DEPARTAMENTO DE ELÉCTRICA Y ELÉCTRONICA

CARRERA DE INGENIERÍA EN ELECTRÓNICA E INSTRUMENTACIÓN

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO ELECTRÓNICO EN INSTRUMENTACIÓN

TEMA: “HARDWARE-IN-THE-LOOP PARA EL CONTROL DE LOS PROCESOS PRESIÓN, CAUDAL Y NIVEL, MEDIANTE EL SISTEMA EMBEBIDO MY RIO A REALIZARSE EN EL LABORATORIO DE REDES INDUSTRIALES Y CONTROL DE PROCESOS”

AUTORES: ALULEMA GARZÓN, JUAN CARLOS
MORA NÚÑEZ, NICOLÁS SEBASTIÁN

DIRECTOR: ING. PRUNA PANCHI, EDWIN PATRICIO

LATACUNGA

2018

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA EN ELECTRÓNICA E INSTRUMENTACIÓN

CERTIFICACIÓN

Certifico que el trabajo de titulación, **"HARDWARE-IN-THE-LOOP PARA EL CONTROL DE LOS PROCESOS PRESIÓN, CAUDAL Y NIVEL, MEDIANTE EL SISTEMA EMBEBIDO MY RIO A REALIZARSE EN EL LABORATORIO DE REDES INDUSTRIALES Y CONTROL DE PROCESOS"** realizado por los señores, **ALULEMA GARZÓN JUAN CARLOS Y MORA NÚÑEZ NICOLÁS SEBASTIÁN** ha sido revisado en su totalidad y analizado por el software anti-plagio, el mismo cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de las Fuerzas Armadas ESPE, por lo tanto me permito acreditarlo y autorizar a los señores **ALULEMA GARZÓN JUAN CARLOS Y MORA NÚÑEZ NICOLÁS SEBASTIÁN** para que lo sustenten públicamente.

Latacunga, de agosto del 2018

Ing. Pruna Panchi, Edwin Patricio

DIRECTOR

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA EN ELECTRÓNICA E INSTRUMENTACIÓN

AUTORÍA DE RESPONSABILIDAD

Nosotros, **Alulema Garzón Juan Carlos**, con cédula de identidad N°1804600573 y **Mora Núñez Nicolás Sebastián**, con cédula de identidad N°1804551834 declaro que el presente trabajo de titulación, **"HARDWARE-IN-THE-LOOP PARA EL CONTROL DE LOS PROCESOS PRESIÓN, CAUDAL Y NIVEL, MEDIANTE EL SISTEMA EMBEBIDO MY RIO A REALIZARSE EN EL LABORATORIO DE REDES INDUSTRIALES Y CONTROL DE PROCESOS"** ha sido desarrollado considerando los métodos de investigación existentes, así como también se ha respetado los derechos intelectuales de terceros considerándose en las citas bibliográficas.

Consecuentemente declaro que este trabajo es de nuestra autoría, en virtud de ello nos declaramos responsables del contenido, veracidad y alcance de la investigación mencionada.

Latacunga, de agosto del 2018

Alulema Garzón, Juan Carlos

C.C.: 1804600573

Mora Núñez Nicolás, Sebastián

C.C.: 1804551834

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA EN ELECTRÓNICA E INSTRUMENTACIÓN

AUTORIZACIÓN

Nosotros, **ALULEMA GARZÓN JUAN CARLOS Y MORA NÚÑEZ NICOLÁS SEBASTIÁN**, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar en la Biblioteca Virtual de la institución el presente trabajo de titulación **"HARDWARE-IN-THE-LOOP PARA EL CONTROL DE LOS PROCESOS PRESIÓN, CAUDAL Y NIVEL, MEDIANTE EL SISTEMA EMBEBIDO MY RIO A REALIZARSE EN EL LABORATORIO DE REDES INDUSTRIALES Y CONTROL DE PROCESOS"** cuyo contenido, ideas y criterios son de nuestra autoría y responsabilidad.

Latacunga, de agosto del 2018

Alulema Garzón, Juan Carlos

C.C.: 1804600573

Mora Núñez, Nicolás Sebastián

C.C.: 1804551834

DEDICATORIA

Dedico este trabajo a mis padres que son mi fortaleza en momentos de flaqueza, juntos me dedicaron su amor y tiempo en voto de confianza para ver a su hijo crecer, mi devoción y gratitud hacia ustedes, para mí son el ejemplo del trabajo constante. A mi familia que me enseño el significado de la unión, formaron parte indiscutible en mi formación, siempre estuvieron presente en los momentos gratos y malos. Este logro es la recompensa del esfuerzo por construir un camino basado en el trabajo duro y honesto.

Nicolás S. Mora N.

DEDICATORIA

Este trabajo de largas noches de estudios, desvelos, de contratiempos y satisfacciones los dedico a mi Dios por darme la fortaleza y sabiduría para poder llegar a culminar otra meta en mi vida; a mis padres quienes han sido pilares fundamentales con su apoyo incondicional y palabras de aliento en mis momentos más difíciles en que me sentía quebrantar supieron levantarme para seguir adelante; a mis tíos ejemplo de vida y de superación fueron inspiración y aliento de lucha. A mis abuelitos y demás familiares que me abrigaron con su amor y dedicación.

A mis maestros que supieron impartir sus conocimientos para que hoy con este trabajo vean reflejado que su esfuerzo no ha sido en vano.

Juan Carlos Alulema G.

AGRADECIMIENTO

Agradezco a Dios y a la vida que son el origen de mi equilibrio. A mis padres que siempre me demostraron el ejemplo del deber cumplido y el amor incondicional que te brinda una familia. A mis hermanos Xavier y Andrés que siempre fueron el pilar en mi formación personal y profesional. Para Diana una persona que admiro y me brindó su apoyo en todos los momentos transmitiéndome las mejores enseñanzas. Agradezco al ingeniero Edwin Pruna que fue un gran mentor en la elaboración del proyecto que con esfuerzo y dedicación alcanzamos un gran logro. Deseándole los mejores éxitos en el paso de la vida. Gracias por los consejos y motivación serán el mejor aprendizaje para afrontar el siguiente paso de mi vida. A mi familia y amigos que a su manera siempre me transmitieron su cariño y preocupación. Ustedes fueron son un gran impulso para culminar mis metas.

Nicolás S. Mora N.

AGRADECIMIENTO

Agradezco primero a Dios por ser el inspirador y darme fuerza en este proceso y lucha constante, porque me ha guiado en cada paso de esta hermosa experiencia universitaria, y darme esa fuerza necesaria para continuar en este proceso de obtener mi anhelo más deseado, luego a mis padres por darme su amor incondicional, por su trabajo y sacrificio en todos estos años he llegado al lugar en donde estoy y por ser el pilar fundamental para seguir de pie día a día a pesar de las circunstancias que se han presentado, son mi ejemplo a seguir, no puede faltar mi agradecimiento a mis abuelitos que han estado a mi lado consintiéndome e incentivándome a cumplir mis sueños.

Juan Carlos Alulema G.

ÍNDICE DE CONTENIDOS

CARÁTULA
CERTIFICACIÓN ii
AUTORÍA DE RESPONSABILIDAD iii
AUTORIZACIÓN iv
DEDICATORIA v
AGRADECIMIENTO vii
ÍNDICE DE CONTENIDOS ix
ÍNDICE DE FIGURAS xii
ÍNDICE DE TABLAS xiv
RESUMEN xv
ABSTRACT xvi

CAPÍTULO I
INTRODUCCIÓN
1.1. Planteamiento del Problema 1
1.2. Antecedentes 2
1.3. Objetivos 3
1.3.1. Objetivo General 3
1.3.2. Objetivos Específicos 3
1.4. Justificación e Importancia 4

CAPÍTULO II
MARCO TEÓRICO
2.1. Sistemas Embebidos 5
2.2. Estructura y Componentes Básicos 6
2.2.1. Hardware 6
2.2.2. Software 6
2.2.3. Entradas y Salidas 7
2.3. Programación de la NI myRIO-1900 8
2.3.1. NI Labview 8
2.3.2. Módulo NI Labview Control & Design Simulation 11
2.3.3. Módulo NI Labview Real-Time 11
2.3.4. Módulo NI Labview for myRIO 12
2.3.5. Módulo LabVIEW Datalogging and Supervisory Control (DSC) 13
2.4. Controladores Lógicos Programables (PLCs) 14

2.5. Estructura y Componentes Básicos 15
2.5.1. Estructura Externa 15
2.5.2. Estructura Interna 16
2.5.3. Unidad de Procesamiento Central (CPU) 16
2.5.4. Memoria del PLC 16
2.5.5. Entradas 16
2.5.6. Salidas 17
2.6. Programación de los PLCs 17
2.6.1. Siemens Totally Integrated Automation Portal (TIA Portal) 19
2.7. Redes Industriales 20
2.8. Pirámide de Automatización 20
2.9. Niveles de la Pirámide de Automatización 21
2.9.1. Nivel de Campo 22
2.9.2. Nivel de Control 22
2.9.3. Nivel de Supervisión 22
2.9.4. Nivel de Gestión 23
2.10. Comunicación PC – PLC 23

CAPÍTULO III
DESARROLLO DEL SISTEMA EMBEBIDO
3.1. Dinámica de los procesos 24
3.1.1. Dinámica y Función de Transferencia de Caudal 25
3.1.2. Dinámica y Función de Transferencia de Nivel 30
3.1.3. Dinámica y Función de Transferencia de Presión 33
3.2. Configuración inicial de NI myRIO 35
3.3. Programación para la tarjeta NI myRIO. 39

CAPÍTULO IV
CONTROLADOR
4.1. Sistema de Control de Flujo 60
4.1.1. Modelo Matemático 60
4.1.2. Diseño del Controlador PID mediante métodos analíticos 61
4.1. Sistema de Control de Nivel 64
4.2.1. Modelo matemático 64
4.2.2. Diseño del controlador PID mediante métodos analíticos 64
4.3. Sistema de Control de Presión 69
4.3.1. Modelo Matemático 69

4.3.2. Diseño del controlador PID mediante métodos analíticos 69
4.4. Implementación de Control PID 70
4.1.2. Configuración PID 71

CAPÍTULO V
ANÁLISIS Y RESULTADOS
5.1. Validación de los sistemas embebidos 74
5.2. Análisis de los Controladores 79
5.3. Pruebas Experimentales mediante la configuración Hardware in the Loop 82

CAPÍTULO VI
CONCLUSIONES Y RECOMENDACIONES
6.1. Conclusiones 87
6.2. Recomendaciones 88

REFERENCIAS BIBLIOGRÁFICAS 89
ANEXOS 91

ÍNDICE DE FIGURAS

Figura 1. Puertos de Entradas y Salidas myRIO 7
Figura 2. Panel Frontal y Diagrama de Bloques LabView 10
Figura 3. Librería Control Design & Simulation 11
Figura 4. Librería Real-Time 12
Figura 5. Plataforma de LabVIEW para myRIO 13
Figura 6. Librería Datalogging and Supervisory Control (DSC) 13
Figura 7. Beneficios que brinda TIA Portal 19
Figura 8. Pirámide de Automatización 21
Figura 9. Comunicación PC – PLC 23
Figura 10. Lazo de Control del Dispositivo NI myRIO. 25
Figura 11. Diagrama de lazo del Sistema de Control de Caudal 25
Figura 12. Configuración del dispositivo 26
Figura 13. Módulo de entradas y salidas analógicas 27
Figura 14. Configuración del PLC 27
Figura 15. Acondicionamiento de la entrada y la salida 28
Figura 16. Señal de excitación y respuesta de la planta de Caudal 29
Figura 17. Función de transferencia en Matlab de Caudal 30
Figura 18. Diagrama de lazo del proceso de Nivel 31
Figura 19. Señal de excitación y respuesta de la planta de Nivel 32
Figura 20. Parámetros para la función de transferencia de un Sistema Integrante 32
Figura 21. Diagrama de lazo del Sistema de Control de Presión 33
Figura 22. Señal de excitación y respuesta del Sistema de presión 34
Figura 23. Modelo matemático obtenido en Matlab 34
Figura 24. Pantalla Inicial LabVIEW 35
Figura 25. Pantalla Inicial LabVIEW 36
Figura 26. Pantalla myRIO USB Monitor 37
Figura 27. Ventana de Reconocimiento myRIO 37
Figura 28.Ventana para nombrar al dispositivo Fuente: NI myRIO 38
Figura 29. Ventana para nombrar al dispositivo 38
Figura 30. Ventana probar los elementos internos 39
Figura 31. Proyecto para myRIO 40
Figura 32. Creación del proyecto para myRIO 41
Figura 33. Creación del proyecto para myRIO 41
Figura 34. Frames del programa principal del proyecto 42
Figura 35. Flujo de Datos entre Dispositivos 43
Figura 36. Frames del programa principal del proyecto 44
Figura 37. Menú de herramientas 45
Figura 38. Navegador de imágenes 45
Figura 39. Creación de imágenes 46
Figura 40. Creación de imágenes 47
Figura 41. Interfaz Humano Máquina del Sistema de Nivel 48
Figura 42. Interfaz Humano Máquina del Sistema de Caudal 48
Figura 43. Interfaz Humano Máquina del Sistema de Presión 49
Figura 44. Esquema de programación 49
Figura 45. Frame de Inicialización 50
Figura 46. Bloques de control e indicadores principales 50
Figura 47. Bloques de control de válvulas 51
Figura 48. Bloques de interruptores 52

Figura 49. Bloques de control myRIO 52
Figura 50. Bloques de estado de las alarmas.....53
Figura 51. Bloque de impresión de la información 53
Figura 52. Registro de datos. 54
Figura 53. Bloques para imprimir los mensajes de advertencias 55
Figura 54. Mensajes adicionales para la interfaz. 55
Figura 55. Esquema básico con flujo de datos. 56
Figura 56. Toolkit Control & Simulation. 56
Figura 57. Bloque para la intermitencia en las tuberías 57
Figura 58. Bloques de acondicionamientos de la interfaz. 58
Figura 59. Bloque de salidas analógicas de NI myRIO. 58
Figura 60. Frame de finalización. 59
Figura 61. Configuración de control en lazo abierto 61
Figura 62. Configuración de control en lazo cerrado 65
Figura 63. Oscilaciones Sucesivas en Sistema de Control de Nivel 66
Figura 64. Bloque PID Compact 71
Figura 65. Menú bloque PID 72
Figura 66.Normalizar y Escalar entradas y salidas del Sistema 73
Figura 67. Respuesta Sistema Real y Sistema Embebido de Caudal 75
Figura 68. Respuesta Sistema Real y Sistema Embebido de Nivel 75
Figura 69. Respuesta Sistema Real y Sistema Embebido de Presión 76
Figura 70.Función de Transferencia Sistema Embebido y Sistema Real de Caudal 76
Figura 71. Porcentaje de aceptación de las funciones de transferencia del Sistema Real y el Sistema Embebido de Caudal 77
Figura 72. Función de Transferencia Sistema Embebido y Sistema Real de Presión...78
Figura 73.Porcentaje de aceptación de las funciones de transferencia del Sistema Real y el Sistema Embebido de Presión 79
Figura 74. Métodos de Sintonía aplicados al Sistema de Control de Caudal 80
Figura 75. Métodos de Sintonía aplicados al Sistema de Control de Nivel 81
Figura 76. Respuesta del Sistema de Caudal con el Controlador PID 82
Figura 77. Respuesta del Sistema de Nivel con el Controlador PID 83
Figura 78. Respuesta del Sistema de Presión con el Controlador PID 83
Figura 79. Proceso de Presión Subamortiguado 84
Figura 80. Indicadores fundamentales de la Interfaz Humano Máquina 85
Figura 81. Indicadores extra para mostrar el funcionamiento del Sistema 86

ÍNDICE DE TABLAS

Tabla 1 Lanzamiento del software LabView a través de los años 9
Tabla 2. Cálculo de Parámetros del PID en Lazo Abierto mediante Ziegler-Nichols 61
Tabla 3. Cálculo de Parámetros del PID mediante el método Lambda 63
Tabla 4. Cálculo de Parámetros del PID mediante el método de Cohen-Coon 64
Tabla 5. Cálculo de Parámetros del PID mediante Ziegler-Nichols en lazo cerrado 66
Tabla 6. Cálculo de parámetros PID mediante Chien, Hrones y Reswick sin perturbaciones 68
Tabla 7. Cálculo de parámetros PID mediante Chien, Hrones y Reswick con perturbaciones 68
Tabla 8. Parámetros de Ajuste del Controlador mediante la función de transferencia 70
Tabla 9. Comparación de los Métodos de sintonía para el Sistema de Control de Caudal 80
Tabla 10. Comparación de los Métodos de Sintonía para el Sistema de Control de Nivel 81

RESUMEN

El presente trabajo comprende la implementación de un entorno de trabajo bajo la configuración Hardware in-the-Loop (HIL), que consiste en la interacción de componentes reales y virtuales. Los componentes virtuales se encuentra relacionados con la simulación de las estaciones de control a través del software LabVIEW que emula el comportamiento real permitiendo obtener los mismo beneficios que un instrumento físico, siendo programados en la memoria de la tarjeta NI myRIO que es un sistema embebido con tecnología FPGA. La interacción que existe entre el sistema embebido NI myRIO con el PLC Siemens S7-1200 es el que permite trabajar como un entorno Hardware in-the-Loop (HIL). La tarjeta entrega una salida que emula una estación de control del laboratorio, enviando señales eléctricas en un rango de (0 -5) voltios que son enviadas al PLC que contiene un controlador PID provocando una respuesta conocida como Control Value que son devueltas a la tarjeta NI myRIO que se encarga de receptar esta señal y convertirlas en acciones, cumpliendo el ciclo normal como si se tratara de un proceso real. Con la implementación de Hardware-in-the Loop (HIL) se contribuye en el aprendizaje del control automático.

PALABRAS CLAVE:

- **HARDWARE IN THE LOOP**
- **SISTEMAS EMBEBIDOS**
- **CONTROLADOR PROPORCIONAL INTEGRAL Y DERIVATIVO**

ABSTRACT

The present work includes the implementation of a work environment under the Hardware in-the-Loop (HIL) configuration, which consists of the interaction of real and virtual components. The virtual components are related to the simulation of the control stations through the LabVIEW software that emulates the real behavior allowing to obtain the same benefits as a physical instrument, being programmed in the memory of the NI myRIO card, which is an embedded system with FPGA technology. The interaction between the NI myRIO embedded system and the Siemens S7-1200 PLC allows working as a Hardware in-the-Loop (HIL) environment. The card delivers an output that emulates a laboratory control station, sending electrical signals in a range of (0 -5) voltios that are sent to the PLC containing a PID controller causing a response known as Control Value that are returned to the card NI myRIO that is responsible for receiving this signal and converting them into actions, fulfilling the normal cycle as if it were a real process. The implementation of Hardware-in-the-Loop (HIL) contributes to the learning of automatic control.

CAPÍTULO I

1. INTRODUCCIÓN

1.1. Planteamiento del Problema

El control de procesos busca en la industria la corrección de los errores en el comportamiento del sistema, eso se logra mediante la manipulación de los parámetros más importantes, lo cual contribuye inmensamente en la comprensión de dichos procesos y formación de profesionales con capacidades de solventar problemas que se susciten en la industria. Lastimosamente existe una gran limitación para los usuarios al momento de acceder a dicha manipulación y control de variables por diversos factores, como por ejemplo, plantas industriales donde se manejan o se tratan con productos ya sean líquidos o gaseosos muy nocivos para la salud, así como también la complejidad que presenta el proceso tal como un sistema en cascada, o simplemente que tengan una directa relación con los rangos de funcionamiento soportados por los equipos y dispositivos inmersos en el sistema.

Por dichos impedimentos nace la necesidad de implementar un entorno simulado de un proceso complejo real dentro de un sistema embebido. Esta plataforma proporciona a los usuarios la oportunidad de ejercer prácticas de control de procesos mediante la interacción entre hardware y software, aproximándolos a una experiencia industrial, con la gran ventaja de trabajar en un entorno que facilita la implementación de procesos industriales en las áreas de laboratorios de la Universidad sin correr ningún riesgo al manipular las variables de proceso, de esta forma se convierte en una herramienta de enseñanza-aprendizaje muy útil para la formación de estudiantes e incluso para la capacitación del personal de una industria dentro de lo que se refiere al campo de Control de Procesos Industriales. Recalcando que es primordial disponer de sistemas que permitan simular los procesos y probarlos bajo condiciones reales, contribuyendo con la protección del personal y los equipos, todo esto gracias a la emulación de las condiciones de operación de un sistema real en menor tiempo y menor costo en cuanto al hardware utilizado. La simulación de Hardware-in-the-loop (HIL) es una técnica moderna que se utiliza en el desarrollo, prueba y diagnóstico para sistemas de procesos complejos. La simulación HIL proporciona una plataforma

realmente útil tomando en cuenta que adiciona la complejidad de un sistema real a la plataforma de prueba a través de la función de transferencia en condiciones normales del proceso la cual representa el comportamiento de todos los sistemas dinámicos relacionados.

1.2. Antecedentes

La simulación de procesos es una de las más grandes herramientas de la ingeniería industrial, la cual se utiliza para lograr una representación mucho más simple y entendible del proceso en cuestión. Simular consiste en usar un modelo para calcular y proporcionar cierta variable de salida con el fin de producir evoluciones similares ante idénticos estímulos. En una simulación es preciso conocer el valor de las entradas, ya que, lo que se pretende calcular es el valor de las salidas que dependen de ellas. Para realizar la simulación se proporciona un modelo, una situación inicial y el valor de las entradas en cada instante de tiempo posterior. Además se convierte en muchas ocasiones en algo casi indispensable, ya que sin este procedimiento se torna más complicada la comprensión de los procesos (Arahal, Berenguel, & Rodríguez, 2006). La simulación se desarrolló en paralelo con la aparición del computador y que, poco a poco, se ha impuesto gracias a la mayor velocidad y capacidad que los computadores han ido ofreciendo a la industria. El uso del computador en la simulación se inició con el programa PACER en 1966, y su creciente uso en la simulación de procesos industriales ha sido posible gracias al desarrollo de las técnicas numéricas y al empleo de nuevos lenguajes de simulación (D´Arthenay Bermúdez, 2015). Existen en la actualidad diversas técnicas para la simulación de procesos, las cuales con la ayuda de software y hardware, permiten tener una interacción amigable con el usuario además de un preciso control de dichos procesos simulados, aportando así, una gran cantidad de beneficios al usuario siendo los más destacables el evitar peligrosos daños en los equipos de procesos reales e incluso sobre la vida de los mismos usuarios. La simulación de Hardware-in-the-loop (HIL) es una técnica que se utiliza en el desarrollo y prueba de sistemas de procesos complejos. La simulación HIL proporciona una plataforma efectiva al agregar, la complejidad del control de la planta a la plataforma de prueba y desarrollo mediante una representación matemática de todos los sistemas dinámicos relacionados. Estas representaciones matemáticas se conocen como la "simulación de plantas" (Luque

Navarro, 2017). Hardware-In-the-Loop es una forma de simulación en tiempo real, la cual difiere al agregar un componente real en el ciclo. Este componente puede ser una Unidad de Control Electrónico (ECU). El propósito de un sistema HIL es proporcionar todos los estímulos eléctricos necesarios para ejercitar completamente la ECU. De esta forma, se "engaña a la ECU para que piense que efectivamente está conectada a una planta real. La simulación HIL incluye un modelo matemático del proceso y un dispositivo de hardware donde se puede simular un controlador industrial. El dispositivo de hardware es normalmente un sistema integrado. La aplicación de la tecnología HIL, se utiliza principalmente en Dispositivos Médicos, Máquinas Industriales, Sistemas de Generación de Energía, Industria aeroespacial, procesos de control, entre otros (Arahal, Berenguel, & Rodríguez, 2006).

1.3. Objetivos

1.3.1. Objetivo General

- Desarrollar HIL (HARDWARE-IN-THE-LOOP) para el control de los procesos presión, caudal y nivel, mediante el sistema embebido myRIO a realizarse en el laboratorio de redes industriales y control de procesos

1.3.2. Objetivos Específicos

- Investigar los métodos para obtener los modelos matemáticos que describen el comportamiento dinámico de los procesos.
- Obtener los modelos matemáticos que describen el comportamiento dinámico de los procesos presión, caudal y nivel.
- Programar la tarjeta NI myRIO para la creación de los procesos simulados presión, caudal y nivel.
- Desarrollar el entorno de simulación de los procesos presión, caudal y nivel para el control con el PLC en tiempo real.
- Verificar el correcto funcionamiento del simulador HIL mediante técnicas de control convencional.

1.4. Justificación e Importancia

Dentro del método enseñanza-aprendizaje, el complemento de la parte teórica es la parte práctica que se vuelve indispensable para una experiencia completa dentro de lo que se refiere a la formación de profesionales en las universidades para ejercer Control de Procesos. El enfoque de la seguridad industrial demanda que los profesionales cuenten con la mayor cantidad de conocimientos posibles para que se desenvuelvan en el ámbito laboral como es debido, evitando así correr riesgos innecesarios que conlleven a causar grandes pérdidas tanto en lo que se refiere al factor humano como en lo económico.

Un laboratorio universitario cuenta con varias estaciones de control, los cuales cumplen con la tarea de generar experticia en los estudiantes acerca de la manipulación y control de los distintos elementos que conforman un sistema industrial. Si bien aportan en el desarrollo integral de las habilidades de los operarios, en la industria la mayoría de procesos son procesos más complejos por tal razón, el conocimiento y entrenamiento o ensayo con este tipo de procesos es vital, porque realizarlo de manera física se vuelve muy complicado.

