{H_2}}\right)s + \frac{K_v}{2\sqrt{(H_1 - H_2)}} \frac{K_v}{2\sqrt{H_2}}} \quad (6.11)$$

Considerando $H_1 = 15$, $H_2 = 10$ y $K_v = 55.59 \frac{cm^3}{s}$, tenemos:

$$\frac{H_2(s)}{Q_p(s)} = \frac{12.43}{30625s^2 + 6.5 \times 10^5 s + 109.25} \quad (6.12)$$

- Función de transferencia del sensor de presión:

$$G_{sensor} = \frac{V_{out}(s)}{H(s)} = 0.044 + \frac{0.2174}{H(s)} \quad (6.13)$$

- Función de transferencia de electroválvula:

$$\frac{Q_1(s)}{V(s)} = 17.49 + \frac{39.17}{V(s)} \quad (6.14)$$

- Función de transferencia en lazo abierto:

$$G(s) = \frac{0.09566}{30625s^2 + 650000s + 109.3} \quad (6.15)$$

- Función de transferencia en lazo cerrado:

$$G(s) = \frac{0.09566}{30625s^2 + 650000s + 110.2} \quad (6.16)$$

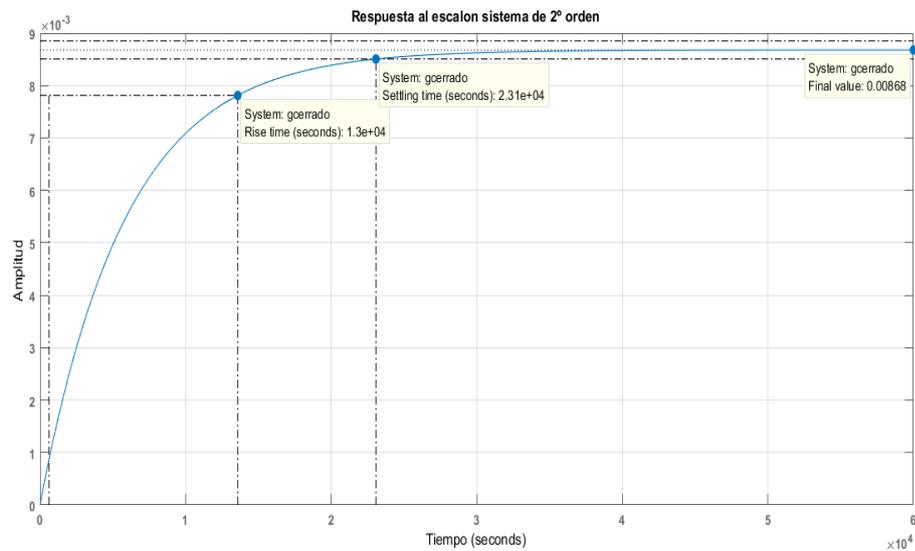


Figura 6.14: Respuesta ante una entrada escalón al sistema de 2º orden. [32]

- Función de transferencia del controlador PI :

$$G_{PI}(s) = K_p + \frac{k_i}{s} \quad (6.17)$$

$$G_{PI}(s) = \frac{205s + 0.06}{s} \quad (6.18)$$

De la ec. (6.18) podemos observar que el controlador que compensa este sistema es del tipo PI con las constantes iguales a:

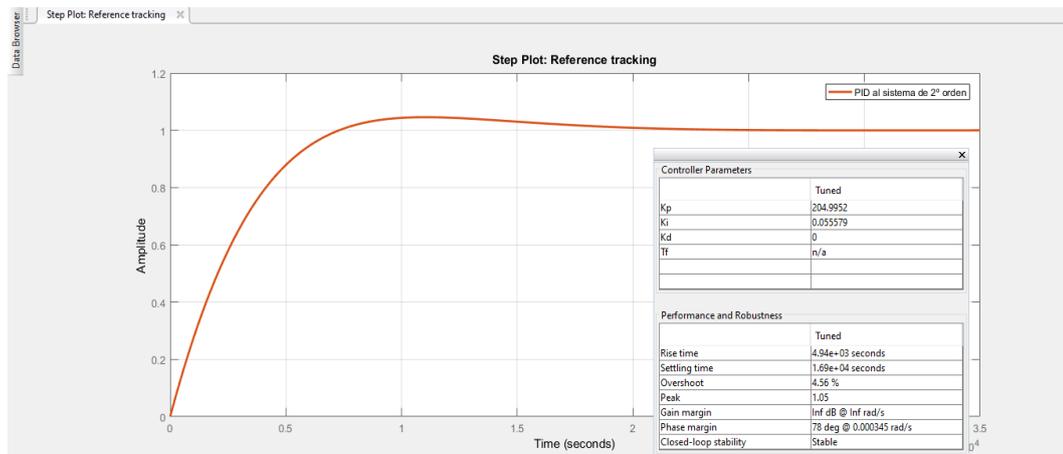


Figura 6.15: Respuesta ante una entrada escalón al sistema de 2º orden compensado con un PI. [32]

- $K_p = 205$
- $K_i = 0.06$

Al aplicar estos valores a través de la pantalla obtenemos lo siguiente:

- Tiempo de establecimiento : No se estabilizo, sino que producía oscilaciones entre 9.3 cm y 8.3 cm.
- sobreimpulso : 0 %

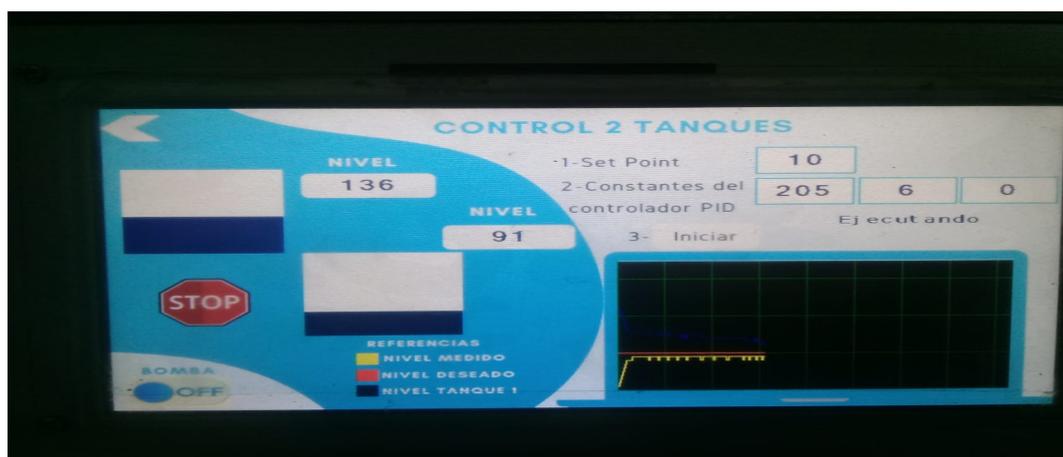


Figura 6.16: Respuesta real del sistema con el controlador PI teórico. [32]

Desde Ziegler-Nichols

Como en el caso anterior, control de un tanque, se encontraron las constantes del controlador a través de la segunda sintonía de Ziegler- Nichols, llevando al sistema a un estado de oscilación.

Este método fue realizado a partir del entorno modo automático control de 2 tanques con las siguientes condiciones:

- Válvula de entrada abierta 20 %.
- Válvula de desagote ubicada debajo del tanque N°1 totalmente cerrada.
- Válvula de desagote ubicada debajo del tanque N°2 totalmente cerrada.
- Válvula de salida abierta un 15 %.
- Potenciómetro del tablero didáctico en la posición '0'.
- El tanque N°1 fue cargado solo una vez al ingresar al modo de control de 2 tanques.

Experimentalmente se utilizó el valor de $K_p = 400$ y se obtuvo una oscilación sostenida entre 10.6 y 9.8 cm. A partir de esta oscilación se tomó el periodo crítico, es decir el tiempo que dura esta oscilación entre un valor dado. Para este caso se tomó el tiempo que tardaba en volver al valor de 9.8 cm .

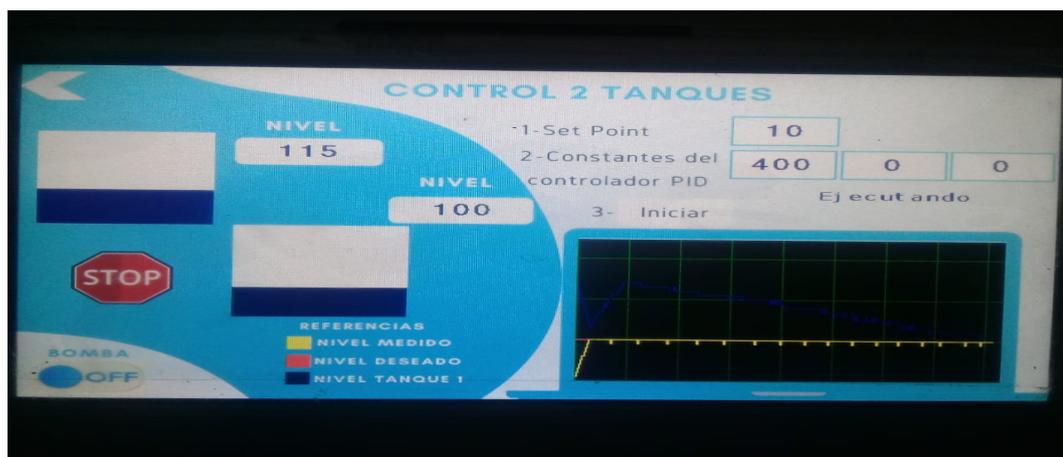


Figura 6.17: Respuesta oscilatoria real del sistema. [32]

Se tomaron 10 muestras del periodo crítico debido a que, el caudal que ingresa al tanque N°2 no es constante, esto se debe al cambio de presión del tanque N°1,

consecuente de bajar su nivel debido al vaciamiento hacia al tanque N°2. Las muestras fueron tomadas desde el nivel de 25 cm al nivel de 12.4 cm del tanque N°1.

N° Muestra	Nivel [cm]	t[s]
1	25	36.16
2	23.6	33.44
3	22.2	37.89
4	20.8	32.79
5	19.4	30.27
6	18	29.38
7	16.6	29.78
8	15.2	30.62
9	13.8	27.46
10	12.4	30.12

Tabla 6.5: Tiempo de muestreo del periodo critico.

El promedio del periodo critico fue igual a 31.79 segundos. A partir del cual se obtuvieron los siguientes valores del controlador.

- $K_{cr} = 400$
- $P_{cr} = 31.79$ s.
- $K_p = 0.6 K_{cr} = 240$
- $T_i = 0.5 P_{cr} = 15.89$
- $T_d = 0.125 P_{cr} = 3.97$

Así, el controlador PID queda formado como:

$$G_c(s) = 240\left(1 + \frac{1}{15.89s} + 3.97s^2\right) \quad (6.19)$$

$$G_c(s) = 952.8s^2 + 15.10s + 240 \quad (6.20)$$



Figura 6.18: Respuesta real con parámetros de Ziegler-Nichols. [32]

Comentarios acerca de este control

Con los valores teóricos del controlador encontrados a través del PID Tuner, podemos observar que el sistema oscila entre 99 y 100 mm, esto lo realiza hasta que el sistema no puede seguir realizando el control, esto es debido a que el tanque N°1 no posee el nivel adecuado para hacerlo. Para solucionar ese problema, y hacerlo al sistema más didáctico, se agregó un botón de dos estados para encender o desactivar la bomba, y así cargar el tanque N°1 para que el control se prolongue.

El tiempo que la unidad, sin volver a cargar el tanque, efectúa el control es de 9:36.16 minutos.

El controlador con los valores obtenidos de manera experimental, presentan una mejora en la respuesta ante el caso anterior. Del análisis realizado a este control, el cual podemos observar en la figura 6.18, se puede concluir que el sistema permanece en estado estable durante 2:45 minutos, donde se produce una perturbación, esta se repite periódicamente 2 veces más, hasta que el controlador no puede efectuarse debido al bajo nivel del tanque N°1.

CONTROLADOR CON $K_p = 240$, $K_i = 15 \times 0.01 = 0.15$, $K_d = 936$.

Capítulo 7

Conclusiones y Trabajos futuros

7.1. Conclusiones

Se logró diseñar y elaborar una unidad didáctica para sistemas de control con tanques en cascada usando sensores de diferentes tipos, actuadores y materiales con referencias existentes en el mercado que facilitaron el diseño mecánico de la unidad, donde se pudo explorar diversas formas para lograr esta unidad y crear una experiencia lo más cercana posible al sector industrial.

La unidad ha sido probada extensamente, tanto en su funcionamiento libre, manual y control automático, con resultados satisfactorios. Es de muy fácil uso y mantenimiento, objetivo que también estaba contemplado para su uso en el laboratorio.

Las pruebas realizadas demostraron que el sistema realizado, cumple con las especificaciones planteadas. Por lo que se afirma, que se alcanzaron todos los objetivos planteados y que además, se demostraron todos los conocimientos técnicos en el área de la electrónica adquiridos en la formación como ingeniero.

El sistema de control implementado en este prototipo, puede ser trasladado a un sistema real a gran escala, el principio de control se mantiene y lo único que debería cambiarse es el rango de trabajo del sensor y la potencia del driver que maneja la bomba.

7.2. Trabajos futuros

Como trabajos futuros de la unidad, sería importante establecer una comunicación online desde un modulo de red. Esto permitirá manipular y visualizar la unidad desde

cualquier lugar con conexión a internet, como lo tienen las grandes empresas que monitorean sus actuadores desde un sistema SCADA conectado a la red de internet.

Para la realización de prácticas en sistemas de control, sería muy provechoso realizar un control multivariable, es decir, controlar el nivel de los dos tanques simultáneamente a través de los dos sensores. Por otro lado, realizar el control mediante el control del caudal entregado por la bomba periférica sería incrementar estos conceptos de control.