La aplicación de este tipo de plataforma, permite obtener los conocimientos necesarios para la manipulación de los parámetros de un proceso industrial que se aproximen de mejor manera al mundo real y laboral, de igual manera garantiza que el personal y los equipos no se encuentren en riesgo alguno y se desarrolle el aprendizaje de manera adecuada.

CAPÍTULO II

2. MARCO TEÓRICO

2.1. Sistemas Embebidos

Un sistema embebido es un circuito electrónico capaz de contener a todo un equipo conformado por diversos componentes y elementos como por ejemplo válvulas, motores, niquelinas, bombas, entre otros, dentro de un dispositivo más compacto pero de recursos limitados lo que lo diferencia de un computador personal, aun así al tener la capacidad de cumplir una función específica son muy útiles en diversas aplicaciones ya que muchos aparatos electrónicos funcionan en base a sistemas embebidos, como en el campo de las telecomunicaciones (módems, enrutadores, teléfonos móviles), en el hogar (microondas, refrigeradoras), incluso en la industria automotriz (navegador GPS), además gestiona los datos del sistema en tiempo real. Por sus características permite realizar aplicaciones ejecutándolas sin conexión a un ordenador remoto.

La interacción de los sistemas embebidos con las variables externas es el tema más 3importante a considerar, sean estímulos de entrada o de salida del sistema. Es por eso que algunas características en las que se debe hacer énfasis para lograr tener una interacción exitosa con el ambiente son la disponibilidad, fiabilidad y seguridad. Otras características propias de un sistema embebido son (Lavi & Kudish, 2004):

- El software que compone a estos sistemas se encarga de controlar y comunicar los elementos hardware hacia el interior y exterior del sistema embebido.
- Funciona jerárquicamente dependiendo del estado de sus componentes, por ejemplo, si un puerto no se encuentra disponible, no se podrá enviar la señal que activa el proceso que activaba dicho puerto.
- La constatación del cambio de estado de los componentes se dan de acuerdo al manejo de eventos, ya sean estos de tipo externo precisamente de acciones ajenas al sistema o interno directamente de software.
- En un sistema embebido se tienen recursos limitados esto gracias a que se debe optimizar factores como el tamaño, la energía entre otros que permitan garantizar la portabilidad del sistema embebido.

- Son sistemas que poseen un alto grado de autonomía. Al ser básicamente “cajas negras” los dispositivos que manejan sistemas embebidos son muy bien realizados ya que se diseñan para trabajar por mucho tiempo y con el mínimo de fallos.
- Los sistemas embebidos poseen dos grandes características, la sincronización y comunicación que son aquellas que permiten el flujo de la información.

2.2. Estructura y Componentes Básicos

Los Sistemas Embebidos se conforman por hardware y software, que se complementan para lograr cumplir la función para la cual se diseñan.

2.2.1. Hardware

Dentro del campo de la educación existen distintos dispositivos que son capaces de operar sistemas embebidos uno de los más conocidos es NI myRIO, el cual es un dispositivo embebido para estudiantes, que incluye en su parte física entradas y salidas analógicas, líneas de entradas y salidas digitales, LED’s indicadores, un acelerómetro interno, un FPGA Xilinx, un push-button y al igual que un computador, un dispositivo embebido básicamente se compone también de una unidad de procesamiento central que consiste en un microprocesador, microcontrolador o un procesador digital de señales en este caso su procesador es un dual-core ARM Cortex-A9. Además posee una memoria RAM que es destinada para almacenar el programa que gobierne una determinada aplicación. Los actuadores y elementos electrónicos que el sistema se encarga de controlar generalmente se encuentran montados directamente sobre la placa base (Instruments National, 2016).

2.2.2. Software

Cuando se tiene la necesidad de programar una FPGA generalmente surgen requerimientos y capacidades que debería poseer el programador como lo son un conocimiento detallado de la configuración específica del hardware y la utilización de un lenguaje de descripción de bajo nivel como VHDL. Esto complica un poco el panorama para quien no maneja este tipo de lenguaje de programación, así es como la tecnología NI RIO se ha centrado en facilitar el acceso a estos dispositivos embebidos por parte de un software más interactivo y amigable, el mismo que

asociado al sistema físico (tarjeta, bus de instrumentación, comunicación serie, etc.) se compone de tres niveles de decisión básicos, sistema operativo, software a nivel driver y software de aplicación (Hernández Cevallos & Ledesma Marcalla, 2010).

2.2.3. Entradas y Salidas

Proporciona entradas y salidas reconfigurables las mismas que se encuentran ubicadas en ambos lados del dispositivo con dos formas distintas de conectores MXP dispuestos de acuerdo a la Figura 1.

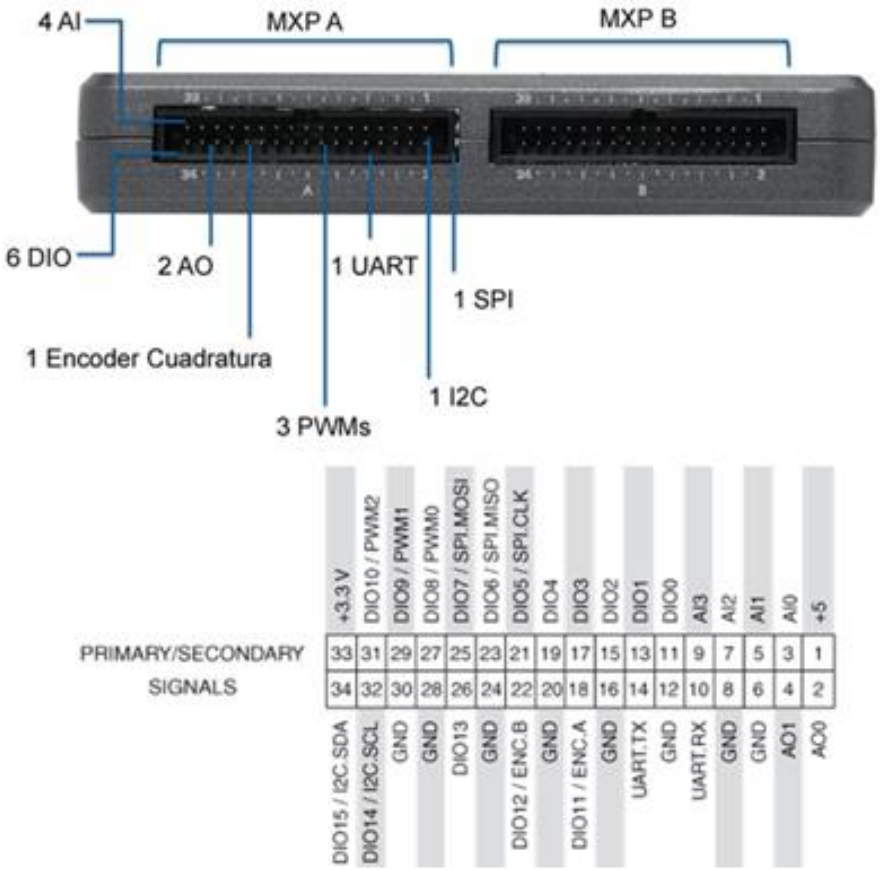

Figura 1. Puertos de Entradas y Salidas myRIO

Fuente: *(Instruments National, 2016)*

Dichas entradas y salidas permiten ingresar señales al sistema desde dispositivos externos como pulsadores, interruptores, sensores que permitan realizar una acción dentro del sistema embebido y extraerlas para lograr tener un comportamiento visible mediante las salidas del dispositivo ya sean estas de tipo análoga o digital. Cuarenta líneas de Entradas y Salidas digitales, ocho entradas analógicas en una sola terminal, cuatro salidas analógicas unipolares, dos entradas analógicas diferenciales y dos

salidas analógicas con referencia a tierra además de una entrada de codificador de cuadratura, PWM, UART e I2C. (Instruments National, 2016).

2.3. Programación de la NI myRIO-1900

National Instruments mediante el software de desarrollo gráfico LabVIEW y sus diferentes módulos permite acceder y configurar el dispositivo NI myRIO integrando los tres niveles de decisión básicos, permitiendo de esta manera realizar varias actividades como diseñar, generar prototipos e implementar aplicaciones embebidas en un solo entorno pero sin perder eficiencia (Hernández Cevallos & Ledesma Marcalla, 2010).

2.3.1. NI Labview

Laboratory Virtual Instrument Engineering Workbench (LabVIEW) es un lenguaje y al mismo tiempo un entorno de programación gráfica en el que se pueden crear aplicaciones de una forma rápida y sencilla. La empresa desarrolladora y propietaria de LabVIEW es National Instruments, se fundó en Austin, Texas en el año de 1976. El desarrollo de LabVIEW que posteriormente sería su producto estrella empezó en abril de 1983, y fue lanzado al mercado exclusivamente para Macintosh en octubre de 1986 con la versión de LabVIEW 1.0. A pesar de que en 1990 salió a la luz la versión 2.0 aún no contaba con soporte para para Windows y se tuvo que esperar a septiembre de 1992 para por fin lograr manejar el software en esta plataforma. Los principales hitos de LabVIEW pueden verse en la Tabla 1 (Lajara Vizcaino & Pelegrí Sebastiá, 2007).

Tabla 1

Lanzamiento del software LabView a través de los años

Fecha	Hito
Abril de 1983	Empieza el desarrollo del software
Octubre de 1986	LabVIEW 1.0 para Macintosh
Enero de 1990	LabVIEW 2.0
Septiembre de 1992	LabVIEW para Windows
Octubre de 1992	LabVIEW para Sun
Octubre de 1993	LabVIEW 3.0 multiplataforma
Abril de 1994	LabVIEW para Windows NT
Octubre de 1994	LabVIEW para Power Macintosh
Octubre de 1995	LabVIEW para Windows 95
Mayo de 1997	LabVIEW 4.0
Marzo de 1998	LabVIEW 5.0
Febrero de 1999	LabVIEW 5.1, LV para Linux y LV Real-Time
Agosto de 2000	LabVIEW 6i
Enero de 2002	LabVIEW 6.1
Mayo de 2003	LabVIEW 7 Express, LabVIEW PDA y FPGA
Mayo de 2004	LabVIEW 7.1
Mayo de 2005	LabVIEW DSP
Junio de 2005	LabVIEW Embedded
Octubre de 2005	LabVIEW 8
Agosto de 2006	LabVIEW 8.20
Agosto de 2007	LabVIEW 8.5
Agosto de 2008	LabVIEW 8.6
Agosto de 2009	LabVIEW 2009
Agosto de 2010	LabVIEW 2010
Agosto de 2011	LabVIEW 2011
Agosto de 2012	LabVIEW 2012
Agosto de 2013	LabVIEW 2013, LabVIEW myRIO
Agosto de 2014	LabVIEW 2014
Agosto de 2015	LabVIEW 2015
Agosto de 2016	LabVIEW 2016
Agosto de 2017	LabVIEW 2017

Fuente: (Lajara Vizcaino & Pelegrí Sebastiá, 2007)

LabVIEW se enfoca en el desarrollo de sistemas de instrumentación, y en este caso al ser gráfico se llama instrumentación virtual por lo que la extensión de sus archivos es VI que se refiere a Virtual Instrument. El entorno del software se centra en dos ventanas principales el primero llamado Panel Frontal donde se mostrará todo el entorno que permitirá interactuar al usuario mientras que la segunda ventana se denomina Diagrama de bloques, aquí se va a realizar toda la programación y es como la circuitería interna de lo que se visualiza en el Panel Frontal. Cabe recalcar que estas dos ventanas se encuentran interconectadas entre sí, es decir si se crea un indicador en el diagrama de bloques este aparecerá en el Panel Frontal y viceversa. En la Figura 2 se muestran el Panel Frontal y el Diagrama de Bloques donde se visualiza que los elementos creados en una ventana aparecen también en la otra. (Lajara Vizcaino & Pelegrí Sebastiá, 2007).

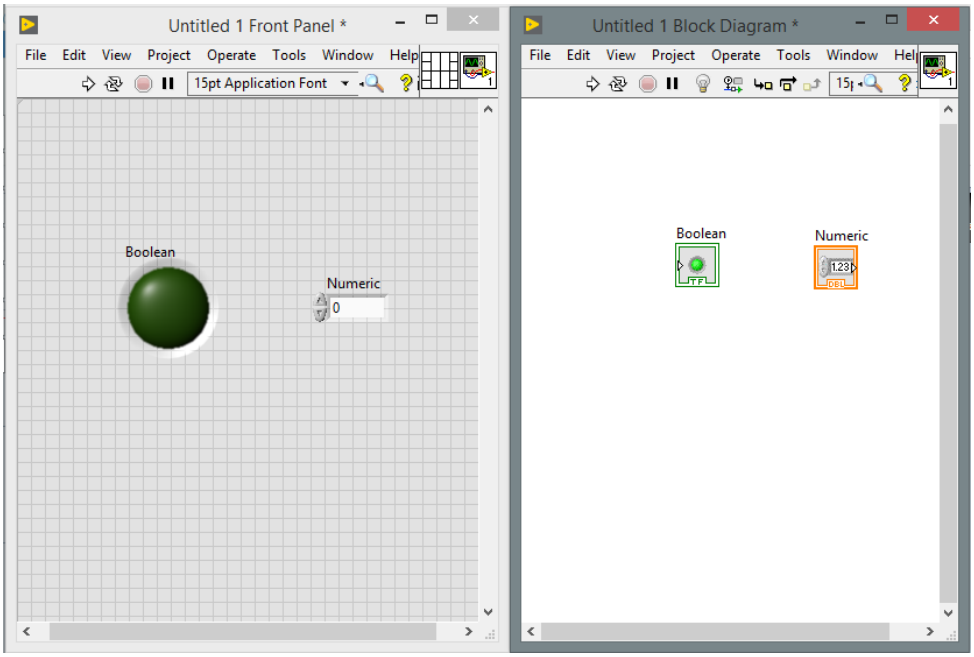

Figura 2. Panel Frontal y Diagrama de Bloques LabView

Fuente: Labview 2012

En el campo de Simulación y Control, LabVIEW tiene varias prestaciones que proporcionan muchas facilidades para cumplir con el objetivo trazado como son los módulos (*Toolkits*) que realizan funciones adicionales a las básicas. Algunos de estos son "LabVIEW Control & Design Simulation", "LabVIEW Real-Time","LabVIEW Datalogging and Supervisory Control".

2.3.2. Módulo NI Labview Control & Design Simulation

El módulo “Control & Design Simulation” permite mediante la función de transferencia, la ganancia del polo cero e incluso el espacio de estados obtenido de un proceso físico, construir una planta de proceso de una forma simulada. Gracias a esto es posible llegar a simular sistemas lineales, no lineales y hasta sistemas discretos de una forma mucho más sencilla. Dentro del módulo “Control & Design Simulation” (Figura 3) se tienen las librerías Simulation, Control Design, System Identification, PID, Fuzzy Logic y Sim Interface, que aportan grandes beneficios cuando existe la necesidad de realizar aplicaciones más complejas como analizar el comportamiento del modelo de lazo abierto, simular sistemas en línea y fuera de línea, diseñar controladores de circuito cerrado y realizar implementaciones físicas.

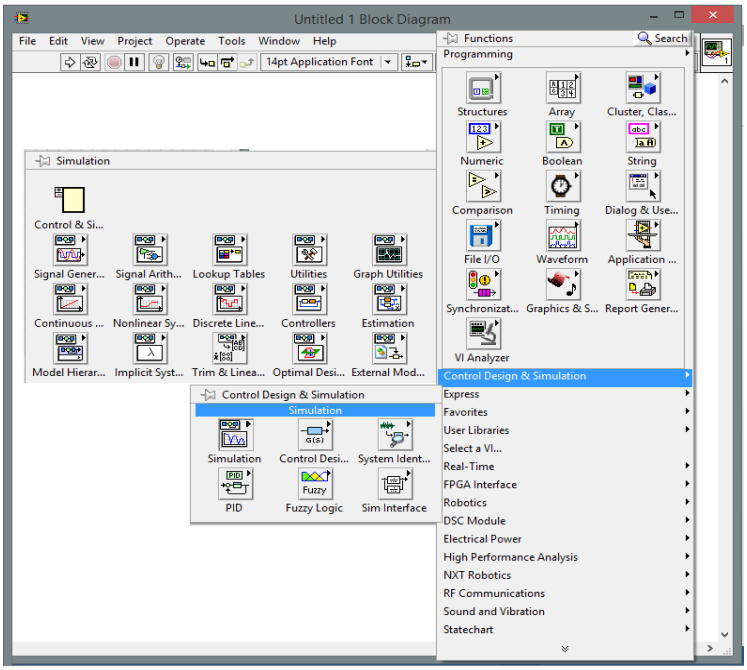

Figura 3. Librería Control Design & Simulation

Fuente: Labview 2012

2.3.3. Módulo NI Labview Real-Time

Dentro de LabVIEW existen aplicaciones que demandan un comportamiento determinístico en tiempo real que los sistemas operativos donde se ejecuta el software LabVIEW como Windows,Mac Os o Linux no pueden garantizarlo, es por

este motivo que surge la problemática de tener un sistema, en este caso un módulo que permita satisfacer esta necesidad. Así aparece LabVIEW Real-Time (Figura 4), que es un módulo que combina la programación gráfica de LabVIEW con el poder de un sistema operativo en tiempo real. Para desarrollar una aplicación en LabVIEW en tiempo real se realiza la programación deseada pero incluyendo los elementos de la librería del módulo Real-Time, aportando una forma sencilla para construir aplicaciones determinísticas en tiempo real dentro de una plataforma estable para las mismas (National Instruments, 2004).

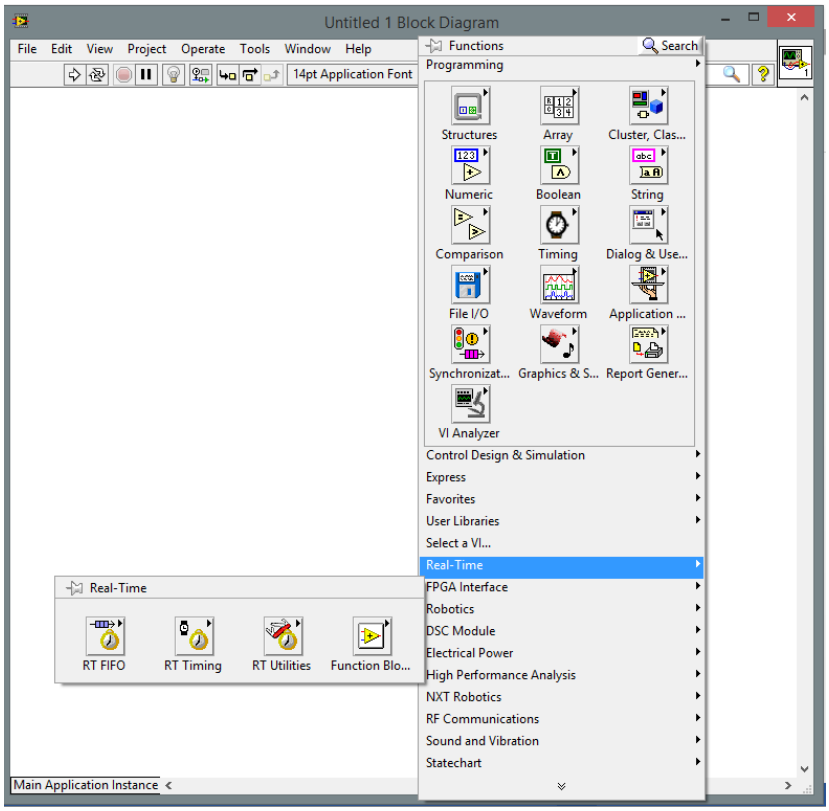

Figura 4. Librería Real-Time

Fuente: Labview 2012

2.3.4. Módulo NI Labview for myRIO

El módulo “LabVIEW for my RIO”, no solo proviene las librerías del Toolkit sino que además proporciona una plataforma específica del dispositivo embebido myRIO de LabVIEW que permite crear proyectos específicos (Figura 5) donde posteriormente se realiza la programación y compilación del VI, este módulo provee herramientas que

permiten acceder a características específicas del dispositivo embebido enfocándose en el desarrollo de sus aplicaciones.

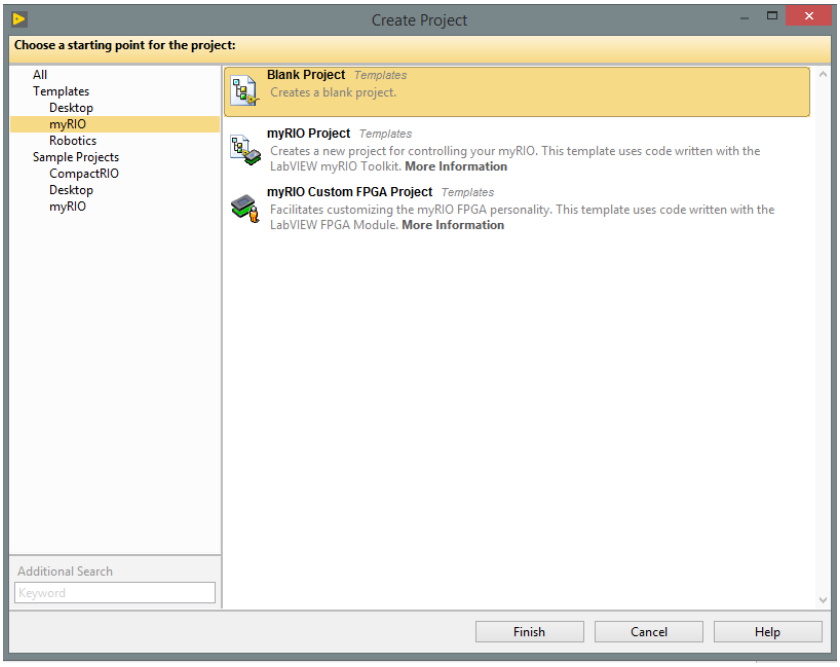

Figura 5. Plataforma de LabVIEW para myRIO

Fuente: Labview 2012

2.3.5. Módulo LabVIEW Datalogging and Supervisory Control (DSC)

Este módulo (Figura 6) amplía el desarrollo gráfico de LabVIEW con algunas funcionalidades adicionales para el desarrollo rápido de entornos que se centran en la medición, control y monitoreo de variables.

Mediante este módulo se puede registrar datos automáticamente; añadir alarmas, seguridad y básicamente realizar el sistema de Supervisión, Control y Adquisición de Datos (SCADA).

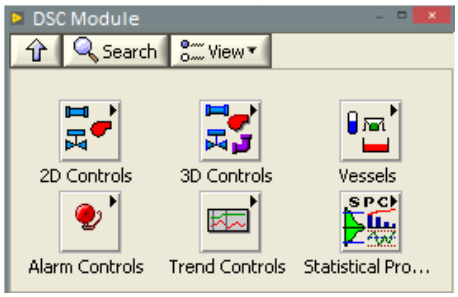

Figura 6. Librería Datalogging and Supervisory Control (DSC)

Fuente: Labview 2012

2.4. Controladores Lógicos Programables (PLCs)

Un controlador lógico programable es un tipo de “caja negra” y puede ser definido como una máquina electrónica la cual está diseñada para controlar en tiempo real procesos industriales secuenciales de control, utiliza una memoria programable que cumple funciones específicas de distintos tipos como lógicas, secuenciales, temporización, conteo y aritméticas, se comunican con el exterior para controlar procesos mediante módulos de entrada/salida los cuales pueden ser de tipo digitales y analógicos. Su inicio empezó en la industria automotriz donde se buscaba reducir costos en los sistemas de control, por lo cual en 1968 la empresa General Motors se planteó la idea de tener un dispositivo con especificaciones suficientes para lograr realizar las acciones de control en la industria de una manera optimizada (Vallejo, 2005).

Las cualidades que debería reunir el dispositivo eran las siguientes:

- El dispositivo de control debe ser fácil y rápidamente programable por el usuario.
- Debe ser capaz de controlar los dispositivos de las sin necesidad de más equipo de soporte.
- Debía ser bastante interactivo con el usuario capaz de mostrar el status del dispositivo para lograr informar sobre errores y mantenimiento, de esta manera haciéndolo más fácil para el usuario.
- El sistema debía ser lo más compacto posible y tener menor consumo de energía comparado al sistema con relevadores.
- Tenía que ser capaz de comunicarse con un sistema central de datos para propósitos de monitoreo.
- Debían tener la capacidad de controlar a los actuadores de los procesos industriales incluso los que funcionen con corriente alterna.
- Debía ser mucho mejor en todos los sentidos al ser comparado con los sistemas anteriores.

Estas cualidades que la empresa definió, llamó la atención de diversas industrias de desarrollo eléctrico que se interesaron en esta problemática y dieron vida a los primeros dispositivos autómatas programables. El concepto de la idea inicial se cambió un poco durante la fabricación de los PLCs para facilitar su instalación, pero

igualmente estos dispositivos al ser los primeros equipos en reemplazar a los relevadores, tenían algunas complicaciones en lo que se refiere a costos, flexibilidad, mantenimiento y capacidad ya que solo podían realizar un control ON/OFF de los procesos lo que hizo que en la siguiente década con el avance de la electrónica concretamente de los microprocesadores se luche por mejorar cada debilidad del dispositivo. Con las mejoras posteriores se agregaron características como inteligencia adicional, que brindaron grandes beneficios en la capacidad de interfaz con el operador, ampliación de datos, uso de términos de video, desarrollo de programas, entre otros. Llegando así a convertir a los PLCs en Dispositivos Electrónicos versátiles y flexibles (Álvarez Pulido, 2004). En la actualidad estos dispositivos son mucho más robustos y capaces de trabajar en ambientes muy agresivos ofreciendo varias ventajas.

Una ventaja importante a nivel de usuario es que la programación y manejo de un PLC es muy sencilla por lo que pueden ser realizados por personal que posea pocos o nulos conocimientos de informática, nada más basta que tenga conocimientos eléctricos. Estos dispositivos pueden cumplir funciones básicas como AND, NOT, OR, NAND, NOR, XOR y funciones especiales como temporizadores o contadores (Vallejo, 2005).

2.5. Estructura y Componentes Básicos

Los Controladores lógicos programables generalmente tienen una estructura compacta donde se encuentran todos sus componentes, aunque dentro de lo que se refiere a hardware, se hablan de dos diferentes estructuras modulares de los PLCs.

2.5.1. Estructura Externa

Un PLC puede ser de forma compacta que quiere decir que todos los elementos se encuentran en un solo bloque y de forma modular donde se divide la clasificación de una manera continental.

La forma americana que simplemente separa las entradas y salidas del módulo que contiene la unidad de procesamiento, y la europea donde cada unidad tales como las entradas y salidas, fuente de alimentación, memoria, las toma como un módulo independiente (Vallejo, 2005).

2.5.2. Estructura Interna

Un PLC internamente se constituye principalmente de tres componentes; la Unidad de Procesamiento Central, Memoria, puertos de entrada y salida de datos.

2.5.3. Unidad de Procesamiento Central (CPU)

Es la parte más importante del sistema ya que cumple varias funciones, como leer y ejecutar el programa realizado por el usuario además de ordenar y organizar la comunicación entre las distintas partes del PLC (Villajulca, 2009).

2.5.4. Memoria del PLC

La memoria de sistema que es de tipo EPROM es un lugar físico en donde se encuentra la BIOS del PLC que es un pequeño programa de diagnóstico desarrollado por el fabricante, la cual permite al procesador saber si hay periféricos conectados al sistema o incluso si hay algún programa grabado en la memoria permitiendo arrancar las operaciones del dispositivo. La memoria del PLC es de tipo RAM y se encarga solamente de almacenar el programa realizado por el usuario y descargado en el dispositivo.

2.5.5. Entradas

Son circuitos electrónicos internos que permiten enviar estas señales a una bornera, que es la que permite mediante un medio cableado, ingresar las señales desde el exterior y estas entradas pueden ser de dos tipos, digitales y analógicas. Las entradas digitales, son aquellas que reciben simplemente señales codificadas de tipo booleano es decir, que varía entre dos valores 0 que indica el valor lógico menor y 1 indica el valor lógico mayor, dichas señales son generalmente de tipo ON/OFF y provienen de elementos como interruptores. Las entradas analógicas son aquellas capaces de recibir valores de tipo real las cuales generalmente son mediciones de una variable física. Estas señales ingresan al dispositivo en una forma de voltaje generalmente de 0 a 10 voltios o de corriente de 4 a 20 miliamperios.

El procesador de un PLC manipula señales de tipo digital por lo cual una señal analógica ingresa al dispositivo e internamente es transformada lo cual debe tener una gran resolución para evitar perder valores de la señal de entrada (Bastidas Miño &

Proaño Cevallos, 2010). Las señales de entrada se asignan en un espacio de memoria definido por el fabricante del dispositivo.

2.5.6. Salidas

Al igual que las entradas son circuitos electrónicos pero la diferencia es que estas realizan un acople con las señales transformadas a digital de las entradas y generan una relación coherente que sea capaz de manipular valores y permitan controlar elementos externos como actuadores. Así mismo las salidas son de tipo analógicas y digitales.

Las salidas digitales son de voltaje o de relé, las de voltaje asigna un valor mayor a un 1 lógico y un valor menor al 0 lógico, mientras que la de tipo relé solo asigna como un switch donde habilita la salida con 1 lógico y deshabilita con 0 lógico. Las salidas analógicas pueden ser de voltaje generalmente de 0 a 10 voltios o de corriente de 4 a 20 miliamperios (Bastidas Miño & Proaño Cevallos, 2010). Al igual que las entradas las señales de salida se asignan a un espacio de memoria definido por el fabricante.

2.6. Programación de los PLCs

Para la programación de un PLC se debe emplear un lenguaje específico que sea entendido por el dispositivo. Dependiendo del fabricante, la programación de los PLCs varía ya sea en la forma en que el programa se crea, se almacena o simplemente la manera en la que el dispositivo interpreta las instrucciones.

El lenguaje de programación se encuentra estandarizado internacionalmente desde el año de 1992 y su estándar es el IEC 1131-3, esto da paso a que se dispongan de varios lenguajes y entornos de programación (Aguilera Martínez, 2002):

- Diagrama de Funciones Secuenciales (SFC)
- Diagrama de Bloques de Funciones (FBD)
- Diagramas de Tipo Escalera (LAD)
- Texto Estructurado (ST)
- Lista de instrucciones (IL o STL)

El Diagrama de Funciones Secuenciales (SFC) inicialmente no fue considerado como un lenguaje de programación sino más bien se lo tomaba como un diagrama para representar el funcionamiento de un sistema y posteriormente gracias a distintos

fabricantes de estos autómatas se logra establecer como un potente lenguaje de programación de tipo gráfico, que representa los sucesivos comportamientos de un sistema lógico, contemplando entradas, acciones a realizar y los procesos intermedios que provocan estas acciones permitiendo así realizar un modelo de un proceso a automatizar (Sofrony, 2009). Lastimosamente no todos los autómatas permiten su programación por este tipo de lenguaje lo que lo limita mucho en relación a otros.

El Diagrama de Bloques de Funciones (FBD) es también un lenguaje de programación de tipo gráfico con la peculiaridad de que este es considerado de alto nivel y permite al usuario programar rápidamente resumiendo muchas de las funciones básicas, tanto expresiones como en lógica booleana. Este lenguaje es muy útil cuando los usuarios son principiantes y no poseen un gran conocimiento de programación. Actualmente es un lenguaje muy utilizado en aplicaciones que implican flujo de información o datos entre componentes de control (Florencio Inga, 2012).

El Diagrama de Tipo Escalera (LAD) es un lenguaje de tipo gráfico que es soportado por casi todos los autómatas programables. Este lenguaje trabaja generalmente con señales booleanas y prácticamente no se utiliza para trabajar con variables analógicas. Utiliza entre sus herramientas de programación barras de alimentación y elementos de enlace y estados, contactos, bobinas y bloques funcionales; El lenguaje tipo escalera es uno de los más utilizados en el campo de la programación de PLCs esto gracias a sus características de simplicidad y compatibilidad (Florencio Inga, 2012).

El lenguaje mediante Texto Estructurado (ST) como su nombre lo dice es de tipo texto de alto nivel y es muy similar a los lenguajes PASCAL, BASIC y C. Aunque dispone de varias ventajas en cuanto se refiere a la construcción del control ya que se pueden manejar de mejor manera los ciclos de repetición mediante instrucciones, son muy poco utilizados puesto que se necesita un gran conocimiento en programación (Florencio Inga, 2012).

Una Lista de instrucciones (IL o STL) es un lenguaje de programación tipo texto más básico y de bajo nivel similar al lenguaje ensamblador por lo que es utilizado para programas de una reducida extensión, la característica más sobresaliente es que trabaja con un registro especial que se denomina acumulador (LD,ST) (Florencio Inga, 2012).

Estos lenguajes de programación necesitan un entorno que permita al usuario comunicarse con el Controlador Lógico Programable y grabar el programa realizado,

por lo cual cada fabricante dispone de una plataforma específica para sus dispositivos como por ejemplo Allen Bradley con RSLogix, y Siemens con el STEP7 MICRO/WIN o STEP7 TIA PORTAL.

2.6.1. Siemens Totally Integrated Automation Portal (TIA Portal)

Para la ingeniería de un sistema de automatización se ha popularizado varias herramientas de configuración. Actualmente la ingeniería ofrece un nuevo concepto, Totally Integrated Automation (TIA Portal) que busca en un futuro ser la base de todos los sistemas de ingeniería, manejando todos los componentes de automatización mediante una plataforma que permita configurar de forma intuitiva y eficiente todos los procesos de planificación, producción, programación y diagnóstico de los controladores y elementos utilizados en automatización Figura 7 (Berger, 2014).

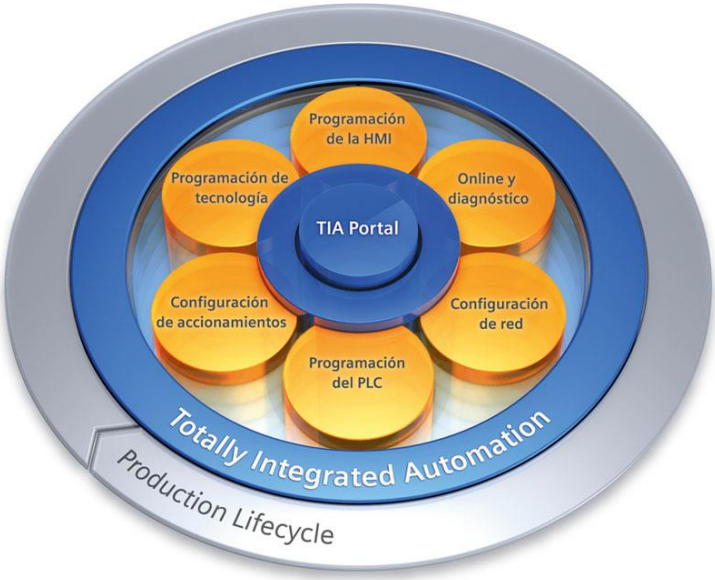

Figura 7. *Beneficios que brinda TIA Portal*

Fuente: (Siemens, 2011)

La gran ventaja de TIA Portal es que posee una interfaz de usuario muy intuitiva con una clara sencillez de sus funciones y la total transparencia de datos permite al usuario optimizar todos los procedimientos de procesamiento, operación de máquinas y planificación.

2.7. Redes Industriales

Un concepto más general define a una red como una serie de equipos o estaciones a los que se los llama “Nodos” que se encuentran interconectados entre sí con el objetivo de lograr tener datos o información disponible en uno, varios o cada uno de los dispositivos que conforman la red. Cuando estas redes se encuentran conformadas por equipos de control a nivel industrial como Computadores Industriales, Controladores, Sensores, Actuadores, Interfaces de operador toman el nombre de Redes Industriales (Luna, Muriel, Pantoja , Giraldo, & Arrechea, 2017).

Hace algunos años se tornaba inimaginable poseer funciones tales como el control continuo y discreto, arquitecturas redundantes o supervisión de los procesos mediante interfaces que permitan al usuario interactuar directamente con los elementos de un proceso industrial, pero en la actualidad con la evolución a pasos agigantados de la tecnológica, junto con la fiabilidad del hardware y software se han integrado los sistemas de automatización y control que permiten administrar las Redes Industriales de una manera más ordenada, propiciando ahorro de materia prima, mano de obra y energía, además aumenta la productividad reduciendo considerablemente los periodos de inactividad y brinda mayor seguridad operativa. La automatización permite a las Redes Industriales tener la capacidad de intercomunicación entre sus dispositivos y el uso de mecanismos estandarizados, abiertos y sobretodo transparentes.

2.8. Pirámide de Automatización

Dentro del campo de la automatización y control existen niveles establecidos de forma jerárquica, la pirámide de automatización (Figura 8) muestra cada uno de ellos.

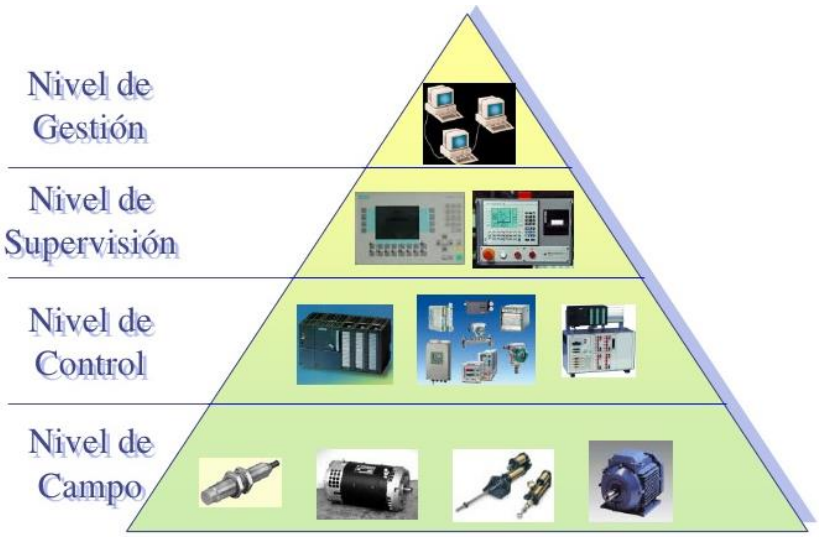

Figura 8. Pirámide de Automatización

Fuente: (Velásquez, 2012)

La Pirámide de Automatización se compone básicamente de cuatro niveles en los que se encuentran dispuestos de acuerdo a la función que realizan cada uno de los elementos que conforman un proceso y que finalmente serán comunicados entre sí.

2.9. Niveles de la Pirámide de Automatización

Los Niveles de la Pirámide de Automatización permiten gestionar de una manera organizada lo que se refiere a Automatización y Control de procesos y son los siguientes:

- Nivel de Campo
- Nivel de Control
- Nivel de Supervisión
- Nivel de Gestión

Estos niveles se vinculan entre sí por medio de los protocolos de comunicación, que pueden ser por ejemplo Profibus DP, Devicenet, AS-i, Ethernet/IP, Modbus TCP/IP, estos protocolos se utilizan de acuerdo a factores como la cantidad de datos a ser enviados. Con esta organización estructural se pueden controlar y monitorear desde pequeños hasta grandes procesos con la gran ventaja de reducir los tiempos de mantenimiento y optimizar la productividad.

2.9.1. Nivel de Campo

Es el primer nivel de la pirámide donde se encuentran elementos que están ubicados directamente en el proceso, capaces de manejar las variables físicas y transformarlas en eléctricas como sensores, actuadores lineales o rotativos, válvula de proceso entre otros (Pacheco Rubio & Palencia Cadena, 2008).

2.9.2. Nivel de Control

Este es el segundo nivel donde se encuentran situados están los distintos dispositivos que permiten realizar monitoreo del sistema y controlar los elementos de campo, aquí se sitúan los PLCs que son los encargados de procesar los datos y las señales que proporcionan los sensores y emite una señal para realizar acciones correctivas a los actuadores. En este nivel también se ubican subsistemas con otro tipo de arquitectura como los sistemas de control distribuido (DCS) o las Unidades de terminales remotos (RTU) (Luna, Muriel, Pantoja , Giraldo, & Arrechea, 2017).

2.9.3. Nivel de Supervisión

Este nivel como su nombre lo indica se encarga de supervisar a cada uno de los elementos de los niveles inferiores que conforman el proceso. Esto lo realiza mediante herramientas muy útiles que permiten diseñar interfaces muy interactivas para el usuario que permita un mejor entendimiento del comportamiento del sistema. Aquí se encuentran las Interfaces Humano Máquina (HMI) que brindan la posibilidad de realizar la supervisión a un nivel local, pero también se cuenta con los sistemas de Supervisión, Control y Adquisición de Datos (SCADA) los cuales permiten al usuario tener una visión más amplia o centralizada, recogiendo la información de distintas fuentes para después mostrarla en una o varias interfaces gráficas. Estos sistemas son manejados en equipos como computadores, por lo que se torna indispensable que estos equipos sean de nivel industrial con grandes capacidades (Pacheco Rubio & Palencia Cadena, 2008) (MICRO Automación, 2000).

2.9.4. Nivel de Gestión

Este es el nivel más alto de la pirámide aquí se puede gestionar todos los datos del proceso brindando la posibilidad de tener dichos datos localmente o enviando a otros dispositivos externos generando una Red. En este nivel se cuenta con equipos de grandes características que eviten la pérdida de conexión y posteriormente de datos (MICRO Automación, 2000).

2.10. Comunicación PC – PLC

La comunicación PC-PLC (Figura 9) se realiza mediante una red punto a punto mediante diferentes protocolos de comunicación ya sean estos RS-232, RS-485, Ethernet, dependiendo de factores como la distancia y fiabilidad que proporcione dicho protocolo o simplemente de la decisión de la disponibilidad de conexión del PLC por parte del fabricante. Las variaciones de los valores que entran o salen del PLC se pueden visualizar a través del software propio de programación, pero para que sea más amigable con el usuario se realizan las Interfaces Humano Máquina que deben comunicarse mediante un servidor OPC que permita crear enlace (Bastidas Miño & Proaño Cevallos, 2010).

Figura 9. Comunicación PC – PLC

CAPÍTULO III

3. DESARROLLO DEL SISTEMA EMBEBIDO

En este capítulo se realiza una explicación sobre la conformación de las plantas simuladas dentro del sistema embebido partiendo desde el punto más básico que viene a ser la obtención de la función de transferencia a través de la utilización del PLC Siemens S7-1200 en conjunto con la aplicación "Ident" de Matlab, construcción de la Interfaz Humano Máquina (HMI) con figuras bidimensionales y tridimensionales utilizadas en el área profesional volviendo intuitivos a los sistemas desarrollados para Hardware in the Loop (HIL) y finalizando con la programación interna de la tarjeta NI myRIO-1900.

3.1. Dinámica de los procesos

En los procesos industriales las variables que comúnmente son controladas son Nivel, Presión, Caudal y Temperatura. De las plantas de proceso que se dispone físicamente se optó por la de Nivel, Caudal y Presión, esto porque tienen un tiempo de respuesta más corto, lo cual se da por las características propias de los elementos empleados en la construcción de dichos módulos. En estos tres procesos aquel que maneja la variable de Nivel se caracteriza por ser desde el punto de vista del controlador de tipo integrante mientras que los otros dos procesos son autorregulados.

Para desarrollar el entorno embebido de los sistemas de Control Industrial de Caudal, Nivel y Presión es necesario entender cómo se comporta cada uno de ellos, lo cual se puede lograr gracias a la obtención de su comportamiento dinámico y su función de transferencia. Para esto se realiza una conexión en lazo cerrado (Figura 10), donde la entrada al sistema es un valor analógico proporcionado por el controlador (Control Value), acondicionado de 0 a 5 voltios que es el rango de voltaje que soporta el dispositivo embebido NI myRIO, la salida (Process Value) es un valor también analógico que varía de acuerdo a las características y funcionamiento de los elementos que componen la planta como la dimensión de las tuberías del sistema, accionamiento de las válvulas de carga, entre otros y de igual manera que las entradas tienen un rango de funcionamiento entre 0 a 5 voltios.

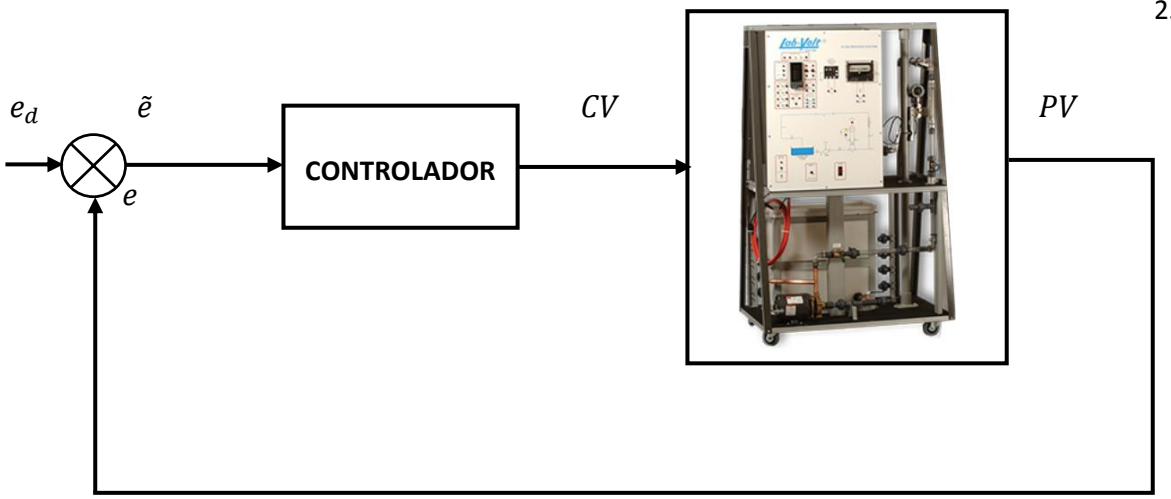

Figura 10. Lazo de Control del Dispositivo NI myRIO.

3.1.1. Dinámica y Función de Transferencia de Caudal

Para la obtención de la dinámica es necesario disponer del módulo físico del proceso de caudal, mediante el diagrama de lazo (Figura 11) se comprende mejor su funcionamiento y principalmente su conexión hacia un dispositivo que permita tomar datos y que acepte el tipo de variable de salida que se extrae de la planta de proceso, dicha señal en el módulo elegido es de corriente con un rango estándar de 4 a 20(mA), por lo cual es más óptimo emplear un PLC Siemens S7-1200 para la elaboración de este trabajo y se debe determinar en la programación del mismo que las entradas deben ser de corriente.

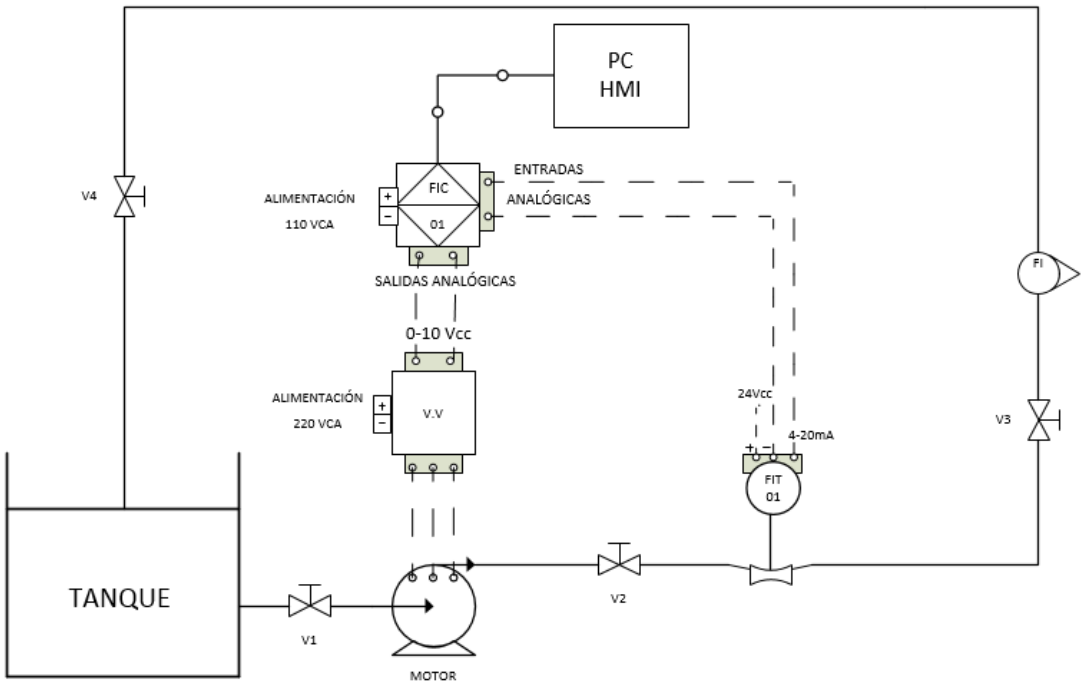

Figura 11. Diagrama de lazo del Sistema de Control de Caudal

El software que permite realizar la programación del PLC S7-1200 es el TIA Portal de SIEMENS. Antes de empezar a ingresar los bloques de programa es necesario configurar el dispositivo que se va a utilizar de acuerdo al modelo de la CPU, después de esto es posible escoger el módulo de entradas y salidas analógicas (Figura 12, Figura 13) y configurarlo donde se elige la variable eléctrica que se toma a la entrada que en este caso es corriente que es lo que proporciona el módulo de caudal LabVolt y la variable que se necesita a la salida del PLC la cual es de voltaje, además se determina la dirección IP del dispositivo y la submáscara de Red (Figura 14).

Cada sistema de Control tiene un valor máximo de operación y cuando de obtener la dinámica se trata es posible tomar ya sea todo el rango o solo una parte de él, aunque lo óptimo es trabajar entre los valores en los que se encuentra calibrado el transmisor para visualizar dentro de la misma planta los valores dentro del proceso y validar lo que se obtiene con el PLC.