El desarrollo de la aplicación para dispositivos móvil, fue limitada a controles tipo ON-OFF . Disponer opciones para el manejo de la posición de la válvula y el tipo de controlador a utilizar junto con sus constantes, optimizaría en gran medida esta utilidad de la unidad.

Apéndice A

Datasheet de Componentes

A continuación, se presenta un resumen de las hojas de datos de los dispositivos utilizados, a fin de tener disponible parte de la información técnica más relevante a la hora de leer este documento.

- Arduino Mega 2560.
- MOC 3041.
- L293D.
- TIP3055.
- Sensor MPX5010.
- Motor Stepper.

A.1. Atmega 2560

Technical Specification

EAGLE files: [arduino-mega2560-reference-design.zip](#) Schematic: [arduino-mega2560-schematic.pdf](#)

Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

the board

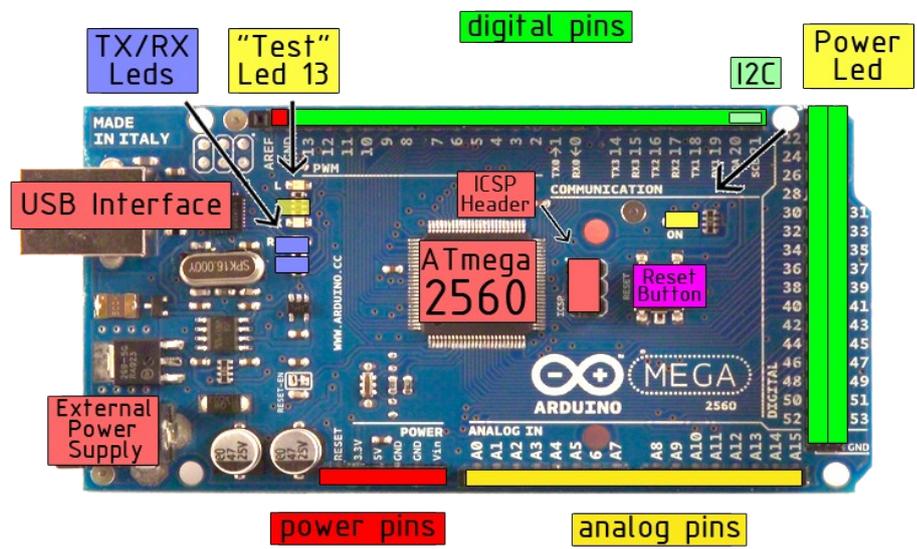


Figura A.1: Arduino Mega. [36]

A.3. MOC 3041



FAIRCHILD
SEMICONDUCTOR®

June 2005

MOC3031M, MOC3032M, MOC3033M, MOC3041M, MOC3042M, MOC3043M

6-Pin DIP Zero-Cross Optoisolators Triac Driver Output (250/400 Volt Peak)

Features

- Simplifies logic control of 115 VAC power
- Zero voltage crossing
- dv/dt of 2000 V/ μ s typical, 1000 V/ μ s guaranteed
- VDE recognized (File # 94766), ordering option V (e.g., MOC3043VM)

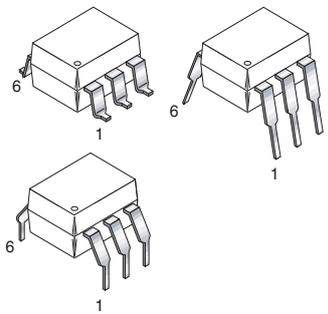
Applications

<ul style="list-style-type: none"> ■ Solenoid/valve controls ■ Static power switches ■ Temperature controls ■ AC motor starters 	<ul style="list-style-type: none"> ■ Lighting controls ■ AC motor drives ■ E.M. contactors ■ Solid state relays
---	---

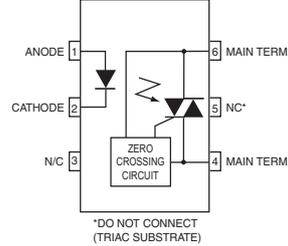
Description

The MOC303XM and MOC304XM devices consist of a AlGaAs infrared emitting diode optically coupled to a monolithic silicon detector performing the function of a zero voltage crossing bilateral triac driver.

They are designed for use with a triac in the interface of logic systems to equipment powered from 115 VAC lines, such as teletypewriters, CRTs, solid-state relays, industrial controls, printers, motors, solenoids and consumer appliances, etc.



Schematic



*DO NOT CONNECT (TRIAC SUBSTRATE)

MOC3031M, MOC3032M, MOC3033M, MOC3041M, MOC3042M, MOC3043M 6-Pin DIP Zero-Cross Optoisolators Triac Driver Output (250/400 Volt Peak)

Figura A.3: MOC 3041. [23]

A.4. L293D



L293, L293D

SLRS008D – SEPTEMBER 1986 – REVISED JANUARY 2016

L293x Quadruple Half-H Drivers

1 Features

- Wide Supply-Voltage Range: 4.5 V to 36 V
- Separate Input-Logic Supply
- Internal ESD Protection
- High-Noise-Immunity Inputs
- Output Current 1 A Per Channel (600 mA for L293D)
- Peak Output Current 2 A Per Channel (1.2 A for L293D)
- Output Clamp Diodes for Inductive Transient Suppression (L293D)

2 Applications

- Stepper Motor Drivers
- DC Motor Drivers
- Latching Relay Drivers

3 Description

The L293 and L293D devices are quadruple high-current half-H drivers. The L293 is designed to provide bidirectional drive currents of up to 1 A at voltages from 4.5 V to 36 V. The L293D is designed to provide bidirectional drive currents of up to 600-mA at voltages from 4.5 V to 36 V. Both devices are designed to drive inductive loads such as relays, solenoids, DC and bipolar stepping motors, as well as other high-current/high-voltage loads in positive-supply applications.

Each output is a complete totem-pole drive circuit, with a Darlington transistor sink and a pseudo-Darlington source. Drivers are enabled in pairs, with drivers 1 and 2 enabled by 1,2EN and drivers 3 and 4 enabled by 3,4EN.

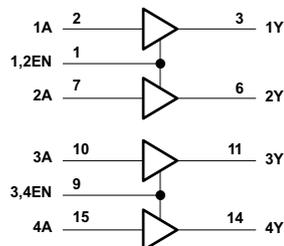
The L293 and L293D are characterized for operation from 0°C to 70°C.

Device Information⁽¹⁾

PART NUMBER	PACKAGE	BODY SIZE (NOM)
L293NE	PDIP (16)	19.80 mm x 6.35 mm
L293DNE	PDIP (16)	19.80 mm x 6.35 mm

(1) For all available packages, see the orderable addendum at the end of the data sheet.

Logic Diagram



An IMPORTANT NOTICE at the end of this data sheet addresses availability, warranty, changes, use in safety-critical applications, intellectual property matters and other important disclaimers. PRODUCTION DATA.

Figura A.4: Doble puente H. [38]

A.5. TIP 3055



TIP2955
TIP3055

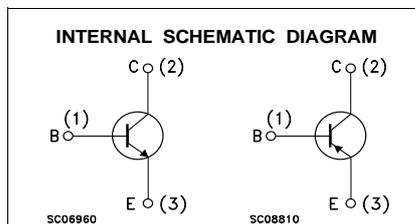
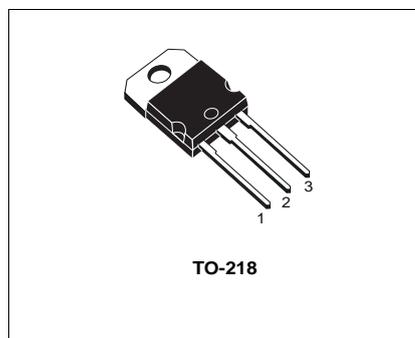
COMPLEMENTARY SILICON POWER TRANSISTORS

- STMicroelectronics PREFERRED SALESTYPES
- COMPLEMENTARY PNP - NPN DEVICES

DESCRIPTION

The TIP3055 is a silicon Epitaxial-Base Planar NPN transistor mounted in TO-218 plastic package. It is intended for power switching circuits, series and shunt regulators, output stages and hi-fi amplifiers.

The complementary PNP type is the TIP2955.



ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value		Unit
		PNP	TIP2955	
V_{CB0}	Collector-Base Voltage ($I_E = 0$)		100	V
V_{CEO}	Collector-Emitter Voltage ($I_B = 0$)		60	V
I_C	Collector Current		15	A
I_B	Base Current		7	A
P_{tot}	Total Dissipation at $T_c \leq 25^\circ\text{C}$		90	W
T_{stg}	Storage Temperature		-65 to 150	$^\circ\text{C}$
T_j	Max. Operating Junction Temperature		150	$^\circ\text{C}$

For PNP types voltage and current are negative.

August 1999

1/4

Figura A.5: Transistor de Potencia. [40]

A.6. Sensor MPX5010DP

MOTOROLA
SEMICONDUCTOR TECHNICAL DATA

Order this document
by MPX5010/D



Integrated Silicon Pressure Sensor On-Chip Signal Conditioned, Temperature Compensated and Calibrated

The MPX5010 series piezoresistive transducer is a state-of-the-art monolithic silicon pressure sensor designed for a wide range of applications, but particularly those employing a microcontroller or microprocessor with A/D inputs. This patented, single element transducer combines advanced micromachining techniques, thin-film metallization, and bipolar processing to provide an accurate, high level analog output signal that is proportional to the applied pressure.

Features

- 5.0% Maximum Error over 0° to 85°C
- Ideally Suited for Microprocessor or Microcontroller-Based Systems
- Patented Silicon Shear Stress Strain Gauge
- Durable Epoxy Unibody Element
- Temperature Compensated over -40° to +125°C

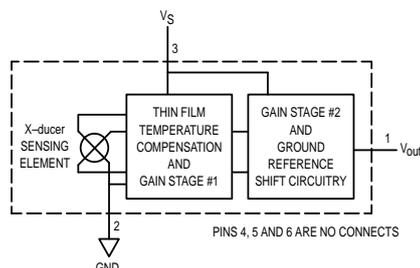


Figure 1. Fully Integrated Pressure Sensor Schematic

MAXIMUM RATINGS(1)

Parameters	Symbol	Value	Unit
Overpressure ⁽²⁾ (P1 > P2)	P _{max}	75	kPa
Burst Pressure ⁽²⁾ (P1 > P2)	P _{burst}	100	kPa
Storage Temperature	T _{stg}	-40 to +125	°C
Operating Temperature	T _A	-40 to +125	°C

1. T_C = 25°C unless otherwise noted.
2. Exposure beyond the specified limits may cause permanent damage or degradation to the device.

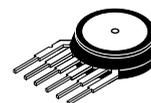
Senseon and X-ducer are trademarks of Motorola, Inc.

REV 2

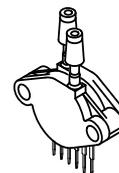
© Motorola, Inc. 1997

MPX5010 SERIES

INTEGRATED
PRESSURE SENSOR
0 to 10 kPa (0 to 1.45 psi)
0.2 to 4.7 V OUTPUT



BASIC CHIP
CARRIER ELEMENT
CASE 867-08, STYLE 1



DIFFERENTIAL
PORT OPTION
CASE 867C-05, STYLE 1

PIN NUMBER

Pin	Function	Pin	Function
1	V _{out}	4	N/C
2	Gnd	5	N/C
3	V _S	6	N/C

NOTE: Pins 4, 5, and 6 are internal device connections. Do not connect to external circuitry or ground. Pin 1 is noted by the notch in the Lead.

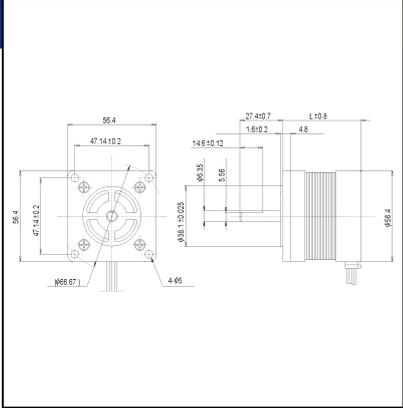
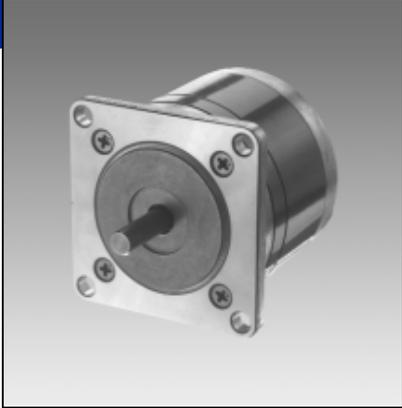


Figura A.6: Sensor de Presión. [34]

A.7. Motor Stepper

SIZE
23
SERIES

STH56D

SPECIFICATION

MODEL	STEP ANGLE	VOLTAGE	CURRENT	RESISTANCE	INDUCTANCE	HOLDING TORQUE	ROTOR INERTIA	NUMBER OF LEADS	WEIGHT	DIMENSION
SINGLE SHAFT	DEG.	V	A/Phase	Ω /Phase	mH/Phase	kg-cm	g-cm ²	LEAD	kg	L
STH56D1904	1.8	3.6	1.2	3.0	3.0	2.0	70	6	0.36	38.0
STH56D1953	1.8	12.0	0.3	40.0	44.0	2.0	70	6	0.36	38.0
STH56D2903	1.8	5.1	1.0	5.1	9.0	3.6	110	6	0.48	49.5
STH56D2909	1.8	12.0	0.4	30.0	40.0	3.3	110	6	0.48	49.5
STH56D2914	1.8	20.0	0.25	80.0	96.0	3.2	110	6	0.48	49.5
STH56D2953	1.8	4.0	1.3	3.1	5.2	3.3	110	6	0.48	49.5
STH56D3951	1.8	6.0	1.2	5.0	10.0	4.5	160	6	0.55	55.5
STH56D3904	1.8	12.0	0.6	20.0	40.0	5.0	160	6	0.55	55.5
STH56D3953	1.8	24.0	0.3	80.0	100.0	4.5	160	6	0.55	55.5

TYPICAL PERFORMANCE

— PULL OUT
- - - - PULL IN

STH56D1953 DRIVER: TYPE A (C = 1 μ F, R = 0) VOLTAGE: $V_v = 12$ V
EXCITING MODE = 2 Phase INERTIAL LOAD: 24 g-cm²

STH56D2905 DRIVER: TYPE A (C = 1 μ F, R = 0) VOLTAGE: $V_v = 24$ V
EXCITING MODE = 2 Phase INERTIAL LOAD: 24 g-cm²

STH56D2909 DRIVER: TYPE A (C = 1 μ F, R = 0) VOLTAGE: $V_v = 12$ V
EXCITING MODE = 2 Phase INERTIAL LOAD: 100 g-cm²

STH56D3905 DRIVER: TYPE A (C = 1 μ F, R = 0) VOLTAGE: $V_v = 12$ V
EXCITING MODE = 2 Phase INERTIAL LOAD: 100 g-cm²

STH56D3953 DRIVER: TYPE A (C = 1 μ F, R = 0) VOLTAGE: $V_v = 24$ V
EXCITING MODE = 2 Phase INERTIAL LOAD: 100 g-cm²

STH56D4920 DRIVER: TYPE D
CURRENT = 1 A/Phase VOLTAGE: $V_v = 24$ V
INERTIAL LOAD: 770 g-cm²

STEPPING MOTORS



21

Figura A.7: Motor paso a paso. [37]

Apéndice B

Diseño de PCB

A continuación, se presenta un resumen de las diferentes capas de diseño del PCB fabricado de la capa principal.