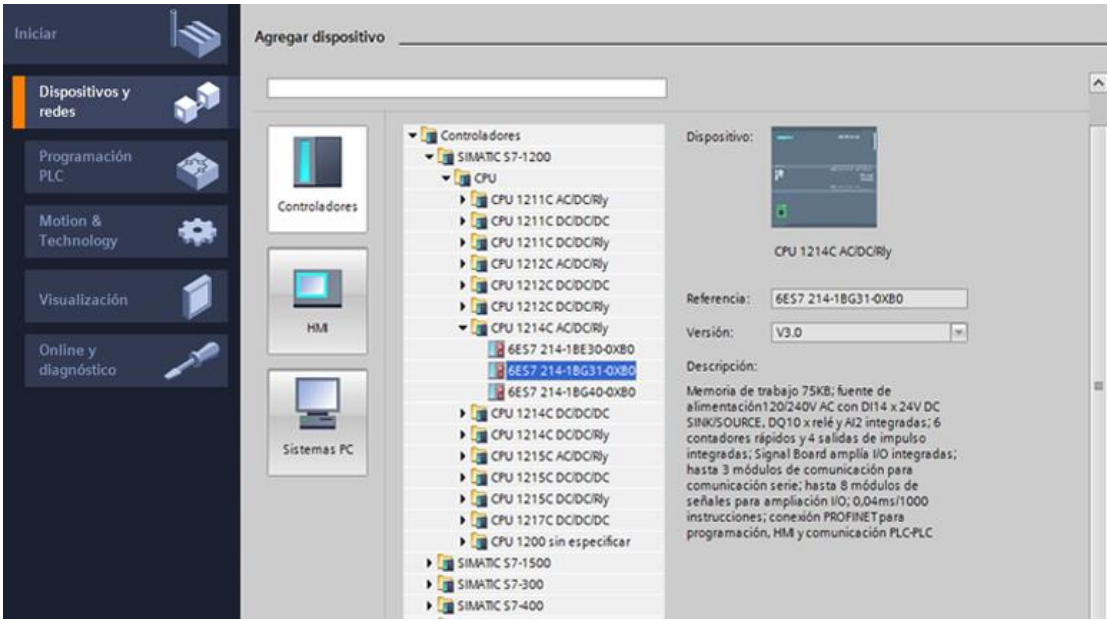

Figura 12. Configuración del dispositivo

Fuente: TIA Portal V13

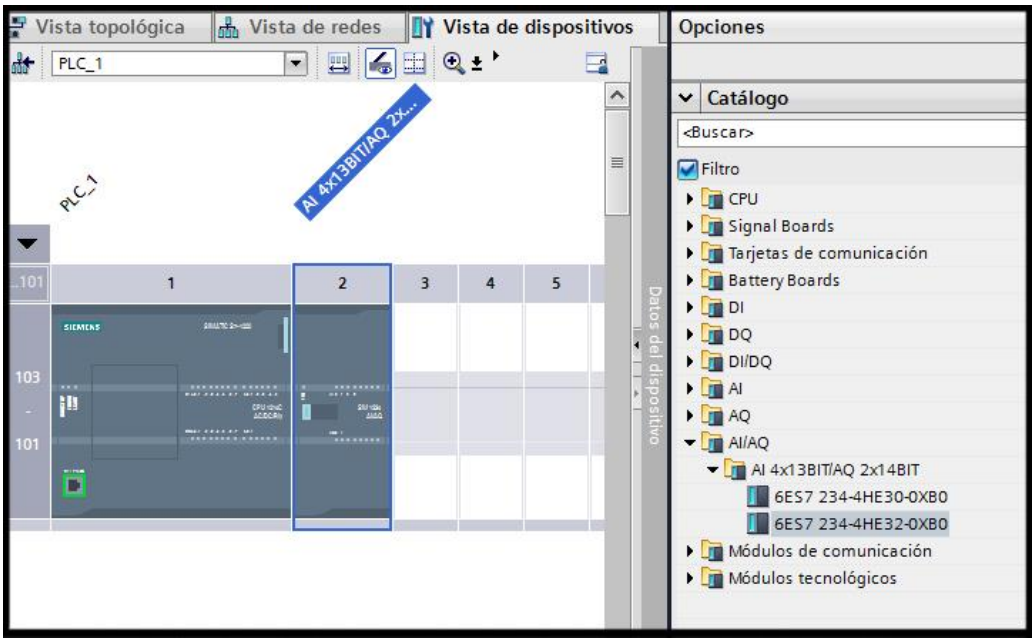

Figura 13. Módulo de entradas y salidas analógicas

Fuente: TIA Portal V13

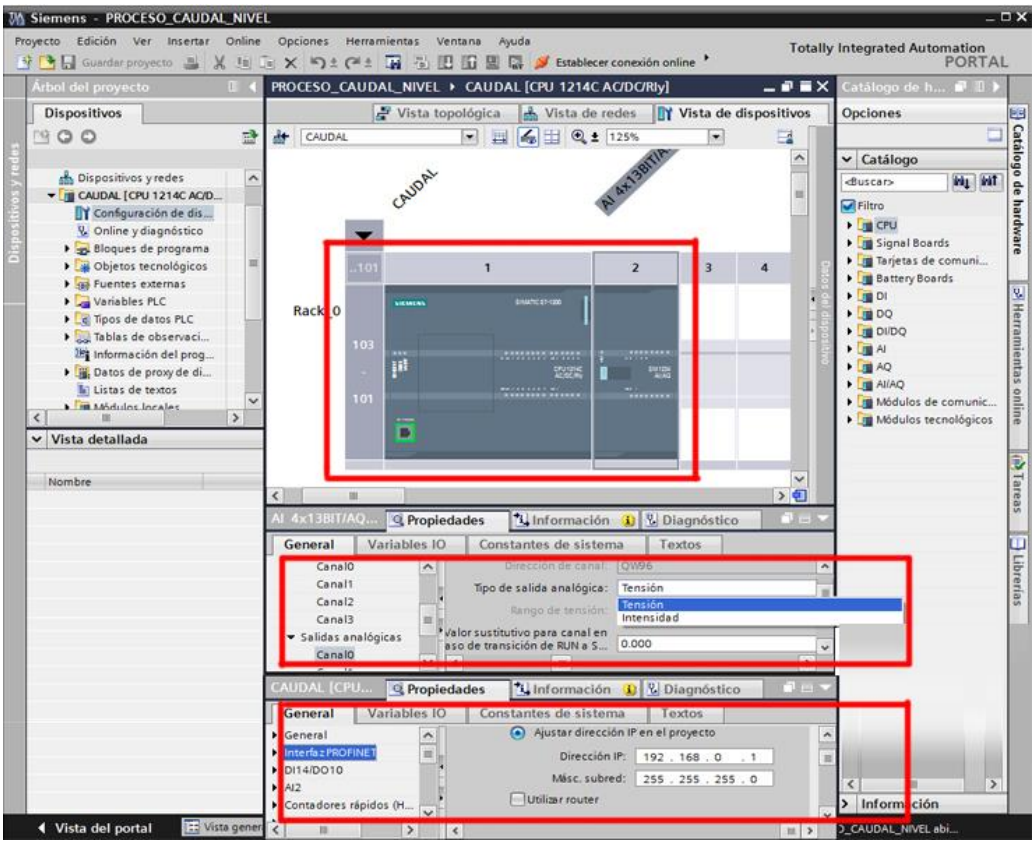

Figura 14. Configuración del PLC

Fuente: TIA Portal V13

El PLC S7-1200 para datos analógicos entiende valores entre [0; 27648] como un rango válido para magnitudes eléctricas estándar. Como una señal de corriente estándar comprende entre 4 y 20(mA) es necesario primeramente normalizar la entrada de datos a los rangos en el que el PLC toma como válido mediante la siguiente fórmula.

$$Valor\ mínimo\ del\ PLC = \frac{Valor\ mínimo\ de\ corriente * valor\ máximo\ del\ PLC}{Valor\ máximo\ de\ corriente} \quad (3.1)$$

El Sistema de Control de caudal tiene su transmisor calibrado en un rango de 10 a 40 litros por minuto por lo que para la entrada del PLC se toma esto como un rango de 0 a 100% y se utiliza un escalamiento. La salida del PLC se encarga de enviar valores de voltaje de 0 a 10 voltios por lo que el escalamiento es tomado en todo el rango, también de 0 a 100% es decir en el rango completo de [0; 27648] (Figura 15) y se enviará a la planta como retroalimentación de igual manera normalizándolo.

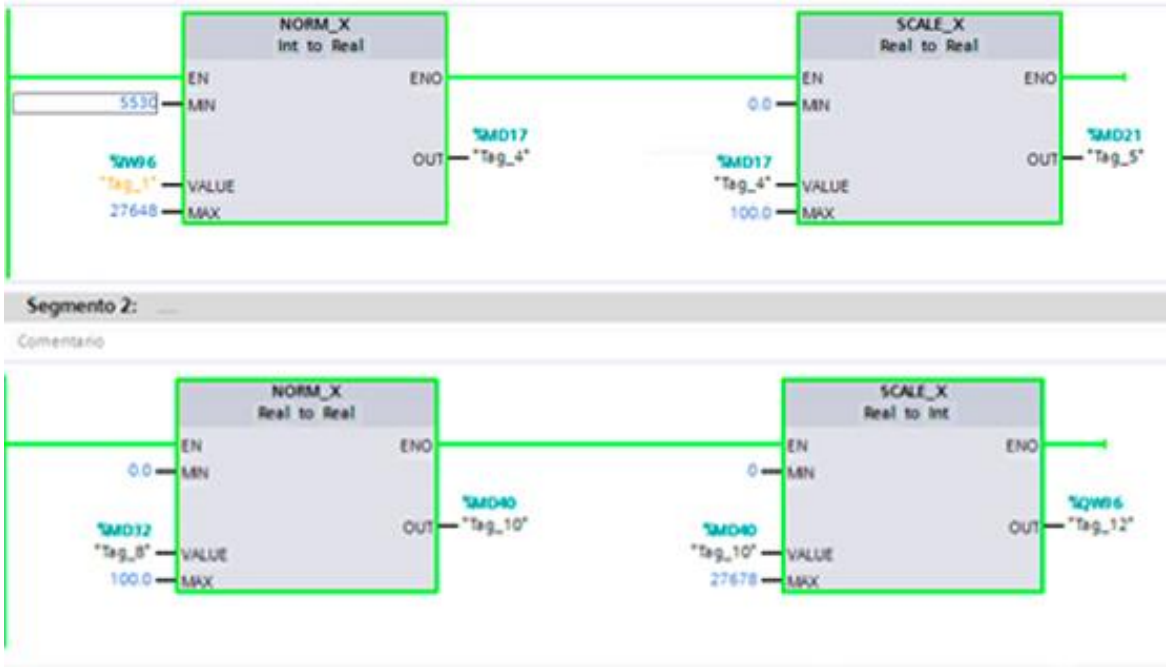

Figura 15. Acondicionamiento de la entrada y la salida

Fuente: TIA Portal V13

Con el programa cargado en el PLC es necesario disponer de un servidor OPC que cumple básicamente con la comunicación entre el controlador y el cliente OPC que es el HMI en el computador, del cual se envía una señal escalón que pueda excitar al sistema para que este sea capaz de enviar una respuesta al controlador y poder verla de regreso en el HMI. Al ser un proceso relativamente rápido en su respuesta, se toman 8 puntos de prueba (hacia arriba y hacia abajo) en el rango total para obtener una respuesta más exacta (Figura 16).

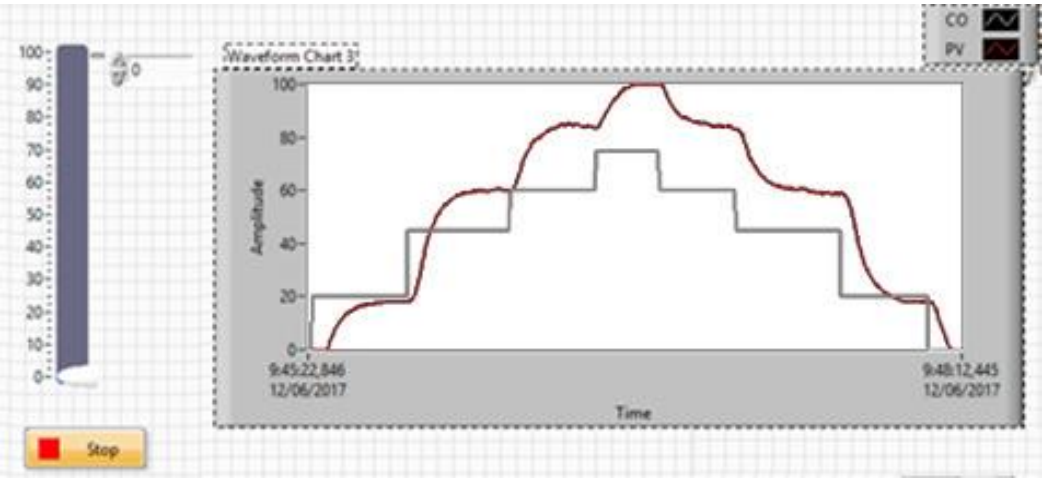

Figura 16. Señal de excitación y respuesta de la planta de Caudal

La Interfaz Humano Máquina brinda la facilidad al usuario para observar cómo se comporta la planta ante una señal de excitación. Desde la gráfica obtenida es posible generar un archivo ".xls" que contenga los datos de entrada y respuesta de la planta, estos datos con la ayuda de MATLAB son procesados y a través de su herramienta *IDENT* permite crear la función de transferencia del sistema real (Figura 17).

Figura 17. Función de transferencia en Matlab de Caudal

Fuente: Matlab 2012a

Donde la ecuación que describe el comportamiento de la planta de caudal es de primer orden más tiempo muerto como se puede ver en la ecuación 3.2, desde aquí se parte para la creación del sistema embebido y también da la pauta para calcular las constantes de sintonía que se hablará en la sección 4.3.

$$G(s) = \frac{1.9211}{1.2168s+1} e^{-0.7023} \tag{3.2}$$

3.1.2. Dinámica y Función de Transferencia de Nivel

Con el motivo de obtener la dinámica se realiza la conexión apoyado en el diagrama de lazo de la Figura 18, lo que permite extraer la señal a la salida de la planta, dicha señal en el módulo de Nivel de LabVolt es de corriente con un rango estándar de 4 a 20 miliamperios, se emplea igualmente el PLC S7-1200.

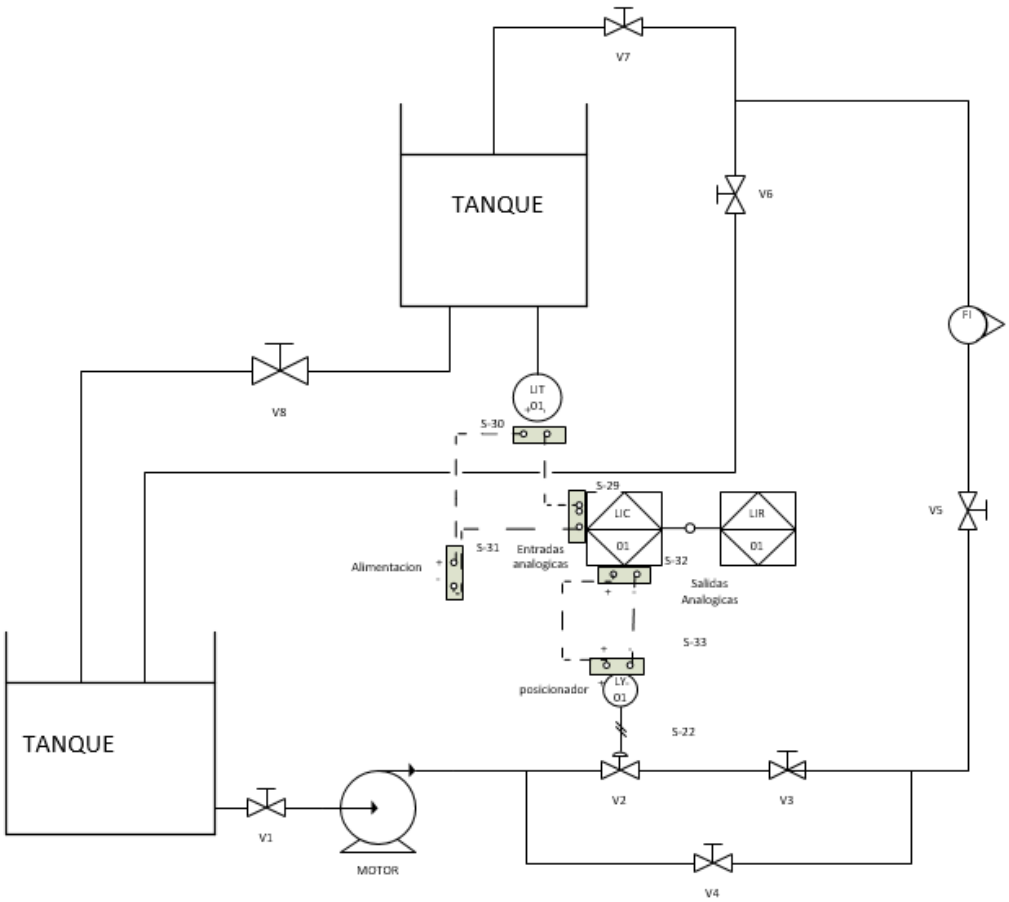

Figura 18. Diagrama de lazo del proceso de Nivel

Por medio de la ecuación 3.1 se calcula el rango de datos analógicos del PLC equivalentes a 4 y 20 miliamperios y se genera el escalamiento en la programación. El proceso de Nivel tiene su transmisor calibrado en un rango de 20 a 70 litros el cual se toma como el rango de 0 a 100% para el PLC. El programa para la obtención de la dinámica de la planta de Caudal es reutilizable para obtener la dinámica de la planta de Nivel (Figura 15). Con el programa cargado en el PLC y los *tags* creados y asignados en el servidor OPC se asigna un valor de la señal escalón para excitar el sistema y se registra la gráfica. (Figura 19).

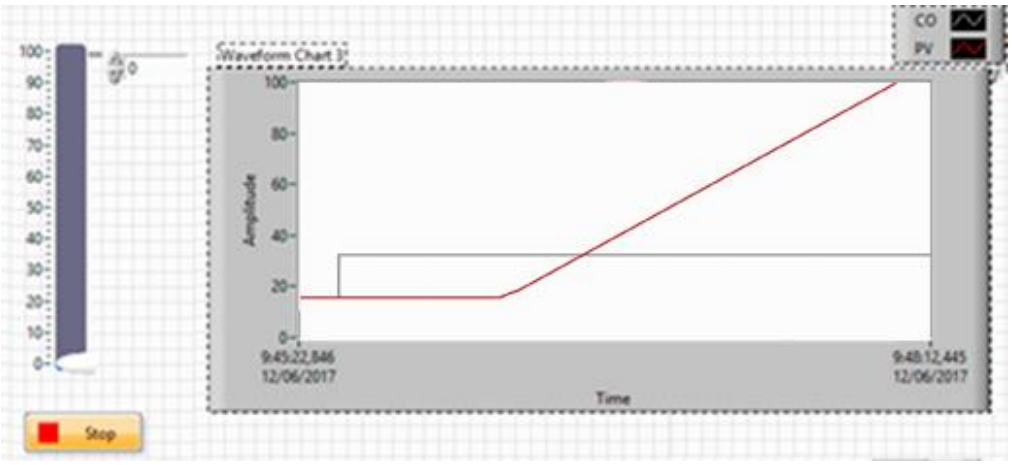

Figura 19. Señal de excitación y respuesta de la planta de Nivel

Con la gráfica se genera un archivo de extensión“.xls” que guarda el registro de datos, este Sistema es de tipo integrante por lo que la respuesta del sistema tiende al infinito. La función de transferencia de este sistema no es posible obtenerla con la herramienta *ident* de MATLAB, pero se la puede calcular a través de las Ecuaciones 3.2 y 3.3 extrayendo de la gráfica como se muestra en la Figura 20 los datos necesarios exactos para satisfacer dichas ecuaciones.

$$K = \frac{S_2 - S_1}{(E_2 - E_1) * (T_3 - T_2)} \tag{3.3}$$

$$Td = K + T_2 - T_1 \tag{3.4}$$

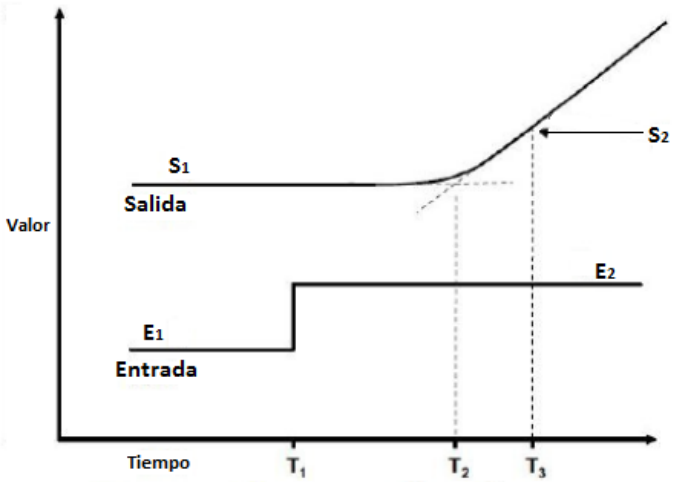

Figura 20. Parámetros para la función de transferencia de un Sistema Integrante

Cuando se han realizado los cálculos necesarios es posible armar la ecuación característica de un proceso integrante más tiempo muerto. La ecuación resultante que describe el comportamiento de la planta de Nivel se puede ver en la ecuación 3.5.

$$G(s) = \frac{0.4997}{s} e^{-4.2} \tag{3.5}$$

3.1.3. Dinámica y Función de Transferencia de Presión

El proceso de presión se lo conecta como en el diagrama de lazo de la Figura 21, esto con el fin de crear un sistema de segundo orden. El diagrama de lazo facilita el entendimiento del funcionamiento de este proceso además de contribuir con la conexión del módulo hacia el PLC para lograr extraer los datos de su variable en el tiempo.

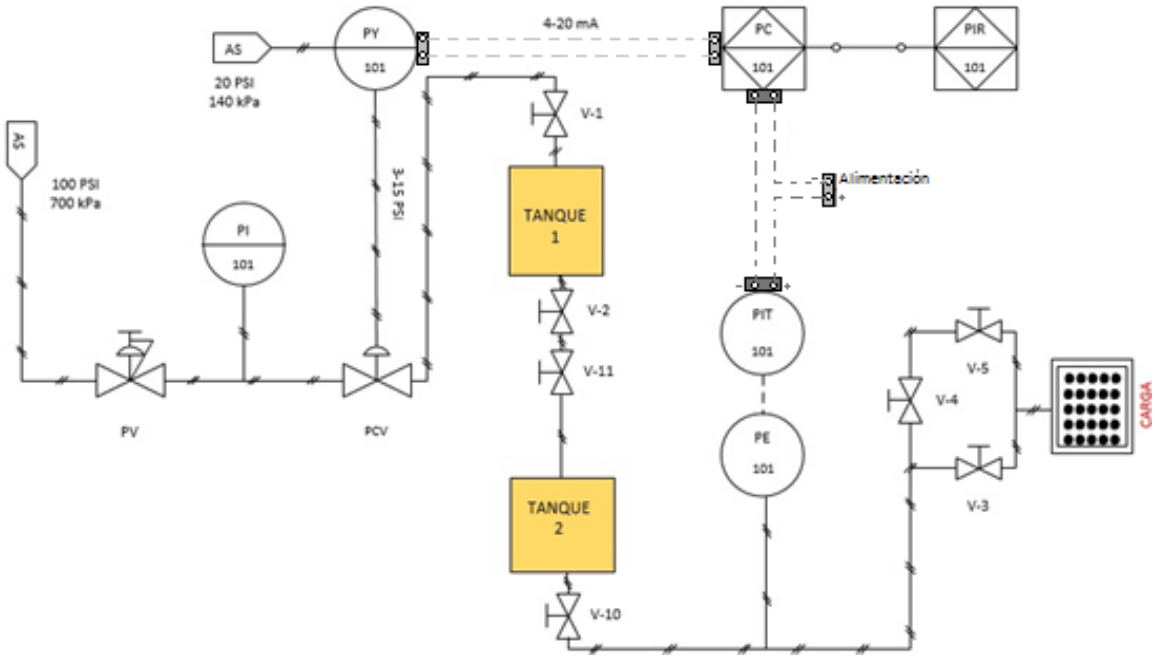

Figura 21. Diagrama de lazo del Sistema de Control de Presión

En el software TIA Portal de SIEMENS se realiza la programación de igual manera que para los módulos de procesos de primer orden, tomando en cuenta que el proceso de Presión tiene su transmisor calibrado en un rango de 0 a 40 PSI el cual se toma como el rango de 0 a 100% para el PLC (Figura 15). Cuando el programa se encuentra cargado en el PLC y los *tags* creados y asignados en el servidor OPC se asigna un valor para la entrada escalón (Figura 22).

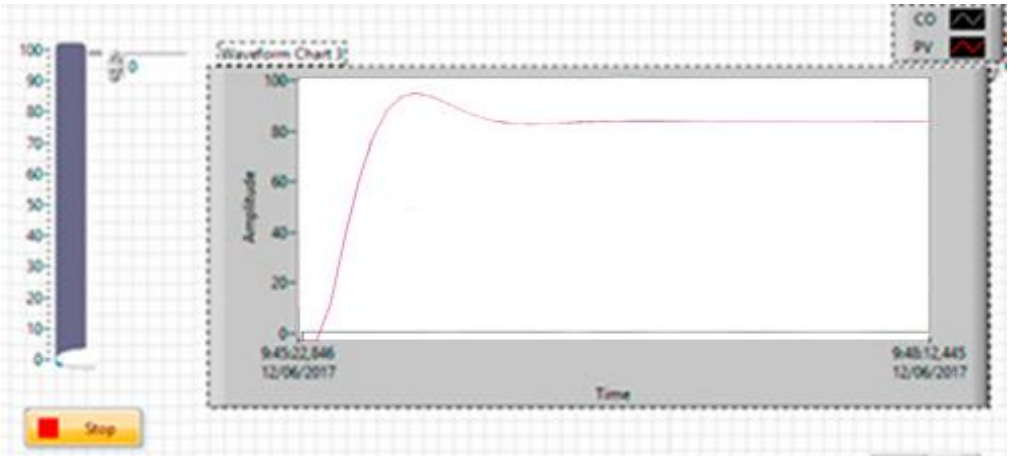

Figura 22. Señal de excitación y respuesta del Sistema de presión

Al tener ya disponible el archivo ".xls" con los datos registrados, mediante la herramienta ident de MATLAB se crea la función de transferencia del sistema real (Figura 23), con la diferencia que aquí se escoge el modelo de una ecuación de orden dos.