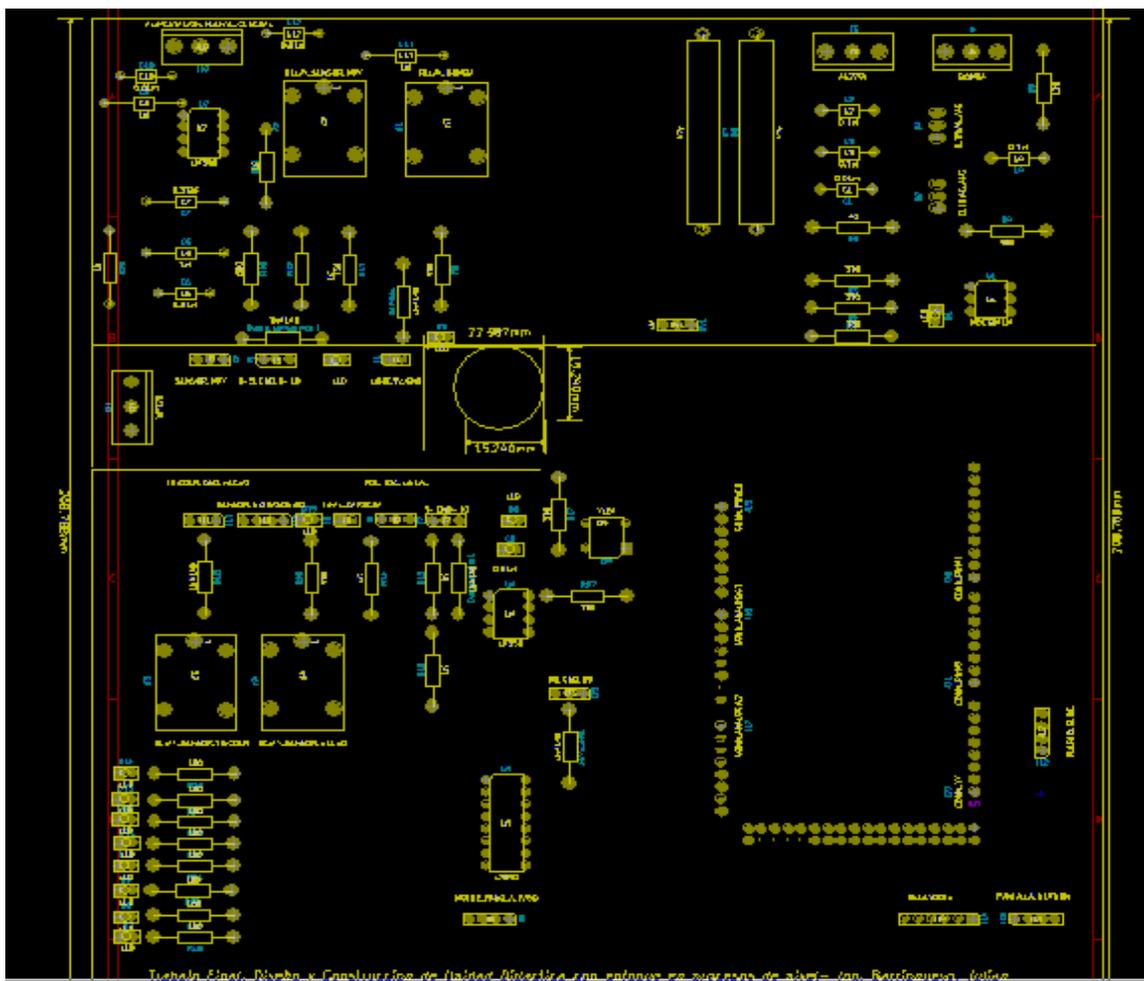
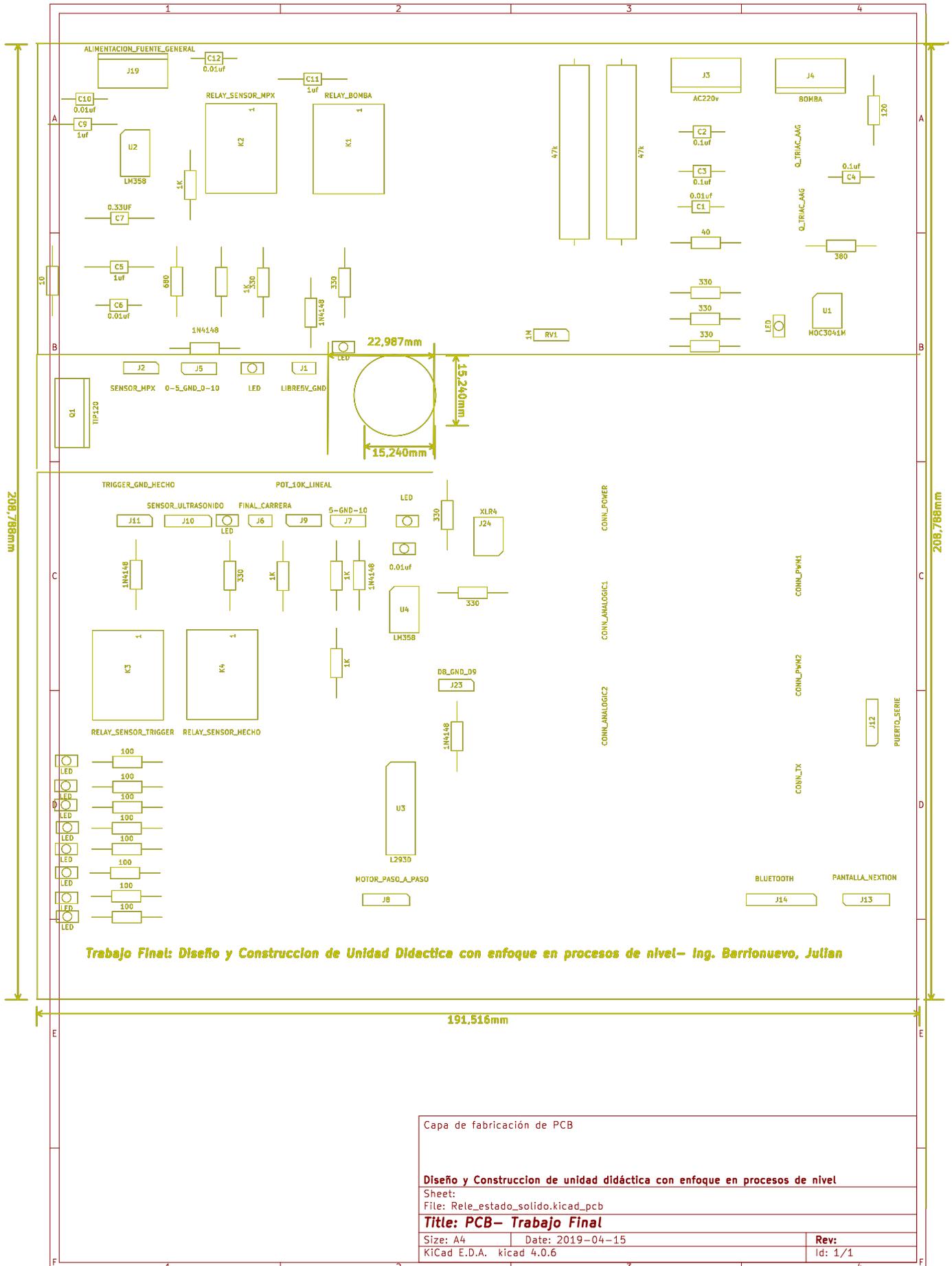


Figura B.1: Capa de fabricación del PCB. [32]



Trabajo Final: Diseño y Construcción de Unidad Didáctica con enfoque en procesos de nivel- Ing. Barrionuevo, Julian

Capa de fabricación de PCB

Diseño y Construcción de unidad didáctica con enfoque en procesos de nivel

Sheet:

File: Rele_estado_solido.kicad_pcb

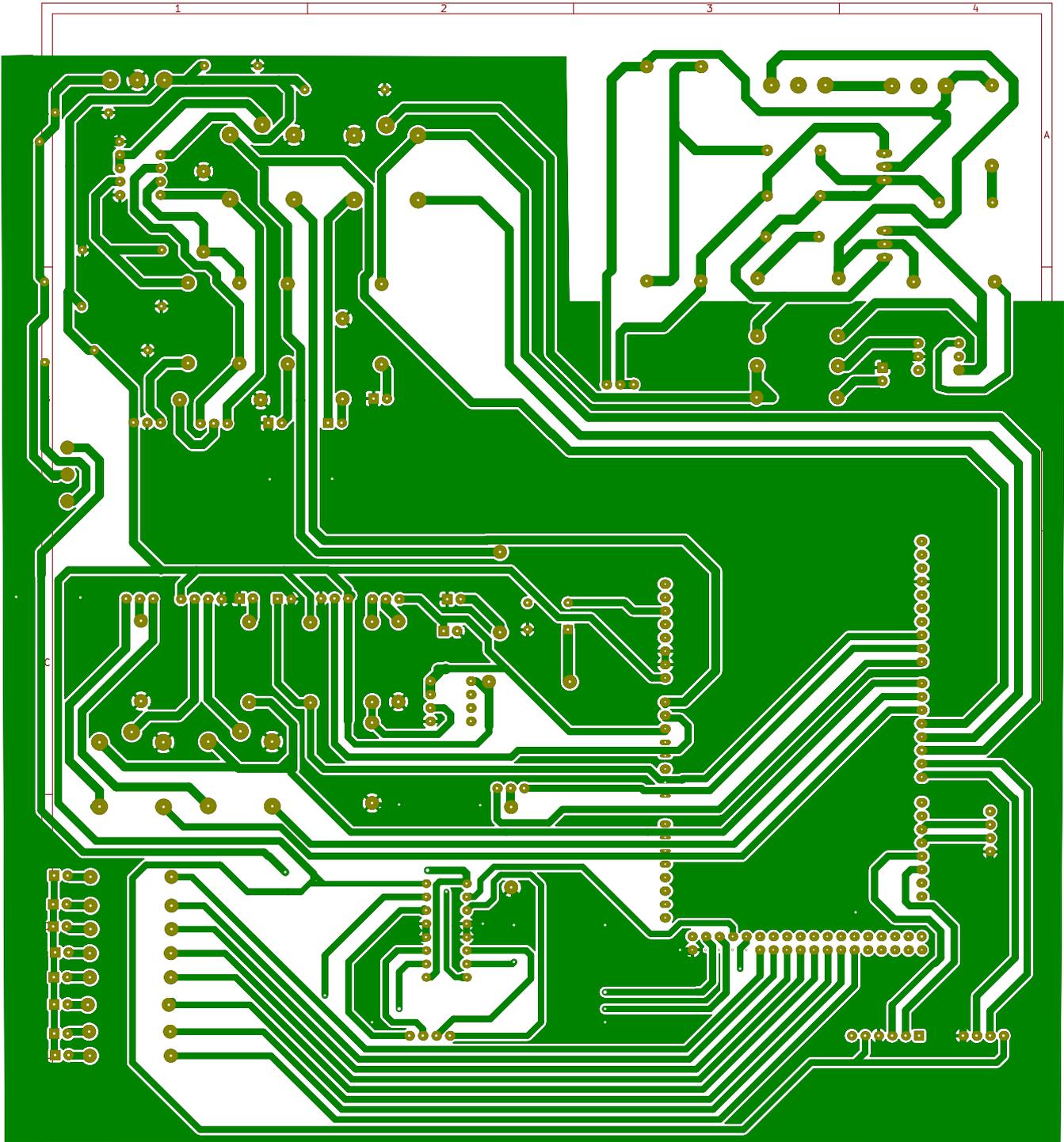
Title: PCB- Trabajo Final

Size: A4 Date: 2019-04-15

KiCad E.D.A. kicad 4.0.6

Rev:

Id: 1/1



Capa B.Cu de PCB

Diseño y Construcción de unidad didáctica con enfoque en procesos de nivel

Sheet:

File: Rele_estado_solido.kicad_pcb

Title: PCB- Trabajo Final

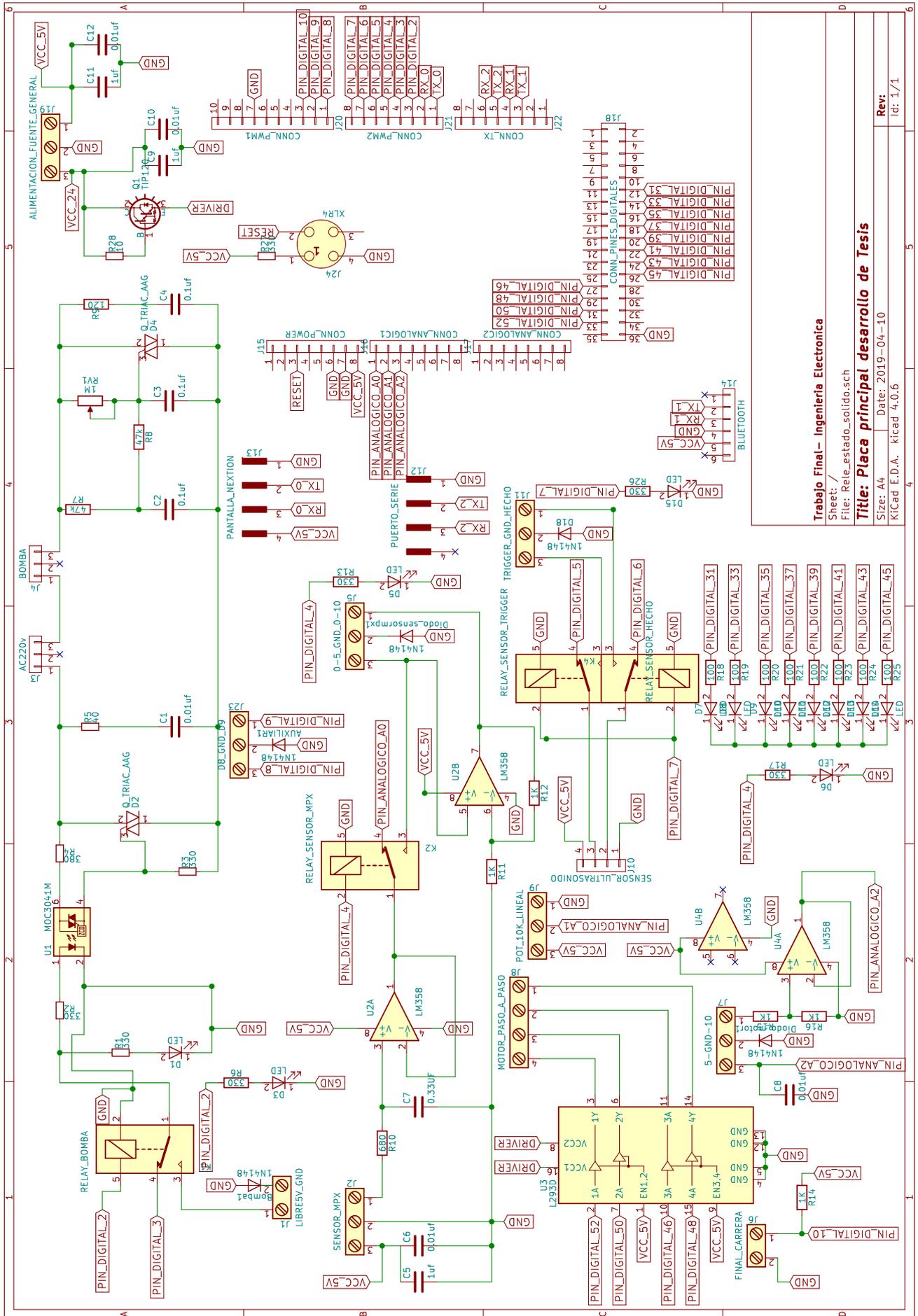
Size: A4

Date: 2019-04-15

Rev:

KiCad E.D.A. kicad 4.0.6

Id: 1/1



Trabajo Final— Ingeniería Electronica

Sheet: /

File: Rele_estado_solido.sch

Title: Placa principal desarrollo de Tesis

Size: A4

Date: 2019-04-10

KiCad E.D.A. kicad 4.0.6

Rev:

1/1

B.1. Especificaciones PCB

En este apartado se detallan las especificaciones más relevantes al diseño del PCB de la placa principal.

- Dimensiones: 208.788 mm x 191.516 mm x 1 mm.
- Material: Placa ETP (Electrolítico Puro), cobre al 99 %.
- Circuito impreso: Simple faz.
- Diámetro de pista: 1.10 mm (0.6 mm más pequeño).
- Conexiones: 183.
- Nodos: 329.
- Redes: 147.
- Impresión de componentes: si, color negro.
- Terminación: Estaño plomo selectivo.
- Máscara antisoldante: si, color barniz.

B.2. Lista de Componentes Utilizados

La tabla B.1 muestra la lista de los componentes utilizados en la placa principal de la unidad, la mayoría de encapsulado tipo THT¹. En la tabla B.2 se detallan los componentes utilizados en el diseño de la fuente de alimentación para la electroválvula. Así mismo, en la tabla B.3 se indican los componentes utilizados en el tablero de modo libre.

¹La tecnología de agujeros pasantes, más conocida por las siglas THT del inglés Through-Hole Technology, es un tipo de tecnología que utiliza los agujeros que se practican en las placas de los circuitos impresos para el montaje de los diferentes elementos electrónicos.

Nombre	Cantidad	Descripción	Encapsulado
Diodo	5	1N4148	THT
Capacitor	5	0.01uf	THT
Capacitor	3	0.1uf	THT
Capacitor	3	1uf	THT
Triac	2	BT137	T0-220
Relay	4	Rele 5v	THT
Led	10	Led de alta luminosidad	THT
Resistor	8	330 - 1/4 w	THT
Resistor	1	380 - 1/4 w	THT
Resistor	1	40 - 1/4 w	THT
Resistor	2	47k - 5 w	THT
Resistor	1	120 - 1/4 w	THT
Resistor	1	680 - 1/4 w	THT
Resistor	5	1k - 1/4 w	THT
Resistor	1	10 - 1/4 w	THT
Resistor	1	120 - 1/4 w	THT
Resistor	8	100 - 1/4 w	THT
Resistor	1	1M - variable	THT
Optoacoplador	1	MOC3041 C	DIP-6
Amplificador operacional	2	LM358	DIP-8
Cuádruple Driver H	1	L293D	16-Pin PDIP
Transistor	1	TIP3055	T0-218
Conector	21	Pin-Headers (Diferentes medidas)	THT
Conector terminal	4	Terminal-Block x3	THT

Tabla B.1: Lista de Componentes Placa Principal.

Nombre	Cantidad	Descripción	Encapsulado
Transformador	1	220 AC- 24 AC 2A	-
Fusible	1	3 A	THT
Conector Terminal	2	Terminal-Block x3	THT
Capacitor	1	4700uf	THT
Capacitor	1	10uf	THT
Capacitor	1	0.1uf	THT
Puente de diodo	1	KBL 10	THT
Diodo	2	1N4148	THT
Led	1	Led de alta luminosidad	THT
Resistor	1	240 - 1/4 w	THT
Resistor	1	10 - 1/4 w	THT
Resistor	1	330 - 1/4 w	THT
Resistor	1	5K- variable	THT
Regulador de Tensión Lineal	1	LM317	TO-220
Regulador de Tensión Positiva	1	LM7805	TO-220
Transistor	1	TIP3055	T0-218

Tabla B.2: Lista de Componentes Fuente de alimentación.

Nombre	Cantidad	Descripción
Cable cinta	0.50 mts	18 hilos
Cable UTP	2 mts	Cat. 6 -250 MHz Clase E
Placa de cobre	1	70 mm x 40 mm
Placa Madera MDF	1	200 mm x 200 mm x 4 mm
Placa Madera MDF	1	200 mm x 200 mm x 6 mm
Placa Madera MDF	1	900 mm x 300 mm x 9 mm
Placa de acrílico	1	200 mm x 200 mm
Conector banana	15	Tipo Hembra
Led	4	Alta luminosidad
Motor CC	1	12 v
Switch	2	Final de carrera NA
Resistor	1	1M- tipo B variable
Switch	1	Tipo palanca
Tornillos	10	Para maderas

Tabla B.3: Lista de Componentes Tablero de modo libre.

Apéndice C

Código de Programación

```
1 #include <Nextion.h>
2 #include <AccelStepper.h>
3 #include <MultiStepper.h>
4
5 ///////////////////////////////////////////////////////////////////
6 // Definicion de elementos de la electrovalvula //
7 ///////////////////////////////////////////////////////////////////
8 #define HALFSTEP 8
9 #define motorPin1 52
10 #define motorPin2 50
11 #define motorPin3 48
12 #define motorPin4 46
13
14 //const int FinalCarrera = 10;
15 int EstadoFinalCarrera = 0;
16 int Posicion_valvula = 0 ;
17 int tension_posicion_valvula = 0 ;
18 int potenciometro_valvula = 0 ;
19 int tension_potenciometro_valvula = 0 ;
20
21
22
23 AccelStepper stepper(HALFSTEP, 52, 50, 48, 46);
24
25 /*
26 -----
27 - Declaracion de componentes de la pantalla -
28 - (Pagina, ID del componente, Nombre del componente)-
29 -----
30 */
31
32 NexButton boton_entrada_libre = NexButton (9,3,"b2"); //Entrada Modo LIBRE
33 NexButton boton_entrada_comando = NexButton (9,4,"b3"); //Entrada Modo COMANDOS
34 NexButton boton_entrada_bluetooth = NexButton (9,5,"b4"); //Entrada Modo BLUETOOTH
35 NexButton boton_automatico_control_2 = NexButton (6,4,"b3"); //AUTOMATICO CONTROL 2
36 NexButton boton_inicio_control_1 = NexButton (7,3,"inicio"); //INICIO CONTROL 1
37 NexButton boton_inicio_control_2 = NexButton (8,5,"b2"); //INICIO CONTROL 2
38 NexButton boton_parada_control_1 = NexButton (7,6,"parada"); //PARADA CONTROL 1
39 NexButton boton_parada_control_2 = NexButton (8,2,"b1"); //PARADA CONTROL 2
40 NexPage pageManual = NexPage(9, 0, "Manual"); //Envia a la pagina MODO-MANUAL
41 NexButton salida_libre = NexButton (10,1,"b0salida"); //Salida MODO MANUAL-LIBRE
42 NexButton salida_comandos = NexButton (11,1,"bcomando"); //Salida MODO-COMANDOS
43 NexButton salida_bluetooth = NexButton (12,1,"b0"); //Salida MODO-BLUETOOTH
44
45 NexCheckbox cMPX = NexCheckbox (10,2,"cMPX" ); //Checkbox para activar sensor MPX
46 NexCheckbox cSONIDO = NexCheckbox (10,3,"cSONIDO" ); //Checkbox para sensor SONIDO
```

```

47 NexCheckbox cBOMBA = NexCheckbox (10,4,"cBOMBA" ); //Checkbox para activar BOMBA
48 NexCheckbox cVALVULA = NexCheckbox (10,5,"cVALVULA" ); //Checkbox para VALVULA
49
50 NexRadio rBOMBA = NexRadio (12,8,"rBOMBA"); //Checkbox para BOMBA-BLUETOOTH
51 NexRadio rVALVULA = NexRadio (12,9,"rVALVULA"); //Checkbox para VALVULA-BLUETOOTH
52
53 NexProgressBar j0comando = NexProgressBar (11,2,"j0comando");
    //Barra de progreso del tanque 1 MODO MANUAL-COMANDOS.
54 NexProgressBar j1comando = NexProgressBar (11,3,"j1comando");
    //Barra de progreso del tanque 2 MODO MANUAL-COMANDOS.
55 NexProgressBar j0Bluetooth = NexProgressBar (12,3,"j0bluetooth");
    //Barra de progreso del tanque 1 MODO MANUAL-BLUETOOTH.
56 NexProgressBar j1Bluetooth = NexProgressBar (12,4,"j1bluetooth");
    //Barra de progreso del tanque 2 MODO MANUAL-BLUETOOTH.
57 NexProgressBar jc = NexProgressBar (7,5,"jc");
    //Barra de progreso del tanque 1 MODO CONTROL 1 TANQUE
58
59 NexDSButton bt0 = NexDSButton(11,4,"bt0");
    //Boton para encender la BOMBA modo MANUAL-COMANDOS.
60 NexDSButton bt1 = NexDSButton(11,5,"bt1");
    //Boton para encender la VALVULA modo MANUAL-COMANDOS.
61 NexDSButton btbombacontrol = NexDSButton(8,17,"btbombacontrol");
    //Boton para encender la BOMBA modo CONTROL 2 TANQUES
62
63 NexText texto_bluetooth = NexText (12,7,"t0"); //Texto de "Conectado"
64 NexText t0 = NexText (8,6,"t0"); //Texto de leyendo "Tanque "
65 NexText t1 = NexText (8,15,"t1"); //Texto de leyendo "Cargando"
66
67 NexNumber n0comando = NexNumber (11,6,"n0comando");
    //Numero para el valor del nivel del tanque 1 modo MANUAL-COMANDOS.
68 NexNumber n1comando = NexNumber (11,7,"n1comando");
    //Numero para el valor del nivel del tanque 2 modo MANUAL-COMANDOS.
69 NexNumber n0Bluetooth = NexNumber (12,5,"n0bluetooth");
    //Numero para el valor del nivel del tanque 1 modo MANUAL-BLUETOOTH.
70 NexNumber n1Bluetooth = NexNumber (12,6,"n1bluetooth");
    //Numero para el valor del nivel del tanque 2 modo MANUAL-BLUETOOTH.
71 NexNumber nc = NexNumber (7,4,"nc");
    //Numero para el valor del nivel del tanque 1 MODO CONTROL 1 TANQUE.
72
73 NexNumber nsetpoint = NexNumber (7,2,"nsetpoint"); //Numero para SET POINT
74 NexNumber nP = NexNumber (7,8,"nP"); //Numero para el valor del P Tanque 1
75 NexNumber nI = NexNumber (7,9,"nI"); //Numero para el valor del I Tanque 1
76 NexNumber nD = NexNumber (7,12,"nD"); //Numero para el valor del D Tanque 1
77 NexWaveform onda_control_1 = NexWaveform (7,10,"s0");
78
79 NexNumber n1_control_2 = NexNumber (8,7,"n4");
    //Numero para el valor del nivel del tanque 1 MODO CONTROL 2 TANQUE.
80 NexNumber n2_control_2 = NexNumber (8,8,"n5");
    //Numero para el valor del nivel del tanque 2 MODO CONTROL 2 TANQUE.
81 NexProgressBar barra_tanque_1 = NexProgressBar (8,3,"j0");
    //Barra de progreso del tanque 1 MODO CONTROL 2 TANQUE
82 NexProgressBar barra_tanque_2 = NexProgressBar (8,4,"j1");
    //Barra de progreso del tanque 2 MODO CONTROL 2 TANQUE
83
84 NexNumber nsetpoint_control_2 = NexNumber (8,10,"n6");
    //Numero para el valor del SET POINT CONTROL 2 TANQUE
85 NexNumber nP2 = NexNumber (8,11,"nP2"); //Numero para el valor del P Tanque 2
86 NexNumber nI2 = NexNumber (8,12,"nI2"); //Numero para el valor del I Tanque 2
87 NexNumber nD2 = NexNumber (8,13,"nD2"); //Numero para el valor del D Tanque 2
88
89 NexWaveform onda_control_2 = NexWaveform (8,14,"s0");
90 /*
91 _____
92 - DECLARACION DE PINES -
93 _____
94 */
95
96 char buffer[100]={0};