Figura 23. Modelo matemático obtenido en Matlab

Fuente: Matlab 2012a

La ecuación resultante que describe el comportamiento de la planta de Presión es de segundo orden más tiempo muerto como se puede ver en la Ecuación 3.3, para el controlador la sintonía de este sistema difiere a uno de primer orden, esto se tratará en la sección 4.4.

$$G(s) = \frac{25.011}{71.1478s^2+70.7257s+1} e^{-0.0862} \tag{3.6}$$

3.2. Configuración inicial de NI myRIO

Estas funciones de transferencia obtenidas son la base de las plantas embebidas a través del software LabVIEW, el sistema embebido reemplazará a la planta real en el método Hard in the Loop (HIL). Un requerimiento básico para utilizar el dispositivo embebido para estudiantes myRIO es tener instalados todos los paquetes del mismo puesto que esto permite crear proyectos compatibles, compilarlos y grabarlos en el dispositivo. La forma de comprobar que el paquete de NI myRIO está correctamente instalado es iniciar el software LabVIEW y en la pantalla principal se observa la palabra "myRIO" como se ve en la Figura 24.

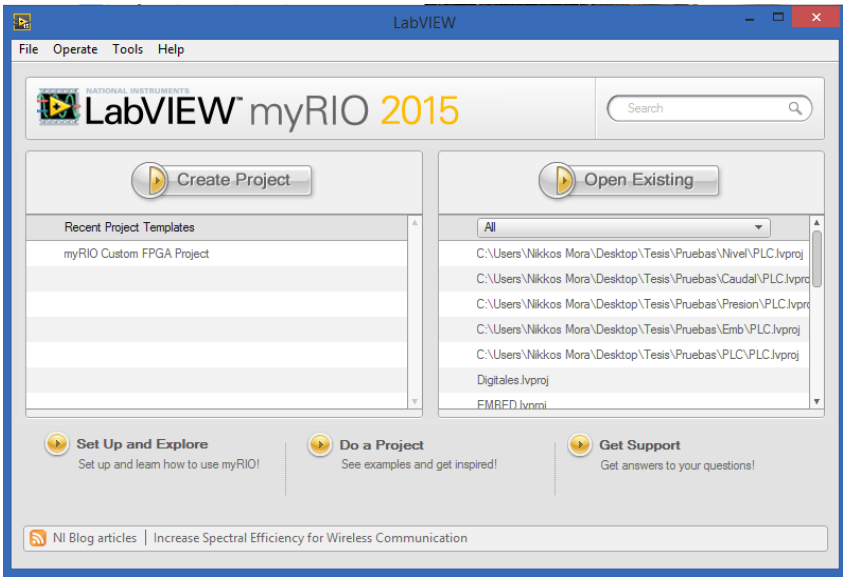

Figura 24. Pantalla Inicial LabVIEW
Fuente: LabVIEW 2015

Para iniciar el desarrollo del entorno es necesario antes configurar el dispositivo embebido, para lo cual se lo debe conectar al computador y esperar a que se aparezca una ventana Monitor propia de myRIO donde se visualizan las características del dispositivo conectado y una lista de opciones.

En el caso de ocurrir un problema en la conexión con el dispositivo aparece una ventana con un mensaje de advertencia donde el Monitor USB myRIO pide la verificación de los cables. Para esto se procede a revisar que el dispositivo esté conectado correctamente con la computadora mediante su conexión USB y también que el dispositivo se encuentre conectado a la alimentación de voltaje, a simple vista también se puede comprobar el estado de la conexión observando los LED's indicadores que se muestra a un costado del dispositivo embebido NI myRIO.

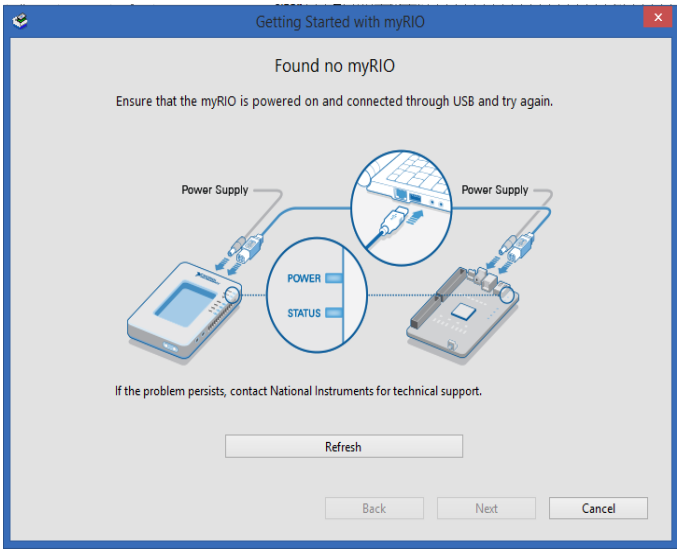

Figura 25. Pantalla Inicial LabVIEW

Fuente: NI myRIO

El dispositivo gestiona tanto la conexión como la existencia de los paquetes necesarios para el correcto funcionamiento entre LabVIEW y myRIO, trabaja con algunos módulos específicos de LabVIEW por lo que de forma automática verifica si se encuentran instalados tanto el software de National Instruments como el módulo Real–Time con lo que se activará la segunda opción del menú.

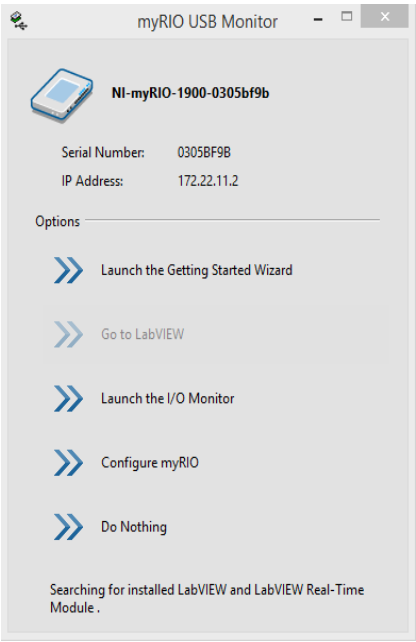

Figura 26. Pantalla myRIO USB Monitor
Fuente: NI myRIO

Para utilizar por primera vez el dispositivo se selecciona la primera opción, donde se inicializan algunas configuraciones. La primera ventana reconoce todos los dispositivos myRIO que detecte el computador, si no detecta y reconoce automáticamente el o los dispositivos conectados se puede actualizar o refrescar el listado para escoger la opción con la cual se va a trabajar.

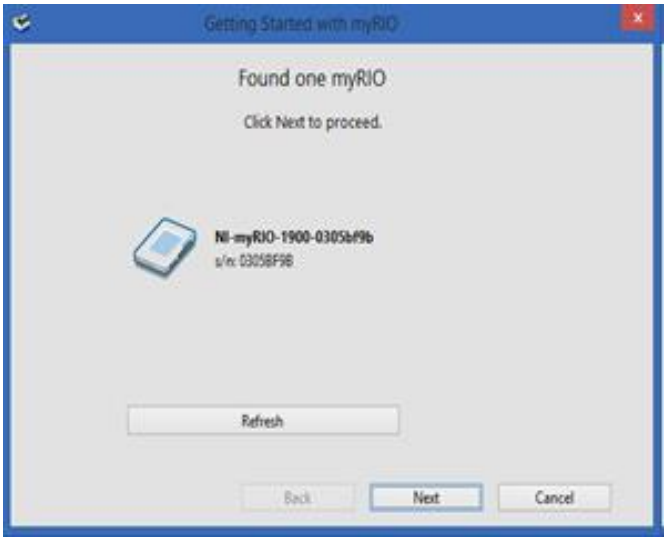

Figura 27. Ventana de Reconocimiento myRIO
Fuente: NI myRIO

La siguiente pantalla permite renombrar el dispositivo o se puede utilizar por defecto el nombre propio del dispositivo.

Figura 28.Ventana para nombrar al dispositivo
Fuente: NI myRIO

En el siguiente paso se chequea que el software de compatibilidad sea correcto, si es así da paso al uso del dispositivo, caso contrario NI myRIO presentará un listado con acciones por parte del dispositivo, se elige la opción recomendada donde instalará la versión de software exacta para que sea compatible y no presente ningún problema en el desarrollo.

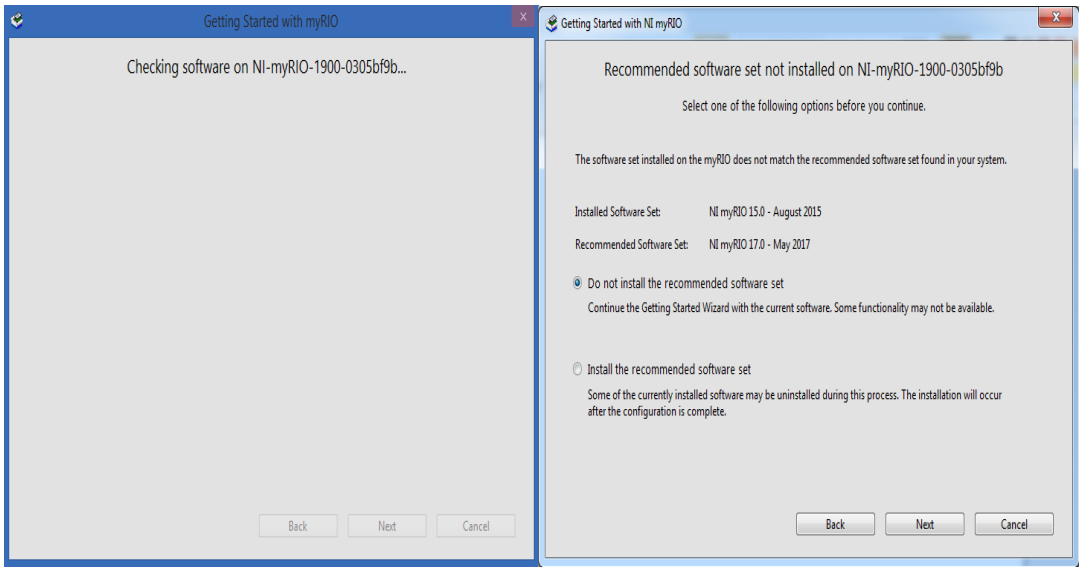

Figura 29. Ventana para nombrar al dispositivo
Fuente: NI myRIO

Se finaliza la configuración mediante una ventana que permite realizar un test del dispositivo y los elementos de los que se encuentran compuesto internamente como el acelerómetro, los Led's de estado que pueden ser programables y un pulsador.

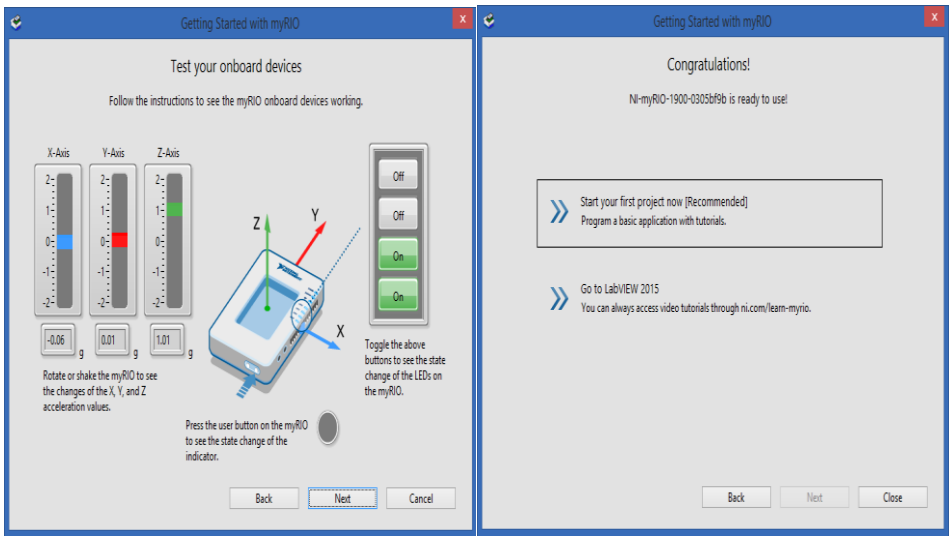

Figura 30. Ventana probar los elementos internos
Fuente: NI myRIO

3.3. Programación para la tarjeta NI myRIO.

3.3.1. Elaboración de proyectos myRIO para LabVIEW

Con el dispositivo preparado, se inicia el software LabVIEW para desarrollar el proyecto donde se realizó la programación. El propósito de utilizar este software es que aporta muchas facilidades al utilizar un lenguaje de programación G o Gráfica por la gran capacidad que nos ofrece este tipo de programación por medio de bloques lógicos y líneas de información que manejan todo el algoritmo para el diseño de hardware y software, el desarrollo de una lógica compacta implementada a sistemas más complejos producen un aumento en la eficiencia de todos los usuarios a través de la simplificación de la programación de sus aplicaciones generando un gran ahorro de tiempo y manteniendo una comunidad capacitada por National Instruments. El desarrollo se puede evidenciar más en sistemas de control, monitoreo, supervisión, reales, simulados y embebidos. Se crea un proyecto para myRIO desde la pantalla principal como se observa en la Figura 31, este proyecto permite crear un entorno donde se pueda realizar la programación que controle el dispositivo embebido y todas sus funciones mediante el toolkit de myRIO.

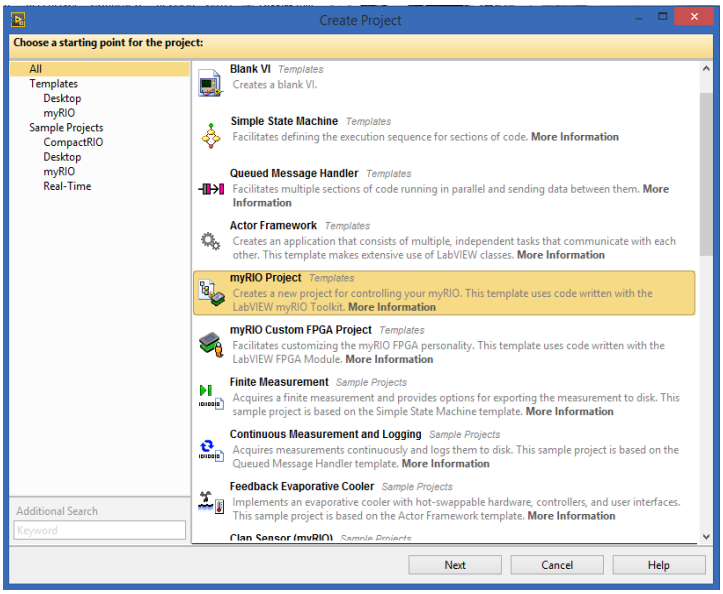

Figura 31. Proyecto para myRIO

Fuente: NI myRIO

Al seleccionar myRIO Project automáticamente se genera una plantilla donde se ingresa el nombre del proyecto y su ubicación, es importante que se encuentre conectado el dispositivo para poder seleccionar el tipo de comunicación por puerto USB con el computador, que es el tipo de comunicación elegido para este trabajo, caso contrario cuando se selecciona esta opción la plantilla y sus funciones aparecen en blanco, el tipo de comportamiento de las entradas y salidas del dispositivo se encuentran en modo por defecto ya que esta característica es suficiente para aplicaciones de control. Posteriormente se termina la creación del proyecto mediante el botón Finish.

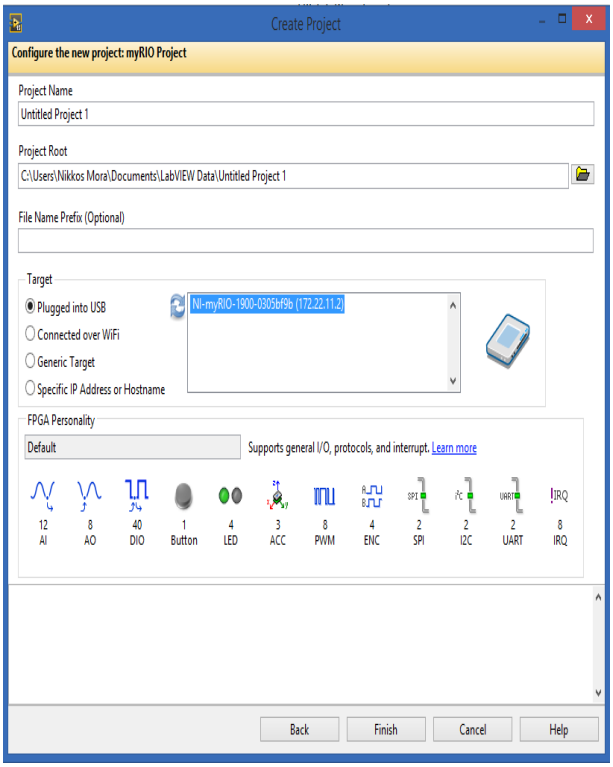

Figura 32. Creación del proyecto para myRIO

Fuente: NI myRIO

Cuando se finaliza la creación del proyecto se presenta la ventana de la Figura 33 donde se abre un menú del proyecto para trabajar en la tarjeta myRIO, directamente en el Main.vi que está dentro del ítem de la tarjeta myRIO con el nombre que fue asignada.

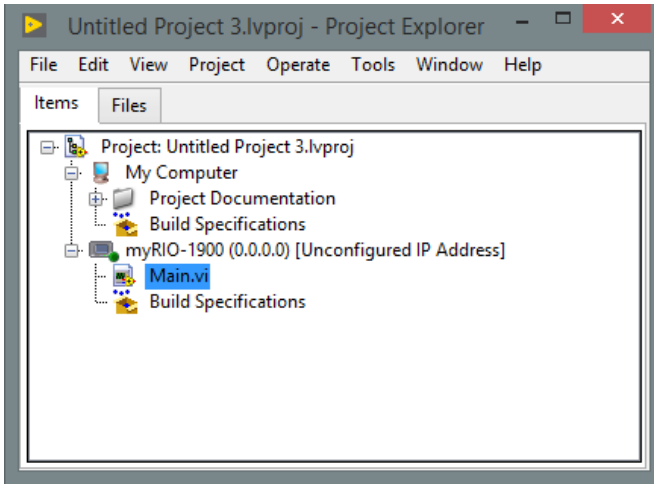

Figura 33. Creación del proyecto para myRIO

Fuente: NI myRIO

En el VI se puede manejar distintas plantillas para formar una estructura. El proyecto maneja un tipo de plantilla formada por un flat sequence con tres frames que nos permiten:

- **Primer frame –** Inicializa los parámetros para NI myRIO como variables para controladores e indicadores y algunos datos para el usuario.

- **Segundo frame –** Desarrolla toda una estructura donde se realiza la adquisición y el procesamiento de los datos que son enviados al dispositivo. Siempre debe mantenerse con una estructura While Loop que permita saltar al último frame cuando se pulse el botón de STOP, en este frame básicamente se desarrolla la programación de los procesos y su funcionamiento mediante las entradas y salidas del dispositivo.

- **Tercer frame –** Finaliza los parámetros de la NI myRIO, variables y datos de usuario donde también se puede presentar datos o mensajes.

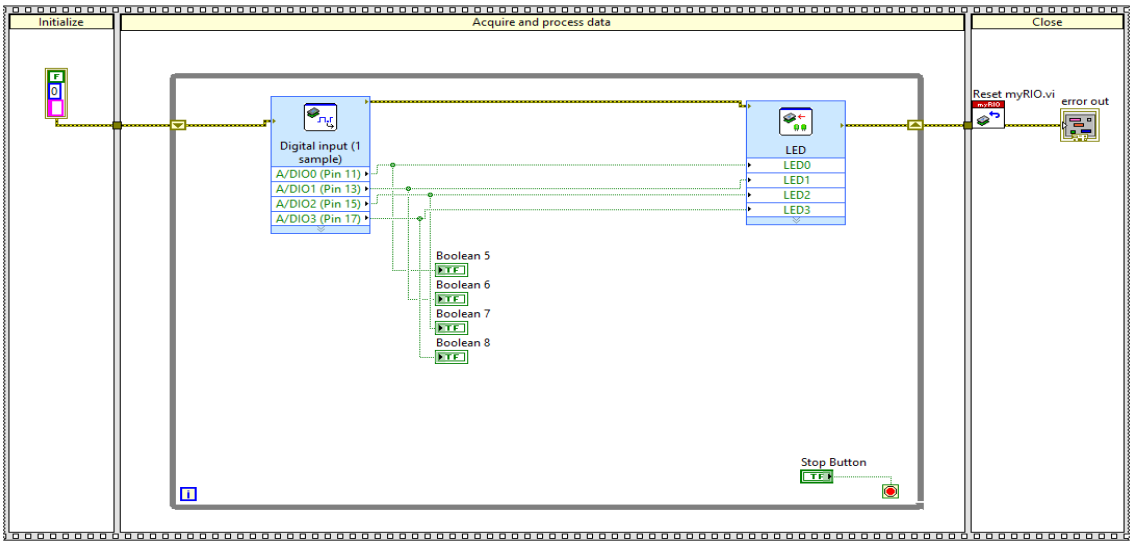

Figura 34. Frames del programa principal del proyecto

Fuente: LabVIEW 2015

Con el módulo de LabVIEW myRIO y sus librerías se facilita el trabajo ya que se cuenta con bloques que manejan las entradas y salidas al igual que algunas funciones del dispositivo NI myRIO-1900. Con la contribución de los módulos Real-Time, Control

Design & Simulation y myRIO es posible crear el sistema embebido para utilizarlo en el método Hardware in the Loop (HIL). La realización del entorno como se mencionó se realiza en el segundo frame que maneja todos los datos. Aquí se incorpora una estructura Timed Loop, Control & Simulation loop del módulo Control Design & Simulation y el Toolkit de myRIO.

Se agrega la estructura Timed Loop que es un bucle temporizado con un periodo definido y un estado crítico que fija una prioridad para el frame. Logrando una sincronización de los datos entre Ni myRIO-1900 y la unidad de control Electrónico (ECU) o PLC Siemens S7-1200.

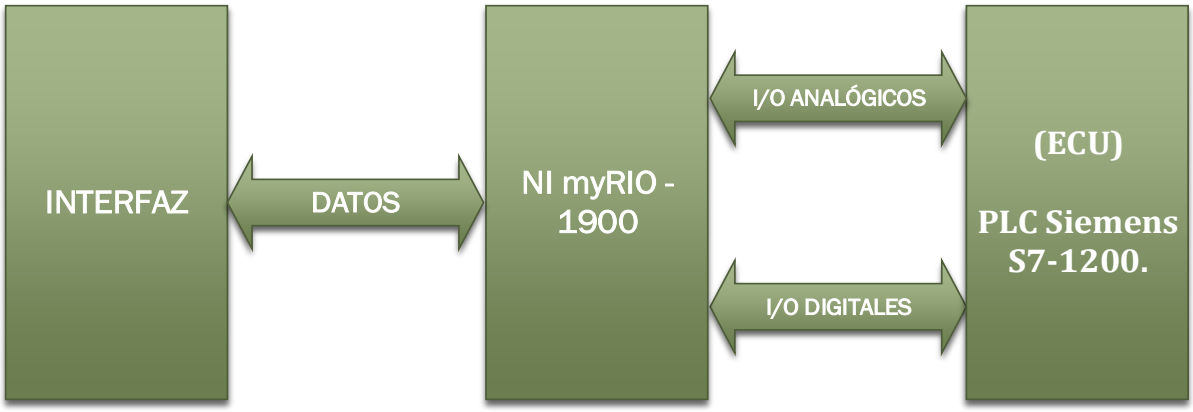

Figura 35. Flujo de Datos entre Dispositivos

Para ingresar el modelo matemático que se obtuvo mediante la dinámica de los procesos se utiliza el Control & Simulation Loop que permitirá el manejo de toda la información que se obtenga de las tres estaciones en desarrollo. Convirtiendo la información en espacio de tiempo, a tipos de datos manejables por el software. Es de gran importancia que en el desarrollo para el manejo correcto de la información no falte encapsular el flujo de datos en un bucle con retro alimentación.

Para el entorno se cierra el bucle con la entrada y salida analógica del PLC Siemens S7-1200. Unificando todas las herramientas que construyen el esquema básico del entorno que posteriormente se utiliza para obtener la dinámica del sistema embebido para que sea validado mediante la dinámica del Sistema Real. Obsérvese en la siguiente figura.