```

```

97
98 const int Trigger = 5; //Pin digital 5 para el Trigger del sensor
99 const int Echo = 6; //Pin digital 6 para el Echo del sensor
100 const int Relay_Bomba_manual = 2 ;
    //Pin digital 2 para activar la BOMBA en modo MANUAL-LIBRE.
101 const int Relay_Bomba = 3;
    //Pin digital 3 para activar la BOMBA.
102 const int Relay_sensor_mpx_manual = 4;
    //Pin digital 4 para activar el sensor MPX5010 en modo MANUAL-LIBRE.
103 const int Relay_sensor_ultrasonido = 7;
    //Pin digital 7 para activar el sensor ULTRASONIDO en modo MANUAL-LIBRE.
104 const int Relay_sensor_ultrasonido_hecho = 12 ;
    //Pin digital 12 para activar el sensor ULTRASONIDO en modo MANUAL-LIBRE.
105 const int Valvula = 8;
    //Pin digital 8 para activar el led de la ELECTROVALVULA.
106 int variable_de_salida = 0;
107 int variable_de_salida_comando = 0;
108 int variable_de_salida_bluetooth = 0;
109 int variable_de_salida_libre = 0;
110 int variable_de_parada_control = 0;
111 int j=0;
112 float suma_sonido =0;
113
114 /*****
115  Variables para asegurar el tiempo de muestreo
116 *****/
117 unsigned long pasado = 0;
118 unsigned long actual = 0;
119
120 /*****
121  Variables para Sistema de Control de Primer Orden
122 *****/
123 float Pv = 0; // Variable de proceso medida.
124 uint32_t Sp = 0; // Set Point recibido desde la pantalla en cm .
125 float Sp1 = 0; // Set Point (Porcentaje)
126 float Pv_1 =0;
127
128 int32_t Cv = 0; //Variable Manipulada (MV) o variable de control (Ec.en Diferencias)
129 uint32_t Cv_1 = 0; //Variable Manipulada (M-1) o variable de control
130 uint32_t Cva = 0;
131 float errSum =0;
132 float Iterm = 0;
133 double dInput1 = 0;
134
135 int PWM = 0;
136 int32_t dErr = 0;
137 int32_t error = 0; //Error Actual
138 int32_t error_1 = 0; //Error Anterior
139
140 int error_bomba =0 ;
141
142 uint32_t Kp = 0; //Ganancia proporcional
143 uint32_t Ki = 0; //Ganancia Integral
144 uint32_t Kd = 0; //Ganancia Derivativa
145
146 float kp = 0;
147 float ki = 0;
148 float kd = 0;
149 float T = 1; //Tiempo de muestreo
150
151 /*****
152  Variables para Sistema de Control de Segundo Orden
153 *****/
154 unsigned long pasado2 = 0; //Variable para asegurar el tiempo de muestreo
155 unsigned long actual2 = 0; //Variable para asegurar el tiempo de muestreo
156 float Pv2 = 0; // Variable de proceso medida.
157 uint32_t Sp2 = 0; // Set Point (Porcentaje)
158 float Pv2_1 = 0 ;

```

```

159
160 float Sp2_double = 0; // Set Point (Porcentaje)
161
162 int32_t Cv2 = 0; // Variable Manipulada M(k).
163 double Cv2_1 = 0; // Variable Manipulada M(k-1)
164
165 int valor_maximo_nivel_2 = 20 ;
166 int valor_minimo_nivel_2 = 5 ;
167
168 float errSum2 =0;
169 int32_t dErr2 = 0;
170
171 float Iterm2 = 0;
172 float dInput2 = 0;
173 int error2 = 0; //Error Actual = e(k)
174 int32_t error2_1 = 0; //Error Anterior = e(k-1)
175 int error_bomba2 =0 ;
176 int PWM_2 = 0;
177
178 uint32_t Kp2 = 0; //Ganancia Proporcional
179 uint32_t Ki2 = 0; //Ganancia Integral
180 uint32_t Kd2 = 0; //Ganancia Derivativa
181
182 float ki2 = 0 ;
183 float kd2 =0 ;
184 float kp2 = 0 ;
185
186 float T2 = 1;
187
188 /*****
189 * Declaracion de variables para obtener *
190 * lectura del sensor MPX *
191 *****/
192 */
193
194 float sensorValue_MPX = 0;
195 float y = 0;
196 float yb = 0 ;
197 int barra = 0 ;
198
199 /*****
200 * Declaracion de variables para obtener *
201 * lectura del sensor ULTRASONIDO *
202 *****/
203 */
204
205 float t = 0; //tiempo que demora en llegar el eco
206 float d = 0 ; //distancia en centimetros
207 float e = 0 ;
208
209
210 /*****
211 * Funcion que activa a los componentes *
212 * que deseamos que sean atendidos *
213 *****/
214 */
215 float pote_valvula ;
216
217 NexTouch *nex_listen_list [] =
218 {
219
220 &boton_entrada_libre ,
221 &boton_entrada_comando ,
222 &boton_entrada_bluetooth ,
223 &salida_comandos ,
224 &salida_libre ,
225 &salida_bluetooth ,
226 &boton_automatgico_control_2 ,

```

```

227 &boton_inicio_control_1,
228 &boton_inicio_control_2,
229 &boton_parada_control_1,
230 &boton_parada_control_2,
231     &bt0,
232     &btbombacontrol,
233     &bt1,
234     &cMPX,
235     &cSONIDO,
236     &cBOMBA,
237     &cVALVULA,
238
239     NULL
240 };
241
242 void leer_pote();
243 void leer_pote(){
244     int i=0;
245     float suma =0;
246     for(i=0;i<50;i++)
247     {
248
249         pote_valvula = ((analogRead(A1)*5.0 )/1023.0) ;
250         delay(1);
251         suma = suma + pote_valvula;
252
253     }
254     pote_valvula=suma/50;
255 }
256
257
258 void setup (void)
259 {
260
261     Serial1.begin(9600); //Conexion a Bluetooth
262     Serial2.begin(9600); // Conexion a LabView
263     Serial.begin(9600); //Conexion a Pantalla Nextion
264     nexSerial.begin(115200);
265     nexInit(); //Inicializacion de la
                pantalla Nextion
266
267 /*****
268 * Definicion de Pines *
269 *****/
270     pinMode(Relay_Bomba_manual, OUTPUT); //pin 2 como salida
271     pinMode(Relay_Bomba, OUTPUT); //pin 3 como salida
272     pinMode(Relay_sensor_mpx_manual, OUTPUT); //pin 4 como salida
273     pinMode(Trigger, OUTPUT); //pin 5 como salida
274     pinMode(Echo, INPUT); //pin 6 como entrada
275     pinMode(Relay_sensor_ultrasonido, OUTPUT); //pin 7 como salida
276     pinMode(Relay_sensor_ultrasonido_hecho, OUTPUT); //pin 12 como salida
277     pinMode(10, OUTPUT); //pin 10 como salida
278     pinMode(Valvula, OUTPUT); //pin 8 como salida
279
280     pinMode(31, OUTPUT); //pin como salida Led 1
281     pinMode(33, OUTPUT); //pin como salida Led 2
282     pinMode(35, OUTPUT); //pin como salida Led 3
283     pinMode(37, OUTPUT); //pin como salida Led 4
284     pinMode(39, OUTPUT); //pin como salida Led 5
285     pinMode(41, OUTPUT); //pin como salida Led 6
286     pinMode(43, OUTPUT); //pin como salida Led 7
287     pinMode(45, OUTPUT); //pin como salida Led 8
288     pinMode(13, OUTPUT); //pin como salida 13
289
290     digitalWrite(31, LOW); //Inicializamos el pin con 0
291     digitalWrite(33, LOW); //Inicializamos el pin con 0
292     digitalWrite(35, LOW); //Inicializamos el pin con 0
293     digitalWrite(37, LOW); //Inicializamos el pin con 0

```

```

294 digitalWrite(39,LOW); //Inicializamos el pin con 0
295 digitalWrite(41,LOW); //Inicializamos el pin con 0
296 digitalWrite(43,LOW); //Inicializamos el pin con 0
297 digitalWrite(45,LOW); //Inicializamos el pin con 0
298
299
300 digitalWrite(13,LOW); //Inicializamos el pin con 0
301 digitalWrite(Trigger, LOW); //Inicializamos el pin con 0
302
303
304 /*****
305 Inicializacion de la electrovalvula *
306 *****/
307 stepper.setMaxSpeed(500.0);
308 stepper.setAcceleration(400.0);
309 //pinMode(FinalCarrera,INPUT_PULLUP);
310
311 void leer_pote();
312
313
314 stepper.setCurrentPosition(0); // Encontro la posicion inicial
315
316
317 /*****
318 Llamadas de los botones de la pantalla *
319 *****/
320
321 btbombacontrol.attachPush(btbombacontrolPushCallback, &btbombacontrol);
322 bt0.attachPush(bt0PushCallback, &bt0);
323 bt1.attachPush(bt1PushCallback, &bt1);
324 salida_libre.attachPush(salida_librePushCallback, &salida_libre );
325 salida_comandos.attachPop(salida_comandosPopCallback, &salida_comandos);
326 salida_bluetooth.attachPush(salida_bluetoothPushCallback, &salida_bluetooth );
327 boton_entrada_libre.attachPush(boton_entrada_librePushCallback,&boton_entrada_libre)
328 boton_entrada_comando.attachPop(boton_entrada_comandoPopCallback, &
    boton_entrada_comando);
329 boton_entrada_bluetooth.attachPush(boton_entrada_bluetoothPushCallback, &
    boton_entrada_bluetooth);
330 boton_automatico_control_2.attachPush(boton_automatico_control_2PushCallback, &
    boton_automatico_control_2);
331 boton_inicio_control_1.attachPush(boton_inicio_control_1PushCallback, &
    boton_inicio_control_1);
332 boton_inicio_control_2.attachPush(boton_inicio_control_2PushCallback, &
    boton_inicio_control_2);
333 boton_parada_control_1.attachPush(boton_parada_control_1PushCallback, &
    boton_parada_control_1);
334 boton_parada_control_2.attachPush(boton_parada_control_2PushCallback, &
    boton_parada_control_2);
335 cMPX.attachPush(cMPXPushCallback,&cMPX);
336 cSONIDO.attachPush(cSONIDOPushCallback,&cSONIDO);
337 cBOMBA.attachPush(cBOMBAPushCallback,&cBOMBA);
338 cVALVULA.attachPush(cVALVULAPushCallback,&cVALVULA);
339 }
340
341
342 void loop (void)
343 {
344 {
345 Envio_Tx_App();
346 nexLoop(nex_listen_list); // Espera a que se produzca un evento de la pantalla
347 Tira_Leds();
348
349 }
350 }
351
352 /*****
353 ***** Funciones del programa *****
354 *****/

```

```

355
356 void salida_comandosPopCallback(void *ptr)// Funcion que realiza cuando se apreta LA
      FLECHA DE SALIDA DE COMANDO
357 {
358
359 variable_de_salida = 1 ;
360 variable_de_salida_libre = 1;
361 variable_de_salida_bluetooth = 1;
362 variable_de_salida_comando = 1;
363
364 nexLoop(nex_listen_list) ;
365 delay (1);
366 nexLoop(nex_listen_list) ;
367 digitalWrite(Relay_Bomba, LOW);
368 Valvula_Cerrada();
369 }
370
371 void salida_librePushCallback(void *ptr)// Funcion que realiza cuando se apreta LA
      FLECHA DE SALIDA DE LIBRE
372 {
373 nexLoop(nex_listen_list) ;
374 variable_de_salida_libre = 1;
375 variable_de_salida_bluetooth = 1;
376 variable_de_salida_comando = 1;
377 variable_de_salida = 1 ;
378
379 delay (1);
380 nexLoop(nex_listen_list) ;
381 digitalWrite(Relay_Bomba, LOW);
382 Valvula_Cerrada();
383 digitalWrite(Relay_sensor_mpx_manual, LOW);
384 delay (1);
385 digitalWrite(Relay_sensor_ultrasonido, LOW );
386 digitalWrite(Relay_sensor_ultrasonido_hecho, LOW );
387 digitalWrite(10, LOW );
388 delay (1);
389 digitalWrite(Relay_Bomba_manual, LOW );
390 }
391
392 void salida_bluetoothPushCallback(void *ptr)// Funcion que realiza cuando se apreta
      LA FLECHA DE SALIDA DE BLUETOOTH
393 {
394 digitalWrite(Relay_Bomba, LOW);
395 delay (1);
396 variable_de_salida_libre = 1;
397 variable_de_salida_bluetooth = 1;
398 variable_de_salida_comando = 1;
399
400 nexLoop(nex_listen_list) ;
401 variable_de_salida = 1 ;
402 delay (1);
403 nexLoop(nex_listen_list) ;
404 digitalWrite(Relay_Bomba, LOW);
405 Valvula_Cerrada();
406 }
407
408
409 void boton_parada_control_1PushCallback(void *ptr)// Funcion que realiza cuando se
      apreta EL BOTON DE PARADA TANQUE 1
410 {
411 digitalWrite(Relay_Bomba, LOW);
412 delay (1);
413 Valvula_Cerrada();
414 variable_de_parada_control = 1;
415 variable_de_salida_bluetooth = 1;
416 variable_de_salida_comando = 1;
417 variable_de_salida = 1 ;
418 digitalWrite(Relay_Bomba, LOW);

```

```

419 delay (100);
420 nexLoop(nex_listen_list) ;
421 delay (1);
422 nexLoop(nex_listen_list) ;
423 }
424
425 void boton_parada_control_2PushCallback(void *ptr)// Funcion que realiza cuando se
      apreta EL BOTON DE PARADA TANQUE 2
426 {
427   digitalWrite(Relay_Bomba, LOW);
428   delay (1);
429   variable_de_parada_control = 1;
430   variable_de_salida_bluetooth = 1;
431   variable_de_salida_comando = 1;
432   variable_de_salida = 1 ;
433   digitalWrite(Relay_Bomba, LOW);
434   Valvula_Cerrada();
435   nexLoop(nex_listen_list);
436   delay (10);
437   nexLoop(nex_listen_list);
438 }
439
440 void boton_inicio_control_1PushCallback(void *ptr)// Funcion que realiza cuando se
      apreta LA ENTRADA A INICIAR CONTROL 1
441 {
442   variable_de_salida_libre = 1;
443   variable_de_salida_bluetooth = 1;
444   variable_de_salida_comando = 1;
445   variable_de_parada_control = 0;
446
447   while (variable_de_parada_control !=1)
448   {
449     digitalWrite(33,LOW);
450     digitalWrite(35,LOW);
451     digitalWrite(37,LOW);
452     digitalWrite(39,LOW);
453     digitalWrite(41,LOW);
454     digitalWrite(43,LOW);
455     digitalWrite(45,LOW);
456
457     digitalWrite(31,HIGH);
458     delay (200);
459     digitalWrite(31,LOW);
460     delay (200);
461     digitalWrite(31,HIGH);
462     delay (200);
463     digitalWrite(31,LOW);
464     nexLoop(nex_listen_list) ;
465     delay (10);
466     Sensor_Mpx();
467     Sensor_Ultrasonido_control();
468     control_pi();
469   }
470 }
471
472 void boton_automatico_control_2PushCallback(void *ptr)// Funcion que realiza cuando
      se apreta LA ENTRADA A AUTOMATICO CONTROL 2
473 {
474   nexLoop(nex_listen_list) ;
475   variable_de_salida = 0;
476   Valvula_Cerrada();
477   unsigned long pasado2 = 0;
478   unsigned long actual2 = 0;
479   int32_t Cv2 = 0;
480   float errSum2 = 0;
481   float Pv2 = 0;
482   float Pv2_1 = 0;
483   int32_t error2 = 0;

```

```

484 int Iterm2 =0;
485 float dInput2 = 0;
486 uint32_t Kp2 = 0; //Ganancia Proporcional
487 uint32_t Ki2 = 0; //Ganancia Integral
488 uint32_t Kd2 = 0; //Ganancia Derivativa
489 float ki2 = 0 ;
490 float kd2 = 0 ;
491 float kp2 = 0 ;
492
493 while (d < 240 && variable_de_salida != 1)
494 {
495 digitalWrite(Relay_Bomba , HIGH);
496 Sensor_Ultrasonido();
497 Sensor_Ultrasonido_control_2();
498 delay (1);
499 nexLoop(nex_listen_list) ;
500 t0.setText("Cargando");
501 t1.setText("Tanque");
502 Desactivar();
503 }
504 digitalWrite(Relay_Bomba , LOW);
505 t0.setText(" ");
506 t1.setText(" ");
507 Activar();
508 }
509
510
511 void boton_inicio_control_2PushCallback(void *ptr)// Funcion que realiza cuando se
apreta LA ENTRADA A INICIAR CONTROL 2
512 {
513 variable_de_salida_libre = 1;
514 variable_de_salida_bluetooth = 1;
515 variable_de_salida_comando = 1;
516 variable_de_parada_control = 0;
517
518 while (variable_de_parada_control !=1)
519 {
520 digitalWrite(33,LOW);
521 digitalWrite(35,LOW);
522 digitalWrite(37,LOW);
523 digitalWrite(39,LOW);
524 digitalWrite(41,LOW);
525 digitalWrite(43,LOW);
526 digitalWrite(45,LOW);
527
528 digitalWrite(31,HIGH);
529 digitalWrite(33,HIGH);
530 delay (100);
531 digitalWrite(31,LOW);
532 digitalWrite(33,LOW);
533 delay (100);
534 digitalWrite(31,HIGH);
535 digitalWrite(33,HIGH);
536 delay (100);
537 digitalWrite(31,LOW);
538 digitalWrite(33,LOW);
539 nexLoop(nex_listen_list) ;
540 delay (100);
541 Sensor_Mpx();
542 Sensor_Mpx_control_2();
543 delay (1);
544 Sensor_Ultrasonido();
545 Sensor_Ultrasonido_control_2();
546 delay (1);
547 control_pid();
548
549 if (y >= 26)
550 {

```

```

551 Valvula_Cerrada();
552 }
553 }
554 }
555
556
557 void boton_entrada_comandoPopCallback(void *ptr)// Funcion que realiza cuando se
    apreta LA ENTRADA A MODO COMANDO
558 {
559     variable_de_salida_libre = 1;
560     variable_de_salida_bluetooth = 1;
561     variable_de_salida_comando = 0;
562     Valvula_Cerrada();
563     while (variable_de_salida_comando !=1)
564     {
565         nexLoop(nex_listen_list) ;
566         delay (2000);
567         Sensor_Mpx();
568         Sensor_Mpx_Modo_Comando();
569         Sensor_Ultrasonido();
570         Sensor_Ultrasonido_Comando();
571         delay (1);
572         Tira_Leds();
573         delay (1);
574     }
575 }
576
577 void boton_entrada_librePushCallback(void *ptr)//Funcion que realiza cuando se apreta
    se apreta LA ENTRADA A MODO LIBRE
578 {
579     variable_de_salida_comando = 1;
580     variable_de_salida_libre = 0;
581     variable_de_salida_bluetooth = 1;
582     Valvula_Cerrada();
583     while (variable_de_salida_libre !=1)
584     {
585         nexLoop(nex_listen_list) ;
586         delay (10);
587     }
588 }
589
590 void boton_entrada_bluetoothPushCallback(void *ptr)// Funcion que realiza cuando se
    apreta LA ENTRADA A MODO BLUETOOTH
591 {
592
593     variable_de_salida_comando = 1;
594     variable_de_salida_libre = 1;
595     variable_de_salida_bluetooth = 0;
596     Valvula_Cerrada();
597     int salida_bluetooth = 0 ;
598
599     while (variable_de_salida_bluetooth !=1)
600     {
601         nexLoop(nex_listen_list) ;
602         delay (2000);
603         Sensor_Mpx();
604         Sensor_Mpx_Modo_Bluetooth();
605         delay (1);
606         Sensor_Ultrasonido_Bluetooth();
607         delay (10);
608
609         if (Serial1.available()>0)
610         {
611             char dato= Serial1.read();
612             texto_bluetooth.setText("Conectado");
613             if (dato=='a')
614             {
615                 rBOMBA.Set_background_color_bco(1344);

```

```

616     digitalWrite (Relay_Bomba,HIGH);
617     delay (1);
618     }
619     else if (dato=='b')
620     {
621     rBOMBA.Set_background_color_bco(53248);
622     digitalWrite (Relay_Bomba,LOW);
623     delay (1);
624     }
625     else if (dato== 'c')
626     {
627     Valvula_Abierta();
628     delay (1);
629     rVALVULA.Set_background_color_bco(1344);
630     }
631     else if (dato== 'd')
632     {
633     Valvula_Cerrada();
634     rVALVULA.Set_background_color_bco(53248);
635     delay (1);
636     }
637     else if (dato== 's')
638     {
639     digitalWrite (Relay_Bomba,LOW);
640     Valvula_Cerrada();
641     rVALVULA.Set_background_color_bco(53248);
642     delay (1);
643     salida_bluetooth = 1;
644     variable_de_salida = 1;
645     pageManual.show();//funcion que me envie a la pagina anterior en la pantalla
646     }
647 }
648 }
649 }
650
651
652 void bt0PushCallback(void *ptr)//Funcion que realiza cuando se apreta BOMBA-MODO-
    COMANDOS de la pantalla
653 {
654 nexLoop(nex_listen_list) ;
655 uint32_t dual_state1;
656 memset(buffer, 0, sizeof(buffer));
657 bt0.getValue(&dual_state1);
658 if(dual_state1)
659 {
660 digitalWrite (Relay_Bomba,HIGH);
661 delay (1);
662 }
663 else {
664 digitalWrite (Relay_Bomba,LOW);
665 delay (1);
666 }
667 }
668
669
670 void btbombacontrolPushCallback(void *ptr)//Funcion que realiza cuando se apreta
    BOMBA-MODO-COMANDOS de la pantalla
671 {
672 uint32_t dual_state3;
673 memset(buffer, 0, sizeof(buffer));
674 btbombacontrol.getValue(&dual_state3);
675 if(dual_state3)
676 {
677 digitalWrite (Relay_Bomba,HIGH);
678 delay (1);
679 }
680 else {
681 digitalWrite (Relay_Bomba,LOW);

```

```

682 delay (1);
683 }
684 }
685
686 void bt1PushCallback(void *ptr)//Funcion que realiza cuando se apreta VALVULA MODO-
        COMANDOS de la pantalla
687 {
688 uint32_t dual_state;
689 memset(buffer, 0, sizeof(buffer));
690 bt1.getValue(&dual_state);
691 if(dual_state)
692 {
693 Valvula_Abierta();
694 delay (1);
695 }
696 else {
697 Valvula_Cerrada();
698 delay (1);
699 }
700 }
701
702
703 void cMPXPushCallback(void *ptr)//Funcion que realiza cuando se apreta SENSOR MPX
        MODO LIBRE de la pantalla
704 {
705 uint32_t estadoMPX;
706 memset(buffer, 0, sizeof(buffer));
707 cMPX.getValue(&estadoMPX);
708 if(estadoMPX)
709 {
710 digitalWrite(Relay_sensor_mpx_manual, HIGH );
711 delay (1);
712 }
713 else {
714 digitalWrite(Relay_sensor_mpx_manual, LOW);
715 delay (1);
716 }
717 }
718
719 void cSONIDOPushCallback(void *ptr)// Funcion que realiza cuando se apreta
        ULTRASONIDO MODO LIBRE de la pantalla
720 {
721 uint32_t estadoSONIDO;
722 memset(buffer, 0, sizeof(buffer));
723 cSONIDO.getValue(&estadoSONIDO);
724 if(estadoSONIDO)
725 {
726 Tira_Leds();
727 digitalWrite(Relay_sensor_ultrasonido, HIGH );
728 digitalWrite(Relay_sensor_ultrasonido_hecho, HIGH );
729 digitalWrite(10, HIGH );
730 delay (1);
731 }
732 else {
733 digitalWrite(Relay_sensor_ultrasonido, LOW );
734 digitalWrite(Relay_sensor_ultrasonido_hecho, LOW );
735 digitalWrite(10, LOW );
736 delay (1);
737 }
738 }
739
740 void cBOMBAPushCallback(void *ptr)// Funcion que realiza cuando se apreta BOMBA MODO
        LIBRE de la pantalla
741 {
742 uint32_t estadoBOMBA;
743 memset(buffer, 0, sizeof(buffer));
744 cBOMBA.getValue(&estadoBOMBA);
745 if(estadoBOMBA)