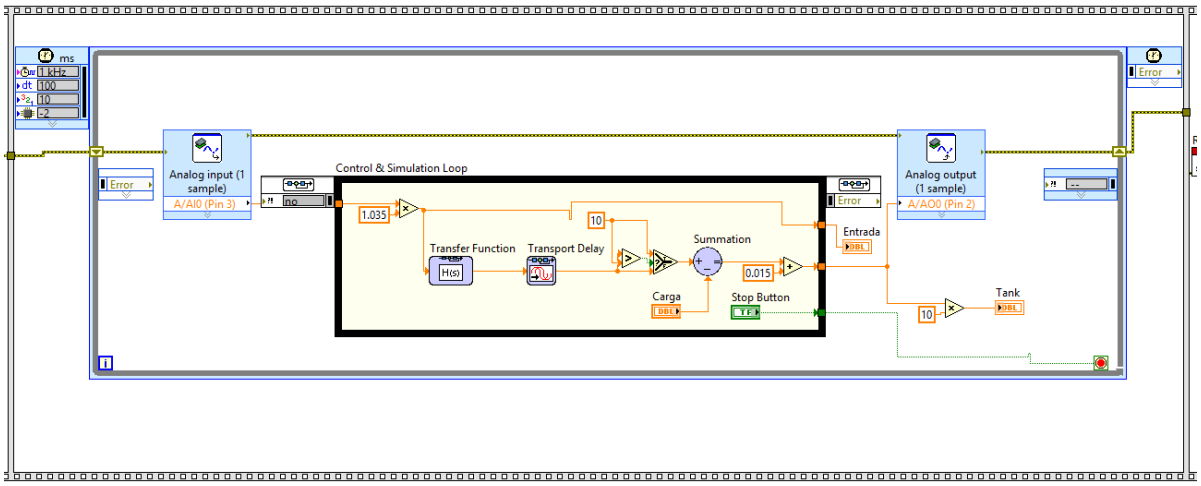

Figura 36. Frames del programa principal del proyecto

Fuente: LabVIEW 2015

En el panel frontal se desarrolló la simulación de las plantas para el laboratorio a través del módulo Datalogging and Supervisory Control (DCS) de LabVIEW, que presenta varias herramientas para la animación de los procesos de nivel caudal y presión generando entornos intuitivos para el usuario.

Para activar una de las herramientas del módulo DCS ingresamos al menú de herramientas en la opción “DCS Module” y seleccionamos “Image Navigator”. Se abre una pantalla con varias imágenes dividas por grupos de trabajos dependiendo la necesidad el diseñador.

3.3.2. Diseño del panel frontal con el módulo DCS de LabVIEW

Para el panel frontal se utilizaron con mayor frecuencia las imágenes de válvulas, tuberías, pulsadores, y equipos que se pueden modificar según sea las características que se le quiera asignar a las imágenes. Con el botón “Options” es posible modificar el relleno de la imagen, su orientación y manejar otras opciones, si los cambios realizados modifican de manera inapropiada la imagen se puede restaurar las imágenes a su estado por defecto con el botón “Defaults” y seleccionar “OK” para trabajar con la misma imagen.

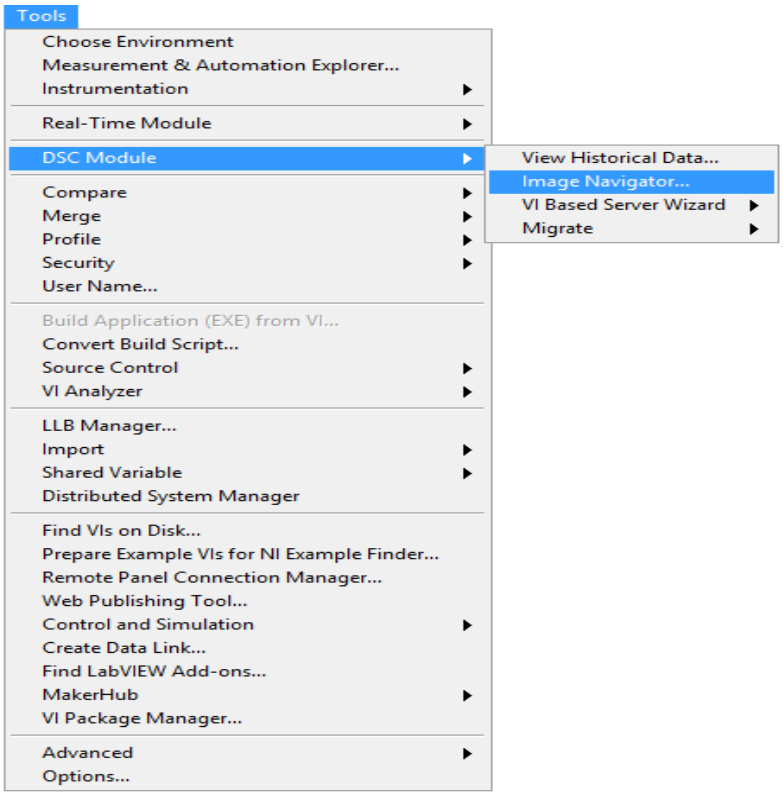

Figura 37. Menú de herramientas
Fuente: LabVIEW 2015

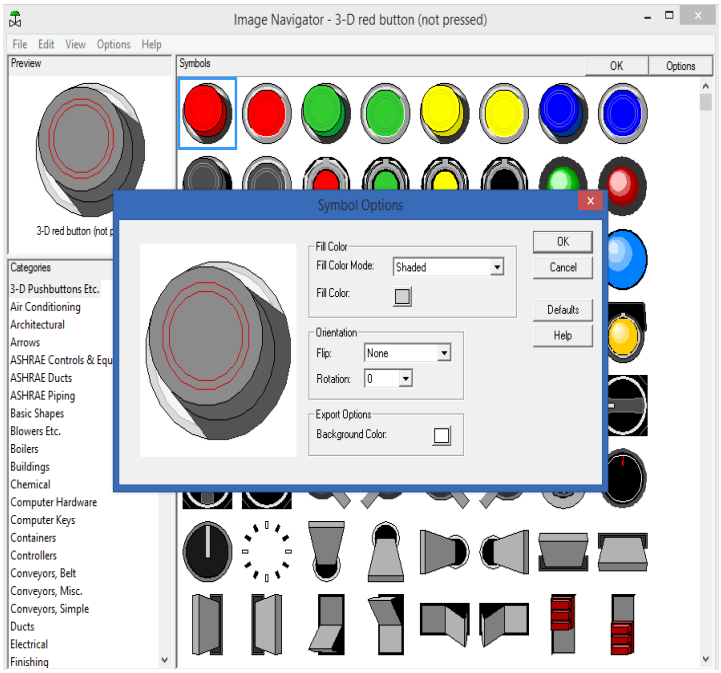

Figura 38. Navegador de imágenes

Fuente: LabVIEW 2015

Existe la opción de incorporar más imágenes en el navegador ingresando al menú “File” y seleccionando “New Category”, donde se abre una ventana para agregar la característica y el nombre de la categoría, para aceptar se selecciona “OK”. Se debe añadir una nueva carpeta donde se almacene las imágenes creadas, esto se logra importando desde el menú “FILE” una imagen de tipo BMP o WMF que se encuentre en el equipo.

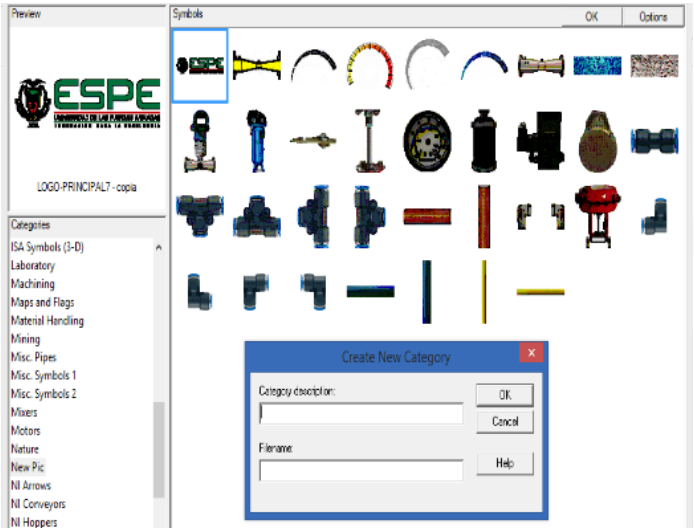

Figura 39. Creación de imágenes

Fuente: LabVIEW 2015

Se requiere una mayor exigencia en las animaciones de la interfaz de los procesos y LabVIEW nos permite crear nuevos bloques de control a partir de botones, sliders entre otros. Para el trabajo planteado es necesario que los controles y elementos que se presenten en la interfaz Humano Máquina sean similares a los que se puede encontrar en un ambiente profesional para generar una mejor inmersión del usuario. Para generar un nuevo bloque de control se necesita tener previamente un bloque control base donde se reemplazará su apariencia más no su funcionamiento, se hace clic derecho sobre él para ingresar a las opciones avanzadas y se selecciona “Customize”. En la Figura 40 se observa una ventana de LabVIEW con un bloque de control a modificar, cuando se observa la figura de un lápiz en la parte superior de la barra, se pueden realizar modificaciones internas del controlador como cambiar el nombre, los textos presentes en el bloque de control, los colores del mismo o importar imágenes en este caso dependiendo del estado que se encuentre el botón, para el caso de otros bloques de control como un slider, se manejan más áreas en donde se modifican los fondos del bloque de control e incluso se cambian los apuntadores. Sobre la figura del lápiz se hace clic izquierdo para cambiar el estado y realizar

únicamente modificaciones exteriores al control. Todos los bloques de control creados se guardan en el proyecto de la NI myRIO-1900 y es una alternativa guardar en el computador.

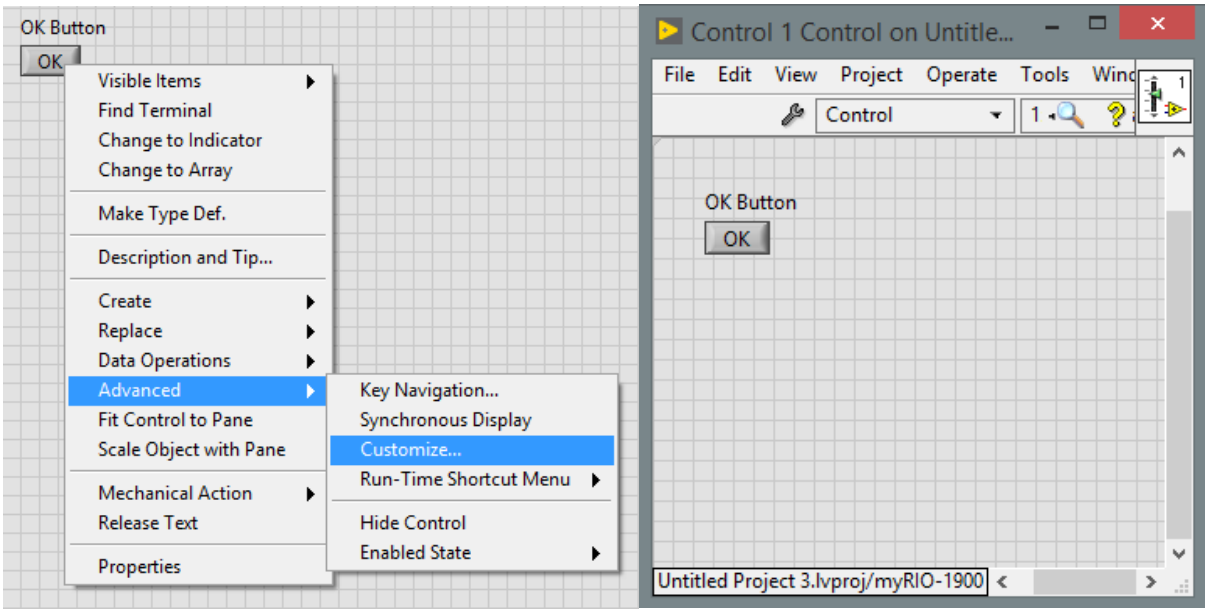

Figura 40. Creación de imágenes

Fuente: LabVIEW 2015

Para la interfaz que se muestra al usuario se ordena el sistema de acuerdo al proceso real creando cada elemento con la mayor similitud posible al proceso real para que pueda ser comprendido de la mejor forma por el usuario, además se generan indicadores de algunos parámetros como el estado de las válvulas del sistema, alarmas, y su función de transferencia como se muestra en las siguientes figuras.

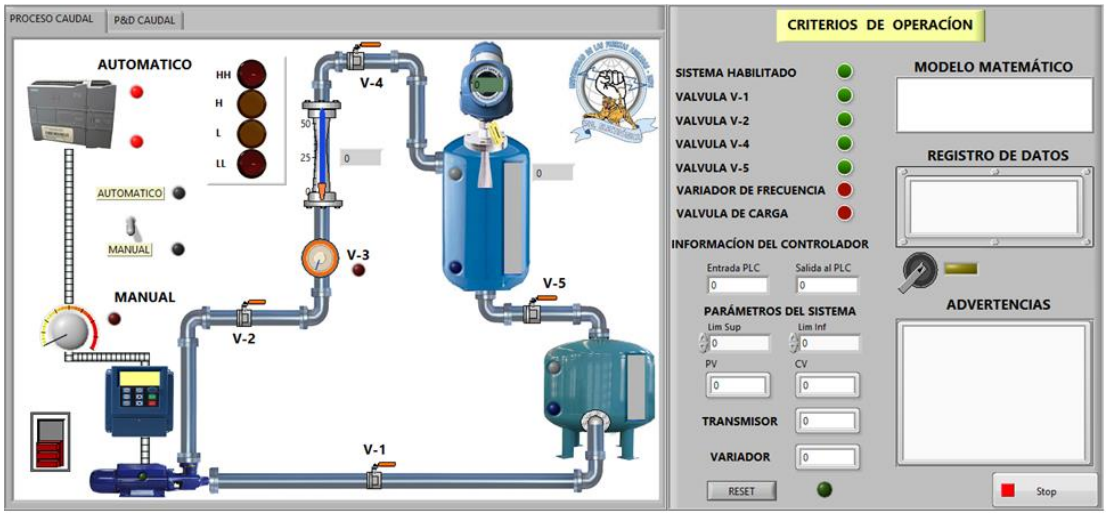

Figura 41. Interfaz Humano Máquina del Sistema de Nivel

Fuente: LabVIEW 2015

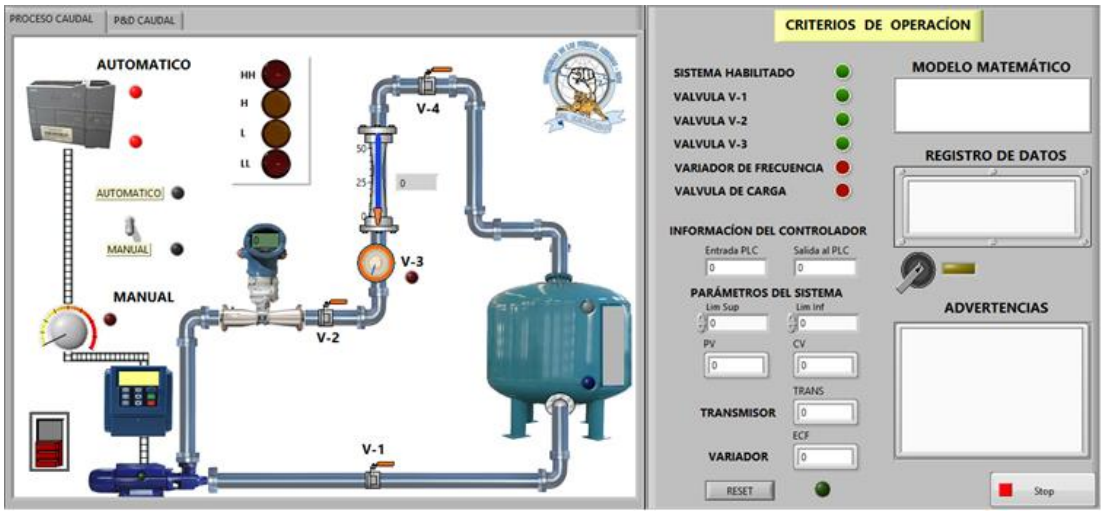

Figura 42. Interfaz Humano Máquina del Sistema de Caudal

Fuente: LabVIEW 2015

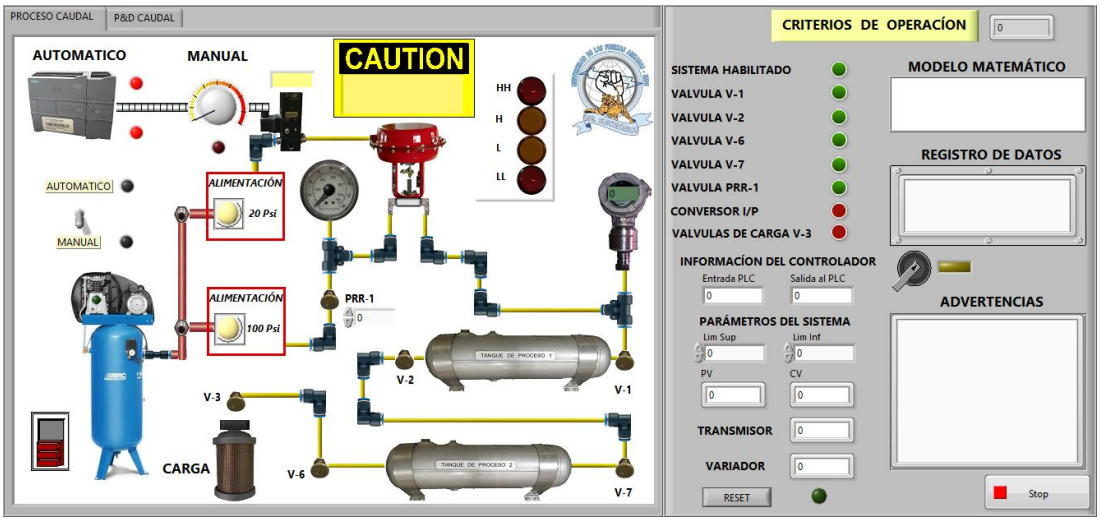

Figura 43. Interfaz Humano Máquina del Sistema de Presión

Fuente: LabVIEW 2015

3.3.3. Programación interna del VI principal para los procesos simulados

Para la programación se tiene estructurado un esquema basado en la plantilla de flat sequence, se describe en la siguiente figura las partes de cada uno de los frames.

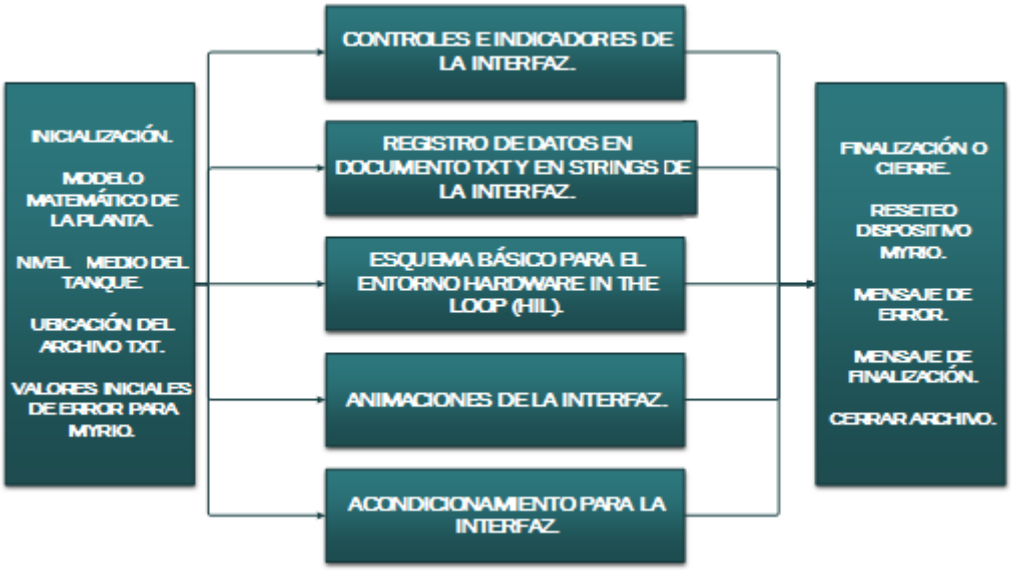

Figura 44. Esquema de programación

Para el primer frame se maneja la inicialización de las variables para el error del módulo de myRIO, se coloca un nivel inicial del tanque, se dibuje la ecuación de la función de transferencia y se selecciona la ubicación del archivo TXT.

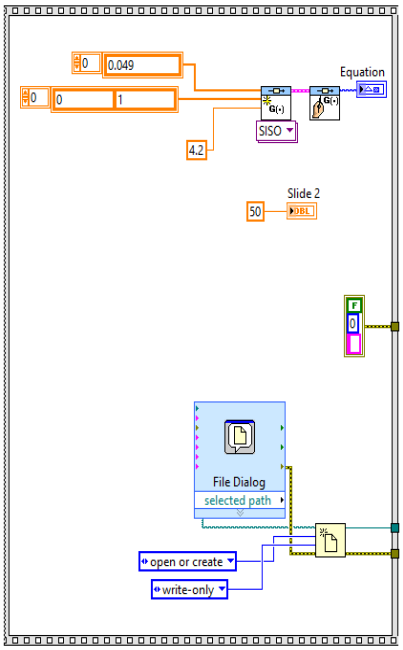

Figura 45. Frame de Inicialización

Fuente: LabVIEW 2015

El segundo frame contiene toda la adquisición y manejo de la información que llega por parte de la unidad de control electrónica (ECU), en la Figura 46 se muestra agrupada por partes la programación para el tratamiento de la información y la creación de acciones más interactivas para el usuario.

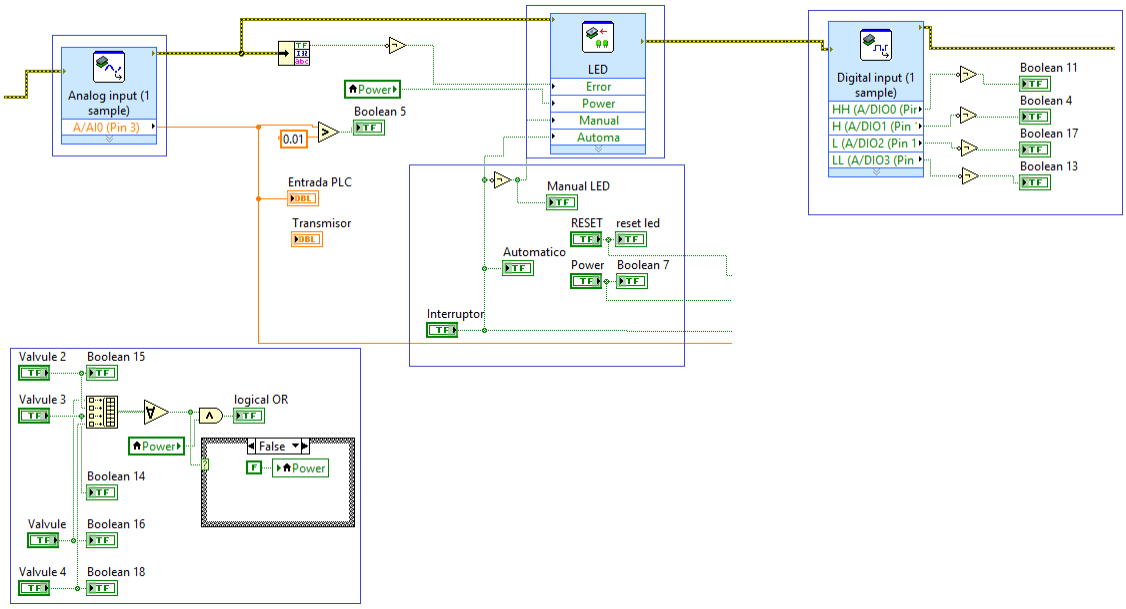

Figura 46. Bloques de control e indicadores principales

Fuente: LabVIEW 2015

Los controles con la palabra “Valvule” tienen la apariencia de una válvula en el panel frontal y se requiere que los bloques de control cumplan la condición necesaria para habilitar todo el sistema. Presentan dos estados cerrado o abierto con el fin de construir una matriz de tipo booleano, cada uno tiene un bloque de tipo “Boolean” que es un led indicador de estado. La salida de la matriz llega a la función “AND Array Elements” que arroja un solo valor de tipo booleano, si todos los valores de la matriz son verdaderos o están vacías se tiene un resultado igual a “TRUE”, caso contrario si presenta alguna desigualdad en la formación de la matriz el resultado es “False”. A continuación existe una estructura “CASE” para controlar el valor de interruptor “Power”, donde para un valor de TRUE está vacío con el motivo de disponer del estado de “Power” para que pueda ser utilizado por el usuario, mientras que en la condición falsa de forma automática se coloca un valor de FALSE al estado de “Power” para que no se habilite el sistema. Luego con una función AND que es el resultado del estado actual de “Power” con “AND Array Elements” indica que se cumple con la condición para habilitar el sistema, si está lista se indicada el estado a través del Led “Logical OR”.

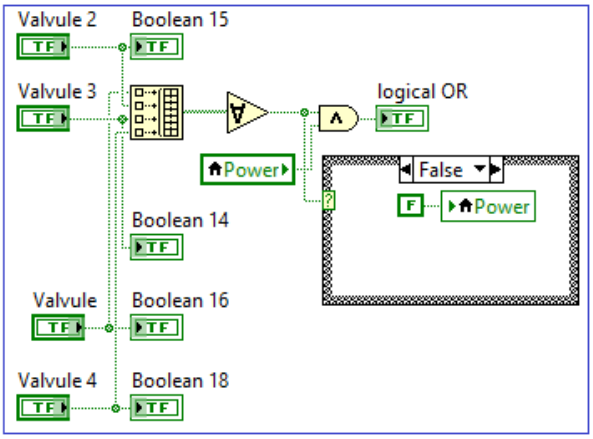

Figura 47. Bloques de control de válvulas

Fuente: LabVIEW 2015

Se utilizan los siguientes interruptores que permiten el flujo de datos a través de los selectores ubicados en el Control & Simulation Loop. El primer interruptor es “RESET” que ingresa a un selector que elige entre dos valores 0.02 y un valor que se mantiene almacenado del feedback que es el último dato que ingresó a la función de transferencia el siguiente selector es para el interruptor “Manual” que permite el paso

entre los valores de la perilla que se encuentra en el panel frontal con el nombre Manual y el bloque analógico de myRIO (Figura 48, Figura 49) que entrega todas las señales enviadas del PLC Siemens S7-1200. Para finalizar se tiene un interruptor “Power” que es el último paso para ingresar a la función de transferencia y se escoge entre los valores de los selectores “RESET” y “MANUAL”. Cada interruptor tiene un led indicador, el estado del “Interruptor” va hacia las entradas del módulo LED de la myRIO (Figura 48) y se los puede identificar con los LEDs “Manual LED” y “Automatico”.