```

```

746 {
747 digitalWrite(Relay_Bomba_manual, HIGH );
748 delay (1);
749 }
750 else {
751 digitalWrite(Relay_Bomba_manual, LOW );
752 delay (1);
753 }
754 }
755
756 void cVALVULAPushCallback(void *ptr)// Funcion que realiza cuando se apreta VALVULA
      MODO LIBRE de la pantalla
757 {
758 uint32_t estadoVALVULA;
759 memset(buffer, 0, sizeof(buffer));
760 cVALVULA.getValue(&estadoVALVULA);
761 if(estadoVALVULA)
762 {
763 variable_de_salida_libre = 0;
764 while (variable_de_salida_libre !=1)
765 {
766 nexLoop(nex_listen_list) ;
767 delay (100);
768 digitalWrite(Valvula, HIGH );
769 delay (100);
770 digitalWrite(Valvula, LOW );
771 delay (100);
772 posicion_valvula();
773 }
774 }
775 else {
776 variable_de_salida_libre = 1;
777 digitalWrite(Valvula, LOW );
778 delay (1);
779 }
780 }
781
782 /*****
783 ** Descripcion de funciones *****/
784 *****/
785
786 void Sensor_Ultrasonido()
787 {
788 nexLoop(nex_listen_list) ;
789 digitalWrite(Triple, LOW);
790 delayMicroseconds(4);
791 digitalWrite(Triple, HIGH);
792 delayMicroseconds(10); //Enviamos un pulso de 10us
793 digitalWrite(Triple, LOW);
794 t = pulseIn(Echo, HIGH); //obtenemos el ancho del pulso
795 d = 296-(t/5.9); //escalamos el tiempo a una distancia en mm tenia 292
796 delay(10);
797
798 e =((d)*100)/300; //escalamos el tiempo porcentualmente para la barra de progreso en
      la pantalla Nextion.
799 }
800
801 void Sensor_Ultrasonido_Bluetooth()
802 {
803 nexLoop(nex_listen_list) ;
804 nOBluetooth.setValue(d);
805 delay(5);
806 nexLoop(nex_listen_list) ;
807 delay(5);
808 jOBluetooth.setValue(e);
809 }
810
811 void Sensor_Ultrasonido_Comando()

```

```

812 {
813   nexLoop(nex_listen_list);
814   nOcomando.setValue(d);
815   delay(5);
816   nexLoop(nex_listen_list);
817   delay(5);
818   jOcomando.setValue(e);
819 }
820
821 void Sensor_Ultrasonido_control()
822 {
823   nexLoop(nex_listen_list) ;
824   jc.setValue(e);
825   delay(2);
826   nexLoop(nex_listen_list) ;
827   nc.setValue(d);
828   delay(1);
829 }
830
831 void Sensor_Ultrasonido_control_2()
832 {
833   nexLoop(nex_listen_list) ;
834   n1_control_2.setValue(d);
835   delay(5);
836   nexLoop(nex_listen_list) ;
837   delay(5);
838   barra_tanque_1.setValue(e);
839 }
840
841 void Tira_Leds()
842 {
843   Sensor_Ultrasonido();
844   delay(8);
845
846   if (d < 30)
847   {
848     digitalWrite(31, LOW);
849     digitalWrite(33, LOW);
850     digitalWrite(35, LOW);
851     digitalWrite(37, LOW);
852     digitalWrite(39, LOW);
853     digitalWrite(41, LOW);
854     digitalWrite(43, LOW);
855     digitalWrite(45, LOW);
856   }
857   else if (d > 30 && d <= 30.75)
858   {
859     digitalWrite(31, HIGH);
860     digitalWrite(33, LOW);
861     digitalWrite(35, LOW);
862     digitalWrite(37, LOW);
863     digitalWrite(39, LOW);
864     digitalWrite(41, LOW);
865     digitalWrite(43, LOW);
866     digitalWrite(45, LOW);
867   }
868   else if ( d > 30.75 && d <= 70.50)
869   {
870     digitalWrite(31, HIGH);
871     digitalWrite(33, HIGH);
872     digitalWrite(35, LOW);
873     digitalWrite(37, LOW);
874     digitalWrite(39, LOW);
875     digitalWrite(41, LOW);
876     digitalWrite(43, LOW);
877     digitalWrite(45, LOW);
878   }
879   else if ( d > 70.50 && d <= 110.25)

```

```

880  {
881  digitalWrite(31, HIGH);
882  digitalWrite(33, HIGH);
883  digitalWrite(35, HIGH);
884  digitalWrite(37, LOW);
885  digitalWrite(39, LOW);
886  digitalWrite(41, LOW);
887  digitalWrite(43, LOW);
888  digitalWrite(45, LOW);
889  }
890  else if ( d > 110.25 && d <= 150)
891  {
892  digitalWrite(31, HIGH);
893  digitalWrite(33, HIGH);
894  digitalWrite(35, HIGH);
895  digitalWrite(37, HIGH);
896  digitalWrite(39, LOW);
897  digitalWrite(41, LOW);
898  digitalWrite(43, LOW);
899  digitalWrite(45, LOW);
900  }
901  else if ( d > 150 && d <= 180.75)
902  {
903  digitalWrite(31, HIGH);
904  digitalWrite(33, HIGH);
905  digitalWrite(35, HIGH);
906  digitalWrite(37, HIGH);
907  digitalWrite(39, HIGH);
908  digitalWrite(41, LOW);
909  digitalWrite(43, LOW);
910  digitalWrite(45, LOW);
911  }
912  else if ( d > 180.75 && d <= 220.50)
913  {
914  digitalWrite(31, HIGH);
915  digitalWrite(33, HIGH);
916  digitalWrite(35, HIGH);
917  digitalWrite(37, HIGH);
918  digitalWrite(39, HIGH);
919  digitalWrite(41, HIGH);
920  digitalWrite(43, LOW);
921  digitalWrite(45, LOW);
922  }
923  else if ( d > 220.50 && d <= 260.25)
924  {
925  digitalWrite(31, HIGH);
926  digitalWrite(33, HIGH);
927  digitalWrite(35, HIGH);
928  digitalWrite(37, HIGH);
929  digitalWrite(39, HIGH);
930  digitalWrite(41, HIGH);
931  digitalWrite(43, HIGH);
932  digitalWrite(45, LOW);
933  }
934  else if ( d > 260.50 )
935  {
936  digitalWrite(31, HIGH);
937  digitalWrite(33, HIGH);
938  digitalWrite(35, HIGH);
939  digitalWrite(37, HIGH);
940  digitalWrite(39, HIGH);
941  digitalWrite(41, HIGH);
942  digitalWrite(43, HIGH);
943  digitalWrite(45, HIGH);
944  digitalWrite(Relay_Bomba, LOW);
945  }
946  Sensor_Mpx();
947  if (y >= 26)

```

```

948 {
949   Valvula_Cerrada();
950 }
951 }
952
953
954 void Valvula_Cerrada()
955 {
956   nexLoop(nex_listen_list) ;
957   stepper.runToNewPosition(0);
958   delay(100);
959 }
960
961
962 void Valvula_Abierta()
963 {
964   nexLoop(nex_listen_list) ;
965   stepper.runToNewPosition(-1000);
966   delay(100);
967 }
968
969
970 void Sensor_Mpx()
971 {
972   sensorValue_MPX = analogRead(A0) ;
973   delay(10);
974
975   y = (0.09 * sensorValue_MPX) - 4.35;           // Valor del sensor en cm
976   yb = ((0.09 * sensorValue_MPX) - 4.35)* 10 ;   // Valor del sensor en mm
977
978   delay(1);
979   barra = (yb*100)/300;
980   nexLoop(nex_listen_list);
981 }
982
983 void Sensor_Mpx_Modo_Bluetooth()
984 {
985   delay(1);
986   nexLoop(nex_listen_list) ;
987   j1Bluetooth.setValue(barra);           //Envio a modo BLUETOOTH
988   delay(1);
989   n1Bluetooth.setValue(yb);             //Envio a modo BLUETOOTH
990   nexLoop(nex_listen_list) ;
991 }
992
993
994 void Sensor_Mpx_Modo_Comando()
995 {
996   delay(1);
997   j1comando.setValue(barra);           //Envio a modo COMANDO
998   nexLoop(nex_listen_list) ;
999   delay(1);
1000  nicomando.setValue(yb);             //Envio a modo COMANDO
1001  nexLoop(nex_listen_list) ;
1002  delay(1);
1003 }
1004
1005
1006 void Sensor_Mpx_control()
1007 {
1008   delay(1);
1009   nexLoop(nex_listen_list) ;
1010   jc.setValue(barra);
1011   delay(1);
1012
1013   nc.setValue(yb);
1014   nexLoop(nex_listen_list) ;
1015   delay(1);

```

```

1016 }
1017
1018 void Sensor_Mpx_control_2()
1019 {
1020   delay(1);
1021   nexLoop(nex_listen_list) ;
1022   barra_tanque_2.setValue(barra);
1023   delay(1);
1024
1025   n2_control_2.setValue(yb);
1026   nexLoop(nex_listen_list) ;
1027   delay(1);
1028 }
1029
1030 void control_pi()
1031 {
1032   actual = millis(); //Milisegundos desde que se inicio el programa
1033   unsigned long dt = actual - pasado;
1034
1035   nexLoop(nex_listen_list) ;
1036
1037   int j=0;
1038   float suma_sonido =0;
1039
1040   digitalWrite(Relay_Bomba, HIGH);
1041
1042   if ( d > 240 ) // Condicion para control on-off bomba
1043   {
1044     digitalWrite(Relay_Bomba, LOW);
1045   }
1046
1047   memset(buffer, 0, sizeof(buffer));
1048   nsetpoint.getValue(&Sp); // Recibo el set point enviado desde la pantalla
1049   delay(10);
1050   nP.getValue(&Kp); // Recibo el Kp enviado desde la pantalla
1051   delay(10);
1052   nI.getValue(&Ki); // Recibo el Ki enviado desde la pantalla
1053   delay(10);
1054   nD.getValue(&Kd); // Recibo el Kd enviado desde la pantalla
1055
1056
1057
1058
1059   if(dt >= 1000 ) // T Periodo de muestreo
1060   {
1061
1062     for(j=0;j<50;j++)
1063     {
1064       digitalWrite(Triple, LOW);
1065       delayMicroseconds(4);
1066       digitalWrite(Triple, HIGH);
1067       delayMicroseconds(10); //Enviamos un pulso de 10us
1068       digitalWrite(Triple, LOW);
1069
1070       t = pulseIn(Echo, HIGH); //obtenemos el ancho del pulso
1071       d = 296-(t/5.9); //escalamos el tiempo a una distancia en mm
1072       delay(10);
1073       suma_sonido = suma_sonido+d;
1074     }
1075
1076     d = suma_sonido/50;
1077     delay(2);
1078     e =((d)*100)/300 ; //escalamos el tiempo porcentualmente para la barra de progreso
1079     // en la pantalla Nextion.
1080
1081     kp = (Kp);
1082     ki = (0.01 * Ki * T);
1083     kd = ( Kd / T);

```

```

1083 delay(1);
1084
1085     Pv = d/10 ;           // Obtengo el valor del sensor de ultrasonido en cm
1086
1087     error = ( Sp - Pv );
1088
1089     delay(1);
1090     Iterm += ((-ki) * error);
1091
1092     if(Iterm > 1000)
1093     {
1094         Iterm = 1000;
1095     }
1096     else if(Iterm < 0)
1097     {
1098         Iterm = 0;
1099     }
1100
1101     dInput1 = (Pv - Pv_1) ;
1102
1103     Cv = ((-kp) * error) + Iterm - ((-kd) * dInput1);
1104     delay(1);
1105
1106     if (Cv > 1000)
1107     {
1108         Cv = 1000;
1109     }
1110     else if (Cv < 0)
1111     {
1112         Cv = 0;
1113     }
1114
1115     PWM = map(Cv,0,1000,-0,-1000);
1116     delay(1);
1117
1118     stepper.runToNewPosition(PWM);
1119     delay(1);
1120
1121     Pv_1 = Pv;
1122     pasado = actual;
1123     delay(1);
1124
1125     onda_control_1.addValue(0,Sp); // Envio set point a la grafica de la pantalla
1126     onda_control_1.addValue(1,Pv); // Envio valor del sensor a la grafica de la
    pantalla
1127 }
1128 }
1129
1130
1131 void control_pid()
1132 {
1133     actual2 = millis(); //Milisegundos desde que se inicio el programa
1134     unsigned long dt2 = (actual2 - pasado2) ;
1135     nexLoop(nex_listen_list);
1136
1137     if ( d > 250 ) // Condicion para control on-off bomba
1138     {
1139         digitalWrite(Relay_Bomba, LOW);
1140     }
1141
1142     memset(buffer, 0, sizeof(buffer));
1143     nsetpoint_control_2.getValue(&Sp2); //Recibo el set point desde la pantalla
1144     delay(10);
1145     nP2.getValue(&Kp2); //Recibo el Kp enviado desde la pantalla
1146     delay(10);
1147     nI2.getValue(&Ki2); //Recibo el Ki enviado desde la pantalla
1148     delay(10);
1149     nD2.getValue(&Kd2); //Recibo el Ki enviado desde la pantalla

```

```

1150
1151         if(dt2 >= 1000    ) // T Muestreo
1152     {
1153
1154         digitalWrite(Trigger, LOW);
1155         delayMicroseconds(4);
1156         digitalWrite(Trigger, HIGH);
1157         delayMicroseconds(10);           //Enviamos un pulso de 10us
1158         digitalWrite(Trigger, LOW);
1159
1160         t = pulseIn(Echo, HIGH);           //obtenemos el ancho del pulso
1161         d = 292-(t/5.9);                   //escalamos el tiempo a una distancia en mm
1162         int v = d/10;
1163
1164         e =((d)*100)/300;//escalamos el tiempo porcentualmente para la barra de progreso en
1165         la pantalla Nextion.
1166         delay(1);
1167
1168         Sensor_Mpx();
1169         delay(1);
1170
1171         ki2 = (0.01 * Ki2 * T2 );
1172         kd2 = (Kd2 / T2 );
1173         kp2 = Kp2;
1174         delay(1);
1175
1176         Pv2 = y ;
1177
1178         Sp2_double = Sp2;
1179
1180         error2 = ( Sp2_double - Pv2 ); //Calculo del error
1181
1182         delay(1);
1183         Iterm2 += (ki2*error2);
1184
1185         if (Iterm2 > 1000)
1186         {
1187             Iterm2 = 1000;
1188         }
1189         else if (Iterm2 < 0)
1190         {
1191             Iterm2 = 0;
1192         }
1193
1194         dInput2 = (Pv2 - Pv2_1);
1195
1196         Cv2 = (kp2 * error2) + Iterm2 - (kd2 * dInput2);
1197         delay(1);
1198
1199         if (Cv2 > 1000)
1200         {
1201             Cv2 = 1000;
1202         }
1203         else if (Cv2 < 0)
1204         {
1205             Cv2 = 0 ;
1206         }
1207
1208         PWM_2 = map(Cv2,0,1000,0,-1000);
1209         delay(1);
1210
1211         stepper.runToNewPosition(PWM_2); // Muevo la valvula
1212         delay(1);
1213
1214         Pv2_1 = Pv2;
1215         pasado2 = actual2;
1216
1217         onda_control_2.addValue(0,Sp2);
1218         onda_control_2.addValue(1,Pv2);

```

```

1217     onda_control_2.addValue(2,v);
1218 }
1219 }
1220
1221 void Envio_Tx_App()
1222 {
1223     char buffer[8];
1224     Sensor_Mpx();
1225     Sensor_Ultrasonido();
1226     sprintf(buffer, "%d,%d",yb,d);
1227     Serial1.print(buffer);
1228     delay(500);
1229 }
1230
1231 void posicion_valvula()
1232 {
1233     int giro = 0 ;
1234     Posicion_valvula = (analogRead(A2)) ;
1235     tension_posicion_valvula = Posicion_valvula*(5/1023);
1236     giro = map(Posicion_valvula,0,1023,0,-1000);
1237     delay(10);
1238     stepper.runToNewPosition(giro);
1239     delay(10);
1240 }
1241
1242 void Potenciometro_Valvula()
1243 {
1244     float tension_potenciometro_valvula ;
1245     float pote_valvula = (analogRead(A1)) ;
1246     tension_potenciometro_valvula = ((pote_valvula * 5.0) / 1023.0);
1247 }
1248
1249 void Activar()
1250 {
1251     Serial.print("tsw n6,1");
1252     Serial.write(0xff);
1253     Serial.write(0xff);
1254     Serial.write(0xff);
1255     delay(5);
1256
1257     Serial.print("tsw nP2,1");
1258     Serial.write(0xff);
1259     Serial.write(0xff);
1260     Serial.write(0xff);
1261     delay(5);
1262
1263     Serial.print("tsw nI2,1");
1264     Serial.write(0xff);
1265     Serial.write(0xff);
1266     Serial.write(0xff);
1267     delay(5);
1268
1269     Serial.print("tsw nD2,1");
1270     Serial.write(0xff);
1271     Serial.write(0xff);
1272     Serial.write(0xff);
1273     delay(5);
1274
1275     Serial.print("tsw b2,1");
1276     Serial.write(0xff);
1277     Serial.write(0xff);
1278     Serial.write(0xff);
1279     delay(5);
1280
1281     Serial.print("tsw b0,1");
1282     Serial.write(0xff);
1283     Serial.write(0xff);
1284     Serial.write(0xff);

```

```

1285 delay(5);
1286
1287 Serial.print("tsw btbombacontrol,1");
1288 Serial.write(0xff);
1289 Serial.write(0xff);
1290 Serial.write(0xff);
1291 delay(5);
1292
1293 Serial.print("tsw b3,1");
1294 Serial.write(0xff);
1295 Serial.write(0xff);
1296 Serial.write(0xff);
1297 delay(5);
1298 }
1299
1300 void Desactivar()
1301 {
1302 Serial.print("tsw n6,0");
1303 Serial.write(0xff);
1304 Serial.write(0xff);
1305 Serial.write(0xff);
1306 delay(5);
1307
1308 Serial.print("tsw nP2,0");
1309 Serial.write(0xff);
1310 Serial.write(0xff);
1311 Serial.write(0xff);
1312 delay(5);
1313
1314 Serial.print("tsw nI2,0");
1315 Serial.write(0xff);
1316 Serial.write(0xff);
1317 Serial.write(0xff);
1318 delay(5);
1319
1320 Serial.print("tsw nD2,0");
1321 Serial.write(0xff);
1322 Serial.write(0xff);
1323 Serial.write(0xff);
1324 delay(5);
1325
1326 Serial.print("tsw b2,0");
1327 Serial.write(0xff);
1328 Serial.write(0xff);
1329 Serial.write(0xff);
1330 delay(5);
1331
1332 Serial.print("tsw b0,0");
1333 Serial.write(0xff);
1334 Serial.write(0xff);
1335 Serial.write(0xff);
1336 delay(5);
1337
1338 Serial.print("tsw btbombacontrol,0");
1339 Serial.write(0xff);
1340 Serial.write(0xff);
1341 Serial.write(0xff);
1342 delay(5);
1343
1344 Serial.print("tsw b3,0");
1345 Serial.write(0xff);
1346 Serial.write(0xff);
1347 Serial.write(0xff);
1348 delay(5);
1349 }

```

Listing C.1: orden.

Apéndice D

Recomendaciones

La vida útil del equipo depende de su buen uso, por lo tanto, es necesario observar con cuidado y atención estas normas y recomendaciones. Además, se considera **importante e indispensable** conocer las indicaciones y normas de seguridad básicas para garantizar una utilización segura y exenta de fallas durante la puesta en marcha del sistema.

En cuanto al usuario, se genera un compromiso de responsabilidad en el momento del uso y manipulación del sistema didáctico, mediante una supervisión en lo referente a las normas de seguridad y prevención de accidentes que podrían causar daños tanto al usuario como a los elementos que contiene la unidad.

- Antes de encender la unidad, se debe cargar el tanque reservorio hasta la marca ubicada en un extremo y ajustar todos los acoples universales de las cañerías para evitar pérdidas.
- Una vez encendida la unidad, el sistema ingresa en un modo automático de cierre de electroválvula, se recomienda aguardar hasta que este modo termine de realizar su función para recién utilizar la pantalla con sus acciones, esto evita fallas en el software.
- Se recomienda **accionar periódicamente** la bomba, al menos una vez por semana, para evitar que el eje se trabe por oxidación. Si la bomba no succiona el agua luego de 2 minutos encendida, es factible que necesite ser cebada agregando agua al tornillo ubicado en la parte superior de la misma. En caso de querer sacar la bomba para mantenimiento se lo puede realizar fácilmente desde el acople universal.

- Luego de cada uso de la unidad, es necesario que **todos** los tanques se encuentren vacíos, esto evita manchas en los acrílicos y posibles bacterias que junten mal olor.
- Evitar el uso de detergentes u otros líquidos que puedan corroer la cañería.
- Se recomienda al finalizar el uso de la unidad, bajar el tablero de conexiones para el modo didáctico. Esto evita que puedan ingresar impurezas al interior y produzca perturbaciones en las conexiones.
- Probar periódicamente el interruptor parada de emergencia de la bomba.
- Al momento de desalojar el agua en las tuberías de la unidad, **primero se debe** abrir la válvula de desagote que se encuentra debajo del tanque N°2. Luego la válvula que se encuentra entre ambos tanques. Esto evita que al abrir la válvula del medio, el agua ingrese al tanque N°2 por la vía de desagote.

Bibliografía

- [1] **NMC Horizon Report: 2017 Higher Education Edition.**
Austin, Texas: The New Media Consortium. Enero 2017.
<https://library.educause.edu>.
- [2] **Biblioteca virtual IEEE.**
<https://ieeexplore.ieee.org>
- [3] **Samir S. Soliman and Mandyam D. Srinath**
Señales y Sistemas Continuos-Discretos- 2da Edición, Madrid 1999.
- [4] **Carmona, Aníbal Isidoro. Número de Reynolds.**
Aerodinámica y Actuaciones del avión- 2004.

Referencias de Figuras

- [5] **R. Routledge, 13th edition.**
Discoveries and Inventions of the Nineteenth Century - 1900
- [6] **Fuente**
Butterworth filter Bode Plot - Octubre 2005
https://es.wikipedia.org/diagrama_de_bode
- [7] **Fuente**
Multi-loop Control System.
<http://www.forum.coppeliarobotics.com>
- [8] **Katsuhiko Ogata, 5ta Edición**
Ingeniería de Control Moderna.
Pearson Educación, S.A., Madrid, 2010-Página 21.

- [9] **Benjamín Kuo, 7ma Edición**
Sistema de Control Automático.
Prentice-Hall, S.A., México, 1996-Página 9
- [10] **Norman S. Nise, United States, 6ta Edición**
Control System Engineering - 2011.
- [11] **Raytheon, Marine American**
Defense Department, USA - 2014
- [12] **Benjamín Kuo, 7ma Edición**
Sistema de Control Automático.
Prentice-Hall, S.A., México, 1996
- [13] **Oppenheim, Alan V.**
Señales y Sistemas
Prentice Hall, Inc., New Jersey, 1989-Página 5
- [14] **Benjamín Kuo, 7ma Edición**
Sistema de Control Automático.
Prentice-Hall, S.A., México, 1996-Página 17
- [15] **Benjamín Kuo, 7ma Edición**
Sistema de Control Automático.
Prentice-Hall, S.A., México, 1996-Página 18
- [16] **Katsuhiko Ogata, 5ta Edición**
Ingeniería de Control Moderna.
Pearson Educación, S.A., Madrid, 2010-Página 17.
- [17] **Katsuhiko Ogata, 5ta Edición**
Ingeniería de Control Moderna.
Pearson Educación, S.A., Madrid, 2010-Página 19.
- [18] **Katsuhiko Ogata, 5ta Edición**
Ingeniería de Control Moderna.
Pearson Educación, S.A., Madrid, 2010-Página 23.
- [19] **Katsuhiko Ogata, 5ta Edición**
Ingeniería de Control Moderna.
Pearson Educación, S.A., Madrid, 2010-Página 571.

- [20] **Fuente**
Diagrama PID- 2015
<https://controlautomaticoeducacion.com>
- [21] **Katsuhiko Ogata, 5ta Edición**
Ingeniería de Control Moderna.
Pearson Educación, S.A., Madrid, 2010-Página 102.
- [22] **Katsuhiko Ogata, 5ta Edición**
Ingeniería de Control Moderna.
Pearson Educación, S.A., Madrid, 2010-Página 104.
- [23] **Integrado MOC 3041**
Datasheet del componente.
- [24] **Ing. Marcelo D'amore**
Apuntes vistos en clases de Electrónica 3 - 2015
- [25] **Fuente**
Dimmer. Recurso web.
<https://es.wikipedia.org/wiki/Triac>
- [26] **Fuente**
Triac. Recurso web.
<https://www.monografias.com>
- [27] **Fuente**
Arduino Mega 2560.
<https://www.arduino.cc>
- [28] **Fuente**
Matlab - 2019
<https://la.mathworks.com/products/matlab.html>
- [29] **Fuente**
LabVIEW - 2019
<http://www.ni.com/es-mx/shop/labview.html>
- [30] **Fuente**
Nextion - 2019
<https://www.itead.cc/nextion-nx8048t070.html>

- [31] **Fuente**
Overleaf - 2019
<https://www.overleaf.com/>
- [32] **Fuente**
Propiedad del autor - 2019
Barrionuevo, Julian Daniel
- [33] **Creus, Antonio**
Instrumentación Industrial.
Alfaomega Grupo Editor, S.A., México, 2010.
- [34] **Sensor Mpx5010dp**
[*Datasheet del componente.*](#)
- [35] **Sensor HC-SR04**
[*Datasheet Sensor HC-SR04.*](#)

Componentes, Dispositivos y Equipos

- [36] **Arduino Mega 2560**
[*Datasheet del componente Atmega 2560.*](#)
[*Distribución de pines Atmega 2560.*](#)
- [37] **Motor STH - 55D121.**
[*Datasheet Motor STH - 55D121*](#)
- [38] **Texas Instruments**
[*Datasheet del componente L923D.*](#)
- [39] **Electrobomba Periférica Plumita**
[*Enlace Mercado Libre.*](#)
- [40] **ST Microelectronics**
[*Datasheet del componente TIP 3055.*](#)
- [41] **NXP Semiconductors**
[*Nota de Aplicación del Sensor MPX5010.*](#)

Recursos Web

- [42] NOTA DESCRIPTIVA
Unidad didáctica - Santiago de Chile, Agosto 2018
<https://educrea.cl/una-unidad-didactica-paso-paso/>
- [43] NOTA DESCRIPTIVA
Informe Banco mundial - NW Washington DC USA, Setiembre 2017
<https://www.bancomundial.org>
- [44] NOTA DESCRIPTIVA
Clarín - Argentina 2017
<https://www.clarin.com/campus/convenios-universidades-empresas>
- [45] INFORME DESCRIPTIVO
Desarrollo de un laboratorio de sistemas industriales - Venezuela, 2010
<https://www.researchgate.net>
- [46] PÁGINA WEB
The Industrial Interoperability StandardTM - 2019
<https://opcfoundation.org/about/what-is-opc/>
- [47] PÁGINA WEB
Control derivativo - 2019
https://www.ecured.cu/Control_derivativo
- [48] PÁGINA WEB
Aislamiento Galvánico - 2019
https://es.wikipedia.org/wiki/Aislamiento_galv%C3%A1nico
- [49] PÁGINA WEB
Relé de estado sólido - 2019
https://en.wikipedia.org/wiki/Solid-state_relay