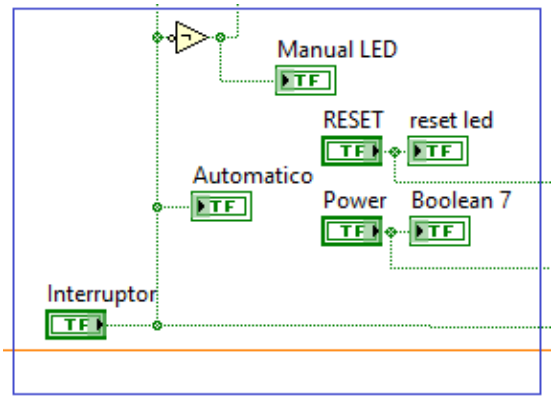

Figura 48. Bloques de interruptores

Fuente: LabVIEW 2015

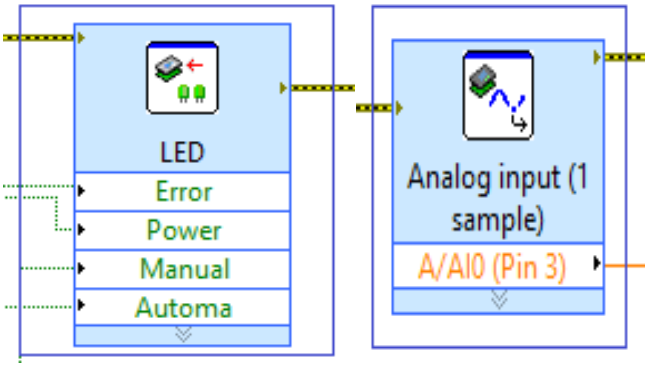

Figura 49. Bloques de control myRIO

Fuente: LabVIEW 2015

Uno de los indicadores principales son los estados de las alarmas enviadas por el PLC Siemens S7-1200, las señales son enviados a la NI myRIO y son extraídas por el módulo MXP para utilizarlas por el software LabVIEW con la herramienta de entradas digitales

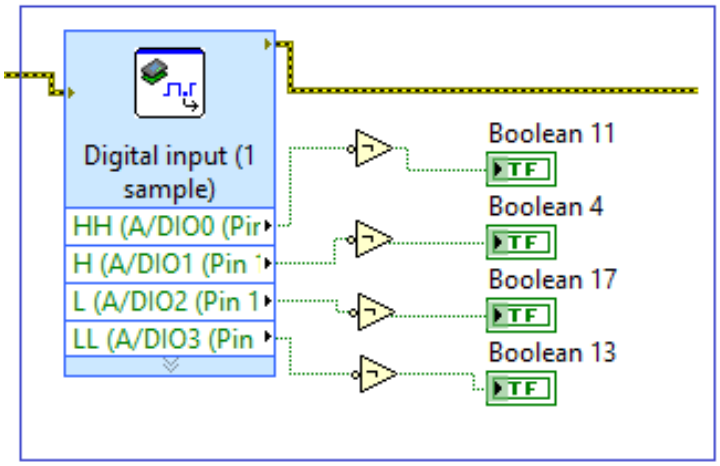

Figura 50. Bloques de estado de las alarmas

Fuente: LabVIEW 2015

La siguiente etapa permite mostrar la información del sistema en la interfaz, esto se maneja dentro del panel frontal en una sección ubicada a la derecha donde se indica el bloque de registro y las advertencias.

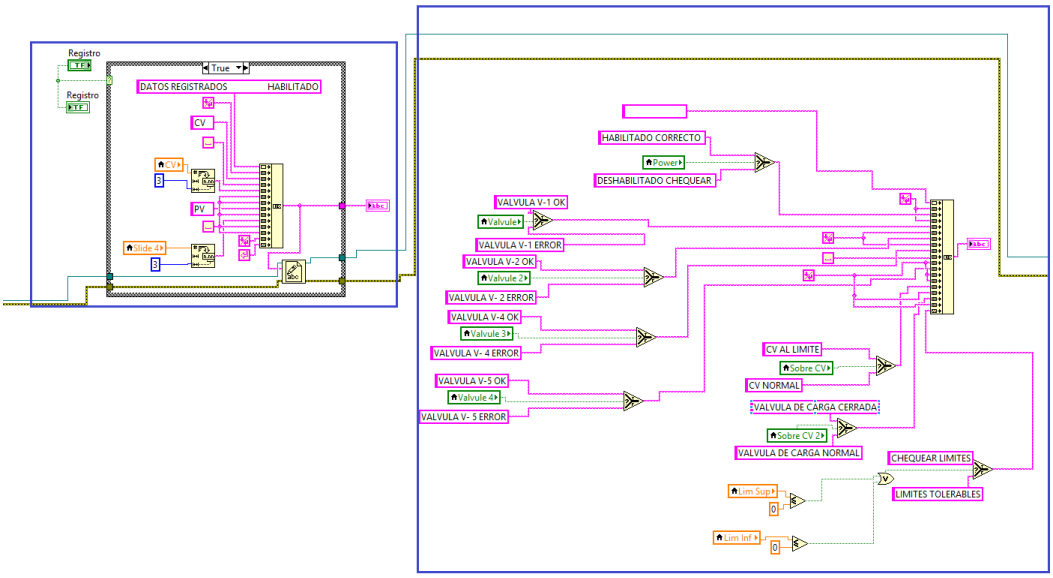

Figura 51. Bloque de impresión de la información

Fuente: LabVIEW 2015

Aquí se maneja un registro de datos elaborado en un documento ".txt" el siguiente bloque se construyó con una estructura case activado por el interruptor "Registro", aquí se realiza la concatenación de los datos tipos strings, se agrega más información

con los tipos numéricos que son los datos acondicionados para mostrar los valores de Procces Value y Control Value.

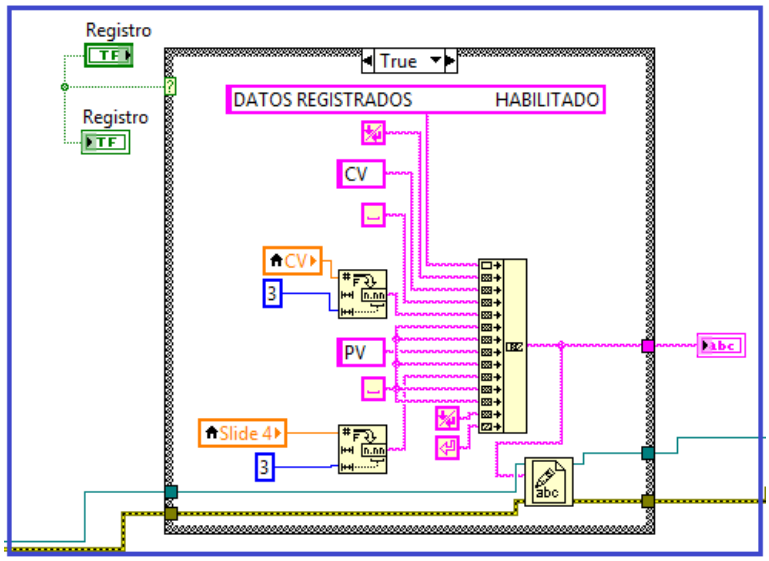

Figura 52. Registro de datos.

Fuente: LabVIEW 2015

En el panel frontal se designa un espacio para las advertencias, que son aquellos mensajes que están interrelacionados con el funcionamiento de la planta. Las plantas de Nivel y Caudal manejan el mismo tipo de válvula en la sección de advertencias y se coloca un mensaje con cada estado, se puede verificar que es correcto cuando se encienden los leds con el color verde, caso contrario se mantendrán apagados, una condición para activar la planta es el correcto accionamiento de las válvulas, como resultado se podrá activar el interruptor de encendido de la planta lo cual genera el encendido del led y su mensaje correspondiente en la sección de advertencias. Se coloca mensajes e indicadores con respecto al funcionamiento de las válvulas de carga y al valor del Control Value, para cuando llegue al límite permitido por la planta se enciende un led de color rojo y se imprime un mensaje de verificación para revisar el estado de la carga o Control Value. Finalmente el último mensaje inspecciona los límites superior e inferior de la planta manteniendo la consigna que deben ser mayor a 0.

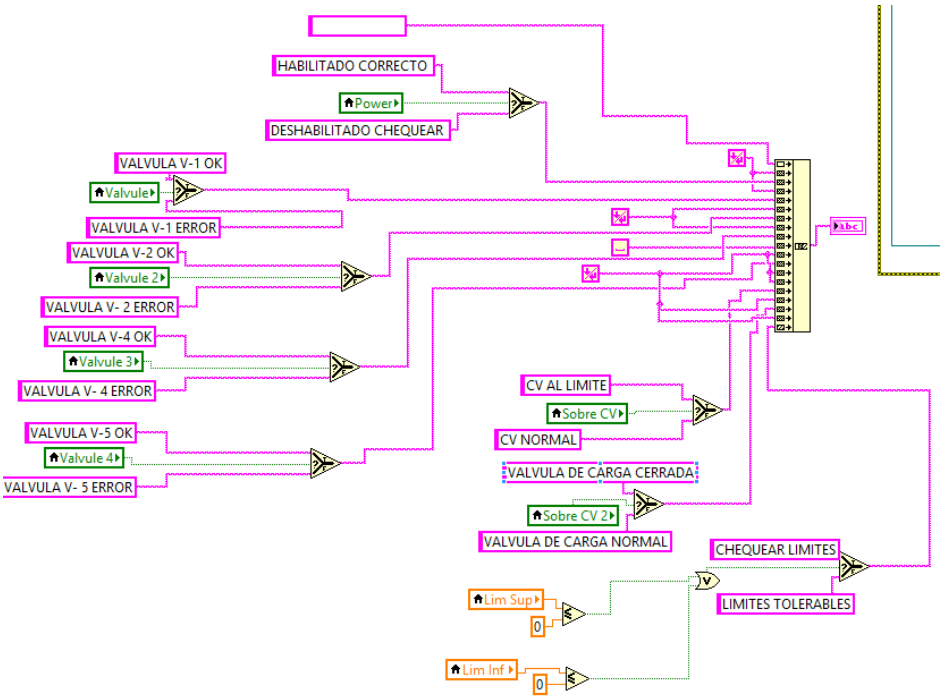

Figura 53. Bloques para imprimir los mensajes de advertencias

Fuente: LabVIEW 2015

En la interfaz del panel frontal se tiene recuadros especiales que permiten saber si los valores de Control Value y la carga están dentro de los rangos tolerables o se encuentran al máximo, si llegaran a estar al límite se activan los mensajes con el error correspondiente, esto con la intención de generar un aviso para tomar una acción correctiva. Para el proceso de Nivel se cuenta con un mensaje que permite verificar el estado del tanque e indica un desborde o vacío del mismo que puede ser perjudicial para el proceso de nivel en el mundo real.

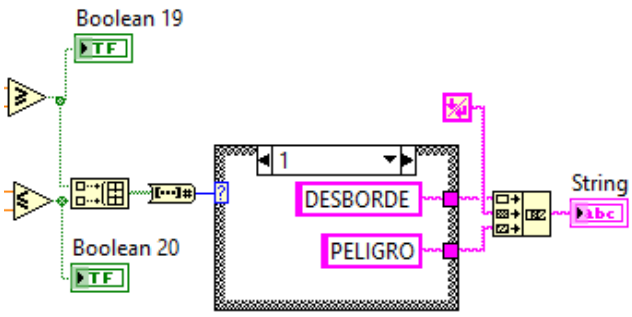

Figura 54. Mensajes adicionales para la interfaz.

Fuente: LabVIEW 2015

La siguiente etapa indica el trabajo realizado con la base del esquema básico utilizando el flujo de datos de los interruptores “RESET”, “Power” y “Manual”, se agregó el modelo de la planta a través de bloques “Transfer Function” y ”Transport Delay” que son parte del toolkit Control & Simulation (Figura 55), con el fin de trabajar con un entorno más real se agregó ciertas condiciones que aproximan la simulación a valores semejantes al proceso real.

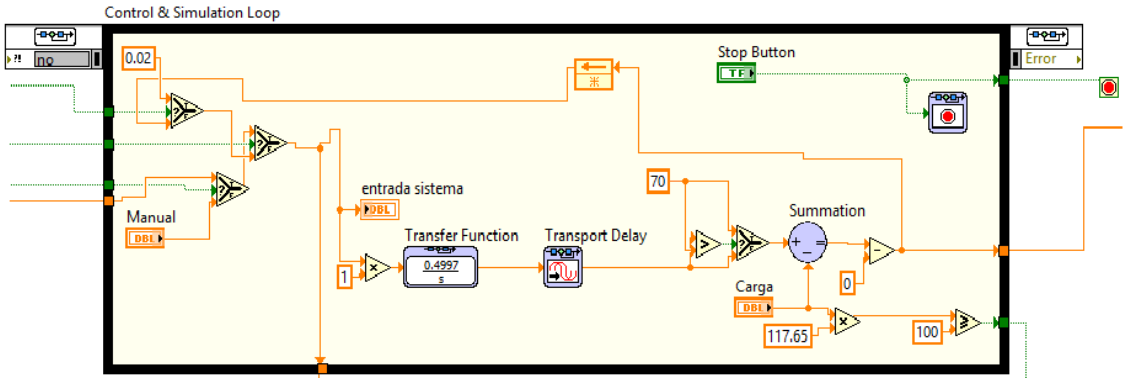

Figura 55. Esquema básico con flujo de datos.
Fuente: LabVIEW 2015

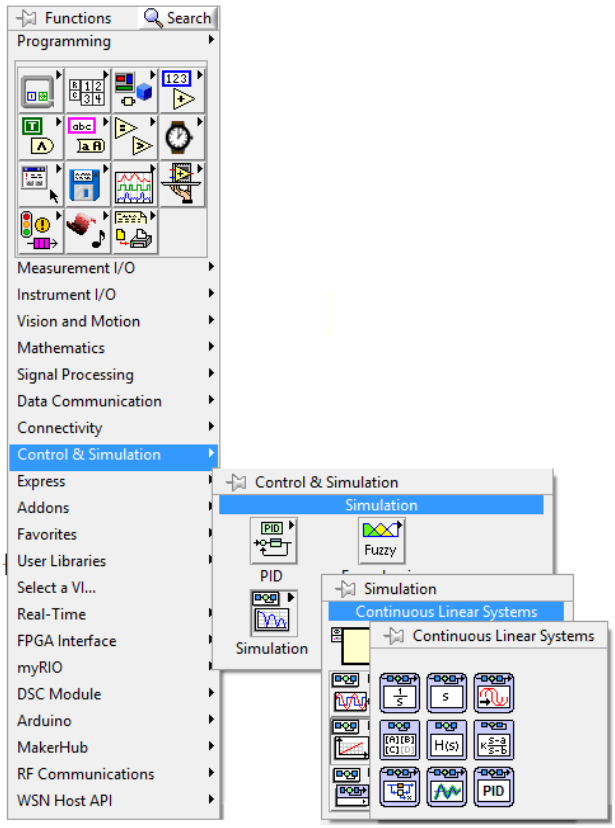

Figura 56. Toolkit Control & Simulation.

Fuente: LabVIEW 2015

La etapa de animación consiste en la construcción de todos los bloques de control e indicadores que formen parte de la interfaz y permitan visualmente una semejanza con los instrumentos utilizados en el mundo real. Una característica del proyecto para los procesos de nivel y caudal, es la intermitencia que se puede visualizar en las tuberías y permiten identificar un paso de agua. Para el proceso de caudal dependiendo del paso de agua cambiará la velocidad de intermitencia del color de las tuberías simulando el flujo y de forma simultánea varía el nivel del tanque almacenador de agua. En el proceso de nivel el paso de agua va determinado el nivel necesario por el tanque, al igual que el caudal el paso de agua del tanque contenedor varía para crear el correcto efecto visual de las plantas.

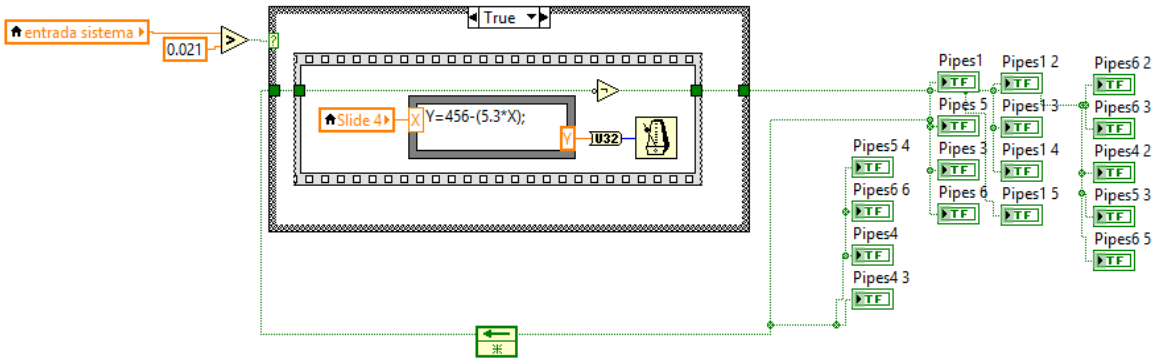

Figura 57. Bloque para la intermitencia en las tuberías

Fuente: LabVIEW 2015

La última etapa es el acondicionamiento realizado para cada uno de los indicadores. Realizando un ajuste a las salida de la estructura Control & Simulation Loop para que los valores se mantengan en un rango entre 0 y 5 estos serán los valores para acondicionar y que serán enviados por el bloque de salida analógico de la NI myRIO a el PLC Siemens S7-1200.

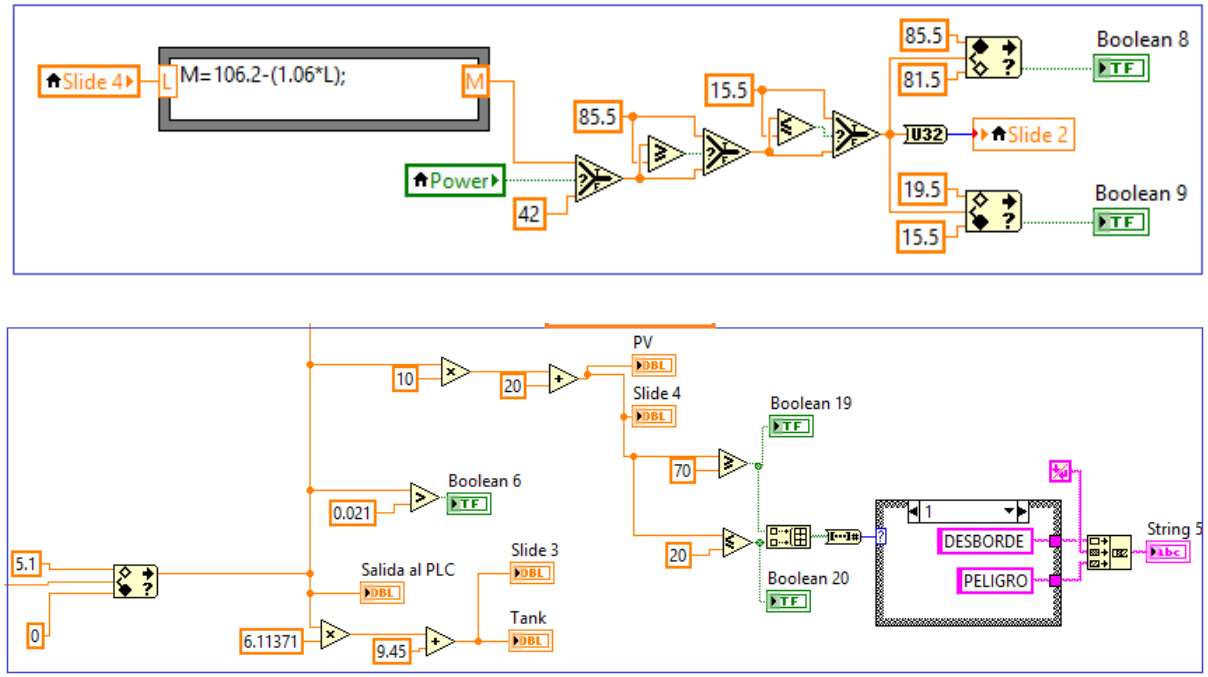

Figura 58. Bloques de acondicionamientos de la interfaz***.***

Fuente: LabVIEW 2015

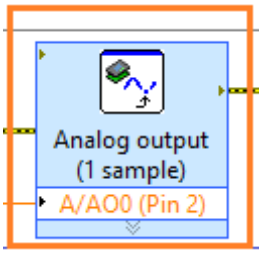

Figura 59. Bloque de salidas analógicas de NI myRIO.

Fuente: LabVIEW 2015

Para el último frame se maneja el error del módulo de myRIO, con un bloque que permite reiniciar el dispositivo una vez finalizado el segundo frame y se selecciona la función para cerrar el archivo “.txt”.

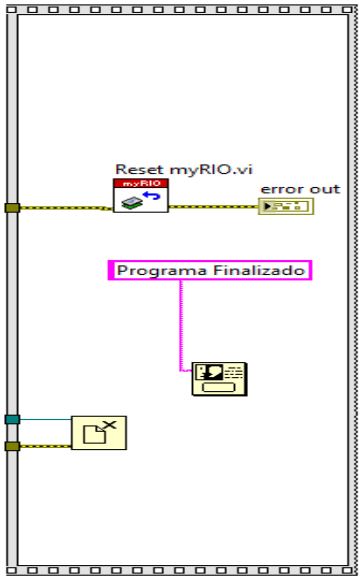

Figura 60. Frame de finalización.

Fuente: LabVIEW 2015

CAPÍTULO IV

4. CONTROLADOR

En el presente capítulo se hará una descripción del diseño e implementación de un controlador con acción Proporcional, Integral y Derivativa (PID) aplicado a los Sistemas de Control de Caudal, Nivel y Presión. La Teoría de control propone que no se puede controlar lo que no se pude medir y peor aún controlar algo que se desconoce o algo de lo que no se sabe su comportamiento por lo que es necesario empezar obteniendo la función de transferencia para cada uno de los Sistemas de Control de Procesos, este paso se lo consiguió en el capítulo anterior. Los mencionados Sistemas son de primero y segundo orden además que el sistema de Nivel es un proceso de tipo integrante, para todos ellos se va a aplicar tres diferentes métodos de sintonía para obtener el mejor desempeño del controlador. Adicionalmente se explica la programación en el software propio del Controlador Lógico Programable Siemens S7-1200 y la generación de una Interfaz Humano Máquina mediante NI LabVIEW que permita al usuario visualizar gráficamente la respuesta del sistema al valor de la variable deseada por el usuario.

4.1. Sistema de Control de Flujo

Para el sistema de Control de Flujo se aplican los métodos de Ziegler-Nichols en lazo abierto, el método Lambda y el método de Cohen-Coon.

4.1.1. Modelo Matemático

El Sistema de Control de Flujo por su configuración es de primer orden más tiempo muerto además es un proceso autorregulado, y donde su función de transferencia característica es:

$$G(s)_{embebido} = \frac{1.9211}{1.2202s+1} e^{-1.4102} \tag{4.1}$$

Donde $Ks = 1.9121$ es la ganancia propia del sistema $L = 1.4102$ representa al tiempo muerto y $T = 1.2202$ es la constante de tiempo del sistema.

4.1.2. Diseño del Controlador PID mediante métodos analíticos

Para obtener un funcionamiento óptimo del controlador es necesario realizar pruebas y ajustes en lo que se refiere a sintonía, para lo cual se realiza la obtención de los parámetros del PID mediante distintos métodos de sintonía, para posteriormente comparar la respuesta del sistema HIL al aplicarse cada método y determinar el más óptimo que contribuya al mejor desempeño del Controlador, ver sección 5.2.

- **El método de Ziegler-Nichols en lazo abierto –** Para el Sistema de Caudal se consideró este método porque además de ser bastante simple, a la vez aporta buenas características al control presentando sobreimpulso máximo del 20% por lo cual es ideal para los procesos a controlar en este trabajo. Para aplicar el método de Ziegler-Nichols, en lazo abierto, es necesario reemplazar el controlador PID por una entrada escalón como señal de excitación del sistema véase en la Figura 61.

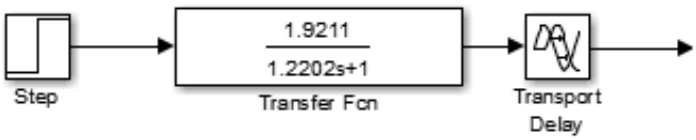

Figura 61. Configuración de control en lazo abierto

Cuando se excita el sistema mediante una entrada escalón, este empieza a reaccionar a este estímulo generando una gráfica resultante donde es posible obtener los parámetros necesarios para aplicar las ecuaciones propuestas en la tabla de sintonía propia del método y así obtener los parámetros para el controlador PID. De la gráfica resultante se determina aproximadamente que:

- $L = 1.8$ segundos
- $T = 1.94$ segundos

Después de obtener estos parámetros se debe dirigir a la Tabla 3 para calcular los valores de los parámetros del PID.

Tabla 2.

Cálculo de Parámetros del PID en Lazo Abierto mediante Ziegler-Nichols

Tipo de Controlador	Kp	Ki	Kd
P	$\frac{T}{L}$	0	0
PI	$0.9 * \frac{T}{L}$	$\frac{0.3}{L}$	0
PID	$1.2 * \frac{T}{L}$	$\frac{1}{2L}$	$0.5\,L$

Fuente: (Ogata, 2003)

Se realizó el cálculo correspondiente de los parámetros para el Controlador PID, y posteriormente se realizó una sintonía fina. Los parámetros obtenidos para la sintonización del controlador fueron $Kp = 0.82597$, $Ti = 4.1346$, $Td = 0$, siendo importante tomar muy en cuenta la configuración del controlador si se toman como Tiempo Integral (Ti) y Tiempo derivativo (Td) o Ganancia Integral (Ki) y Ganancia Derivativa (Kd), dependiendo de esto se debe considerar las relaciones:

$$Ti = \frac{Kp}{Ki} \tag{4.2}$$

$$Td = \frac{Kp}{Kd} \tag{4.3}$$

- **El método Lambda –** El método de sintonía Lambda es muy parecido a Ziegler-Nichols con la diferencia de que posee una acción integral más fuerte lo que permite corregir de una mejor manera el error aunque sacrifica el tiempo de establecimiento. Con los valores obtenidos de $L\,y\,T$ de la respuesta al escalón se obtienen los parámetros para el controlador mediante la Tabla 4.

Tabla 3.
Cálculo de Parámetros del PID mediante el método Lambda

Tipo de Controlador	Kp	Ti	Td
PID	$\frac{1}{k}\left(\frac{\frac{L}{2}+T}{\frac{L}{2}+3T}\right)$	$T+\frac{L}{2}$	$\frac{T*L}{L+2T}$

Fuente: (Ogata, 2003)

Se realizó el cálculo correspondiente de los parámetros para el Controlador PID, y posteriormente se realizó una sintonía fina. Los parámetros obtenidos para la sintonización del controlador fueron $Kp = 0.2199$, $Ti = 2.84$, $Td = 0.6147$.

- **El método de Cohen-Coon –** Se diferencia de los otros dos métodos ya que su criterio principal de diseño es el rechazo de las perturbaciones de carga, igualmente establece una tabla de valores basándose en cálculos analíticos y numéricos, para esto se considera primero las siguientes ecuaciones:

$$a=\frac{KL}{T} \tag{4.4}$$

$$\tau=\frac{L}{L+T} \tag{4.5}$$

Con los datos de $L = 1.8$ segundos y $T = 1.94$ segundos, los valores de $a\ y\ \tau$ son:

$$a = 1.7824$$

$$\tau = 0.4812$$

A través de estos parámetros se puede utilizar la siguiente tabla para encontrar las ganancias del PID.

Tabla 4.
Cálculo de Parámetros del PID mediante el método de Cohen-Coon

Tipo de Controlador	aKp	Ki	Kd
P	$1+\frac{0.35\,\tau}{1-\tau}$	∞	0
PI	$0{,}9*\left(1+\frac{0.092\tau}{1-\tau}\right)$	$\frac{3.3-3\tau}{1+1.2\tau}$	0
PID	$1{,}35*\left(1+\frac{0.18\tau}{1-\tau}\right)$	$\frac{2.5-2\tau}{1-0.39\tau}$	$\frac{0.37-0.37\tau}{1-0.81\tau}$

Fuente: (Ogata, 2003)

Después de realizar el cálculo correspondiente de los parámetros para el Controlador PID, y a través de una sintonía fina. Los parámetros obtenidos para la sintonización del controlador fueron $Kp = 0.5479$, $Ti = 1.1768$, $Td = 0$.

4.1. Sistema de Control de Nivel

Para el sistema de Control de Nivel se aplican los métodos de Ziegler-Nichols en lazo cerrado, el método de Chien, Hrones y Reswick y el método de Cohen-Coon.

4.2.1. Modelo matemático

El Sistema de Control de Nivel por su configuración es de primer orden más tiempo muerto, este sistema es de tipo integrante y su función de transferencia característica es:

$$G(s)_{embebido}=\frac{0.07}{s}e^{-3.6s} \tag{4.6}$$

Donde $Ks = 0.07$ es la ganancia propia del sistema $L = 3.6$ representa al tiempo muerto y $T = 1$ es la constante de tiempo del sistema.

4.2.2. Diseño del controlador PID mediante métodos analíticos

Se obtienen los parámetros del controlador PID mediante los tres métodos propuestos y se compara la respuesta del sistema HIL con cada uno de estos.

- **El método de Ziegler-Nichols en lazo cerrado –** Para el proceso de Nivel al ser de tipo integrante es necesario aplicar el método de Ziegler-Nichols en lazo cerrado (Figura 62) para conseguir una oscilación sostenida de donde se puedan extraer los valores necesarios para calcular los parámetros para el PID.

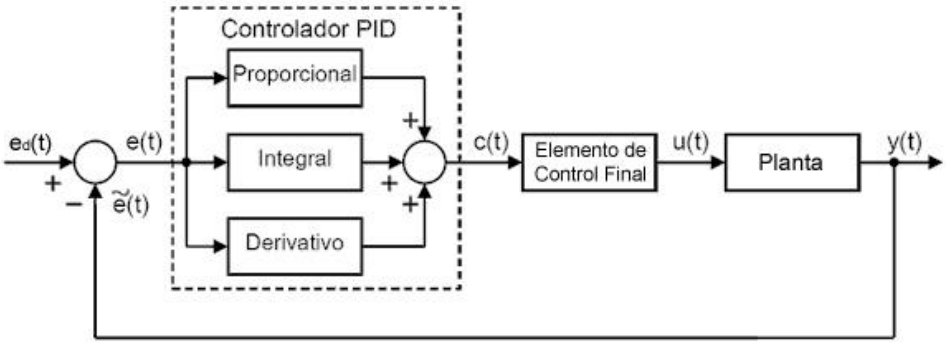

Figura 62. Configuración de control en lazo cerrado

Al tener el controlador ya integrado en el sistema se tiene acceso a los parámetros del PID, por lo que se debe reducir a cero la acción integral y derivativa mientras que se aumenta de a poco la ganancia proporcional esto hace que el sistema empiece a oscilar como se puede ver en la Figura 63, como lo que se busca es que el sistema tenga una oscilación sostenida a cualquier perturbación, se sube la ganancia proporcional hasta lograrlo y cuando se lo consigue se toma este valor como ganancia límite. Además se toma el valor de un periodo de esta señal en segundos, estos parámetros se usan con la tabla que propone Ziegler-Nichols para calcular las ganancias que sintonicen el controlador.

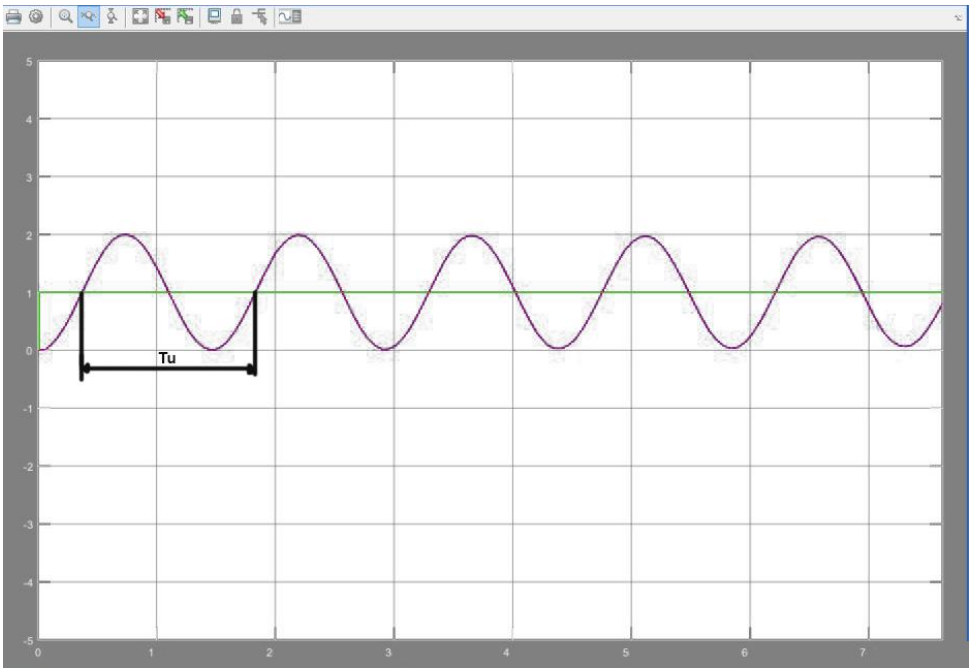

Figura 63. Oscilaciones Sucesivas en Sistema de Control de Nivel
Fuente: MATLAB 2012a

El Sistema de Control de Nivel logra oscilaciones sucesivas con un valor de ganancia de $Kc = 0.0158$ y el tiempo de un periodo en segundos que resulta ser aproximadamente $T = 1.45$, con estos valores es posible recurrir a la Tabla 6 para calcular las ganancias del PID.

Tabla 5.

Cálculo de Parámetros del PID mediante Ziegler-Nichols en lazo cerrado

Tipo de Controlador	Kp	Ti	Td
P	0.50*Kc	0	0
PI	0.45*Kc	$\frac{1.2}{Tu}$	0
PID	0.6*Kc	$\frac{2}{Tu}$	0.125*Tu

Fuente: (Ogata, 2003)

Después de realizar el cálculo correspondiente de los parámetros para el Controlador PID, finalmente se realizó una sintonía fina. Los parámetros obtenidos para la sintonización del controlador fueron $Kp = 7.0519, Ti = 9.4, Td = 3.554$, también es necesario tomar en cuenta la configuración del controlador si se toman como tiempo integral y tiempo derivativo o ganancia integral y ganancia derivativa.

- **El Método de Cohen-Coon –** Para el método de Cohen-Coon se parte de la función de transferencia obteniendo parámetros:

- $Ks = 0.07$
- $L = 3.6$ segundos
- $T = 1$ segundos

Con estos valores se obtienen $a = 0.20958$ y $\tau = 0.80769$ para poder recurrir a la tabla 5 y obtener los parámetros de sintonía para el PID. Después de realizar los cálculos necesarios, y mediante una sintonía fina se logró determinar que los valores de los parámetros del controlador fueron $Kp = 23.989, Ti = 0.4453, Td = 0$.

- **El Método de Chien, Hrones y Reswick –** Se consideró este método ya que propone mejoras en los sistemas de tipo integrantes con cálculos tomando en cuenta un valor de sobreimpulso máximo además de una tabla que permite calcular los parámetros del controlador ante perturbaciones, e igualmente establece una tabla de valores basándose en cálculos analíticos y numéricos. Mediante los datos de la función de transferencia Ks, L y T se recurre a la tabla 7.

Tabla 6.
Cálculo de parámetros PID mediante Chien, Hrones y Reswick sin perturbaciones

Tipo de Controlador	Sin Sobreoscilación			20% Sobreoscilación		
	aKp	Ki	Kd	aKp	Ki	Kd
P	$\frac{0.3}{T*L}$	∞	0	$\frac{0.7}{T*L}$	∞	0
PI	$\frac{0.35}{T*L}$	$1.2L$	0	$\frac{0.6}{T*L}$	L	0
PID	$\frac{0.6}{T*L}$	L	$0.5*L$	$\frac{0.95}{T*L}$	$1.4*L$	$0.47*L$

Fuente: (Ogata, 2003)

Tabla 7.
Cálculo de parámetros PID mediante Chien, Hrones y Reswick con perturbaciones

Tipo de Controlador	Sin Sobreoscilación			20% Sobreoscilación		
	Kp	Ki	Kd	Kp	Ki	Kd
P	$\frac{0.3}{T*L}$	∞	0	$\frac{0.7}{T*L}$	∞	0
PI	$\frac{0.6}{T*L}$	$4*L$	0	$\frac{0.7}{T*L}$	$2.3*L$	0
PID	$\frac{0.95}{T*L}$	$2.4*L$	$0.42*L$	$\frac{1.2}{T*L}$	$2*L$	$0.42*L$

Fuente: (Ogata, 2003)

Al ser un sistema donde se va a trabajar con perturbaciones se realizan los cálculos de los parámetros con la Tabla 8 y después de realizar una sintonía fina se obtuvo los parámetros $Kp = 0.2638$, $Ti = 8.64$, $Td = 1.512$ para la sintonización del controlador y se debe considerar si la acción integral y la acción derivativa son tomadas como tiempos o ganancias.

4.3. Sistema de Control de Presión

Para el Sistema de Control de Presión, puesto que tiene mayor grado de complejidad cuando se desea obtener los parámetros para el controlador PID, solo se aplica el método de Ziegler-Nichols en lazo cerrado por ser práctico y simple. Es posible aproximar el sistema de segundo orden a uno de primero para aplicar diferentes métodos de sintonía pero baja el porcentaje de aceptación de la función de transferencia y puede llevar a errores en el funcionamiento del sistema HIL.

4.3.1. Modelo Matemático

El Sistema de Control de Presión por su configuración tiene una función de transferencia de segundo orden más tiempo muerto, este sistema es de tipo autorregulado y su función de transferencia característica es:

$$G(s)_{embebido}=\frac{25.102}{(70.8452s^2+8.96069s+1)}e^{-1.02} \tag{4.7}$$

4.3.2. Diseño del controlador PID mediante métodos analíticos

- **El Método de Ziegler-Nichols en lazo cerrado –** Se trabaja como con un sistema de primer orden a las constantes Ti y Td se les asigna el valor de cero y a la ganancia proporcional Kp se le asigna un valor que permita al sistema tener una oscilación sostenida, si el sistema no llega a tener una oscilación sostenida, este método no es aplicable. Cuando se logra que el sistema oscile se toma el valor de la última ganancia $Ku = 120$, y el valor del periodo $Pu = 2.094$ segundos, con estos valores se calculan los parámetros del controlador a través de la Tabla 6. Después de calcular los parámetros mediante la tabla y aplicar una sintonía fina se obtuvieron los valores de $Kp = 0.45$, $Ti = 11.014$ y $Td = 0$.
- **Mediante la función de transferencia –** Si se dispone de la función de transferencia el proceso se facilita aunque aquí se debe encontrar la última ganancia y el valor de ese periodo a través de los parámetros que proporciona la ecuación de la siguiente forma:

Se empieza cambiando en el denominador "s" por "jw" en la función de transferencia, el término que contenga "jw" de forma simple en el denominador permite obtener la frecuencia "Wn" aplicando:

$$\omega n = \mp j\sqrt{\omega} \tag{4.8}$$

ωn es la base para los demás parámetros. El último periodo se puede calcular mediante la fórmula:

$$Pu = \frac{2\pi}{\omega n} \tag{4.9}$$

Y de la parte real se calcula la última ganancia reemplazando de la siguiente manera:

$$(jw)^2 + ({\omega_n}^2 + K) = 0 \tag{4.10}$$

$$K_u = -{\omega_n}^2 - j\omega^2 \tag{4.11}$$

Mediante la tabla se ajustan los parámetros del controlador.

Tabla 8.

Parámetros de Ajuste del Controlador mediante la función de transferencia

Tipo de Controlador	Kp	Ti	Td
P	$0.50 * Kc$		
PI	$0.45 * Kc$	$\frac{Pu}{1.2}$	
PID	$0.6 * Kc$	$\frac{Pu}{2}$	$\frac{Pu}{8}$

Fuente: (Ogata, 2003)

Después de realizar los cálculos necesarios se obtuvieron los parámetros para el controlador y fueron $Kp = 0.1326$, $Ti = 71.6531$ y $Td = 0.5218$.

4.4. Implementación de Control PID

El PLC Siemens S7-1200 tiene la capacidad de ser configurado como un controlador de tipo PID y se programa directamente en el software TIA Portal de Siemens, el código de programación es similar para los tres sistemas a ser

controlados. La señal al controlador llega desde el dispositivo embebido que emula la planta como un dato analógico, por lo que el PLC también ocupa una entrada del mismo tipo para tomar los datos. En el ítem bloques de programa es donde se inserta el bloque *PID Compact* ubicado en el menú de instrucciones.

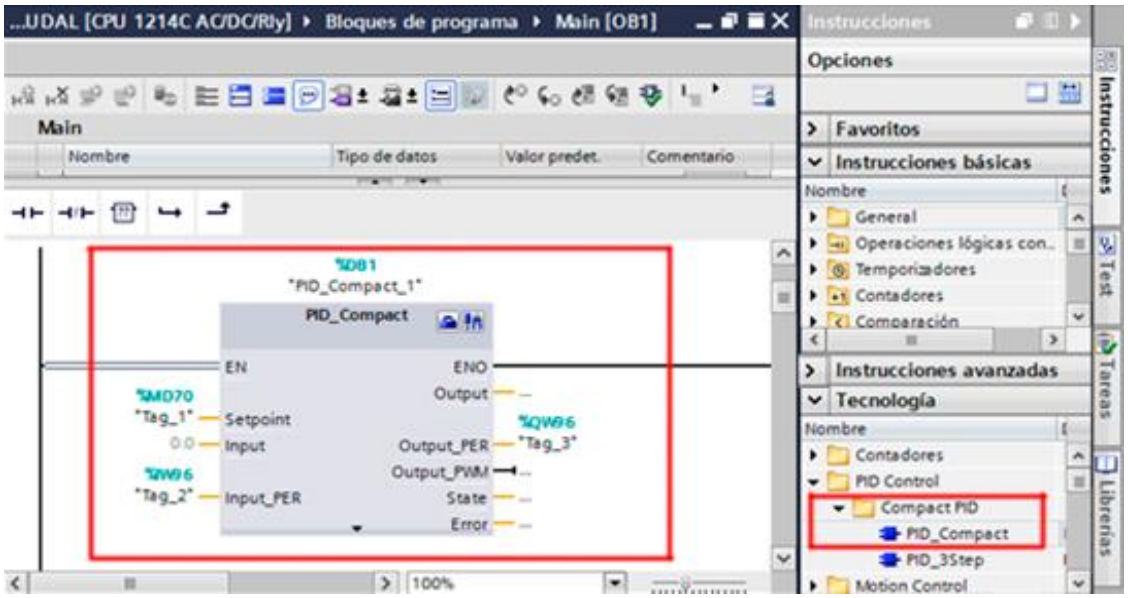

Figura 64. Bloque PID Compact
Fuente: TIA Portal V13

En el bloque PID se utilizan tres variables, dos de entrada y una de salida de datos.

- *Setpoint* es una entrada donde ingresa el valor que el usuario desea que alcance la variable controlada, este dato llega desde la Interfaz Humano Máquina,

- A la entrada *Input_PER* llega el valor enviado desde el sistema embebido, se toma esta entrada porque acepta datos analógicos

- La salida *Output_PER* envía datos de tipo analógico desde el controlador, esta señal es la que se encarga de forzar al sistema a realizar una acción correctiva en la respuesta del sistema.

4.1.2. Configuración PID

El bloque tiene una configuración interna y es aquí donde se diferencia para cada Sistema de Control de Proceso. Es posible ingresar a esta configuración desde el Árbol del proyecto o desde el mismo bloque en la parte superior derecha del bloque en el programa Figura 65.

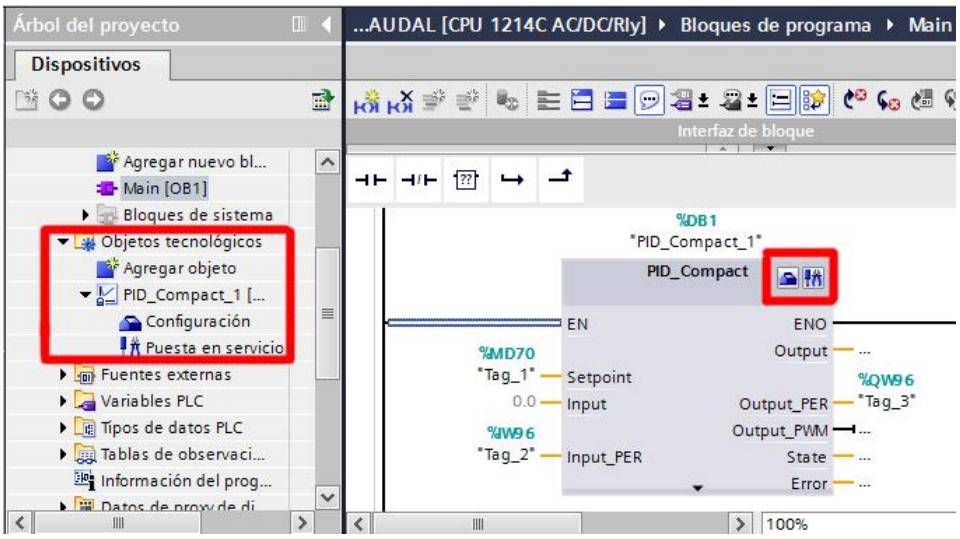

Figura 65. Menú bloque PID

Fuente: TIA Portal V13

La configuración del Bloque PID se divide en tres secciones; en ajustes básicos se ingresa el tipo de variable con la que se va a trabajar para este caso Caudal y sus unidades, además de la asignación de las entradas y salidas del bloque. En el ítem de Ajustes del valor real se ingresa el máximo valor al que puede llegar el proceso y el escalamiento de la variable de entrada con el rango de valores analógicos que maneja el PLC; los valores que se emplean en el escalamiento dependen de cada proceso y rangos de valores en los que se encuentra calibrado el transmisor. En la sección de los ajustes avanzados se configuran los límites de la señal de salida del controlador y los parámetros del PID donde se encuentran las constantes de sintonía que se ingresan desde la interfaz Humano Máquina diseñada para este Sistema.

Desde la herramienta de puesta de servicio del bloque PID se “inicia” el proceso de control automático, la gráfica de las señales de entrada y salida así como también el valor de los parámetros Setpoint, Input_PER y Output_PER. Para tomar los valores que ingresan y que salen del sistema embebido es necesario normalizar los valores y posteriormente escalarlos de acuerdo a los rangos de cada proceso (Figura 66), de esta manera es posible tomar estos valores mediante un servidor OPC para poder visualizarlos en una Interfaz más amigable al usuario en LabView.

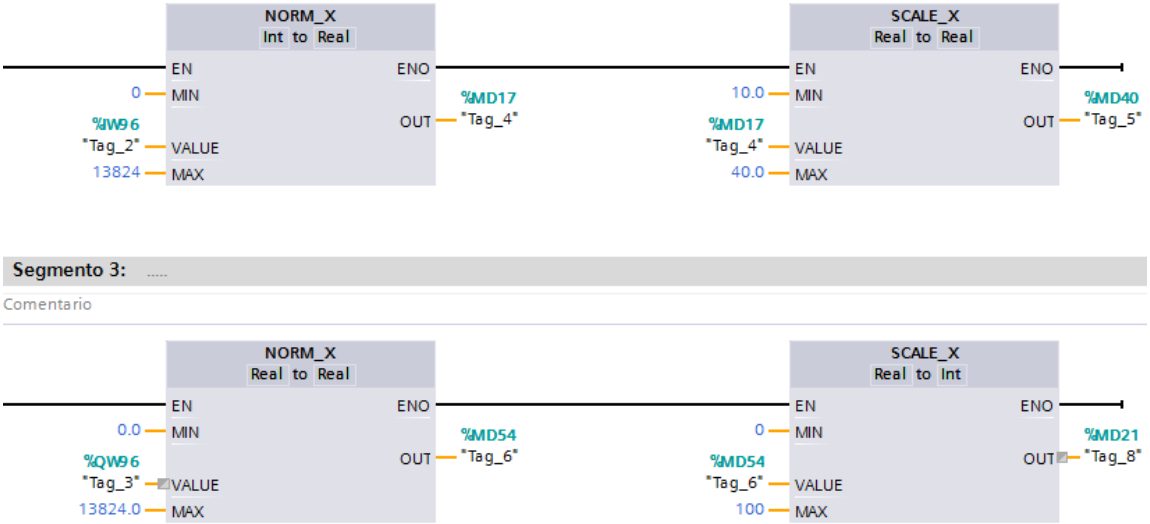

Figura 66.Normalizar y Escalar entradas y salidas del Sistema

Fuente: TIA Portal V13

CAPÍTULO V

5. ANÁLISIS Y RESULTADOS

5.1. Validación de los sistemas embebidos

La validación de los sistemas embebidos se puede conseguir por medio de dos parámetros:

- **Gráficas de respuesta –** Es posible validar el sistema embebido obteniendo la dinámica del mismo y comparando su gráfica con la que se obtiene de la dinámica del proceso real, para esto se debe alimentar al sistema con una entrada escalón que haga posible ver claramente la señal de respuesta y sus parámetros como el tiempo muerto, el tiempo de establecimiento de la señal de respuesta y sobre impulso, de acuerdo a la similitud de estas gráficas se determina si el Sistema Embebido trabaja de igual forma que el Sistema Real y puede ser utilizado dentro de la configuración HIL. Para facilitar la comparación lo más óptimo es graficar en un archivo de Excel la respuesta de los sistemas mediante el histórico de datos (Figura 74, Figura 75, Figura 76).

- **Función de Transferencia –** Para validar el sistema embebido mediante un análisis matemático, es necesario obtener la dinámica de la planta embebida y crear un histórico de datos a través de su gráfica, mediante las herramientas de MATLAB es posible generar la función de transferencia de las plantas embebidas de los Sistemas de Control de Procesos, y por último con la herramienta *ident* de Matlab se determina el porcentaje de aceptación, el cual si es mayor al 80% es aceptable y se puede determinar que el Sistema embebido es válido para ser utilizado dentro del entorno HIL.

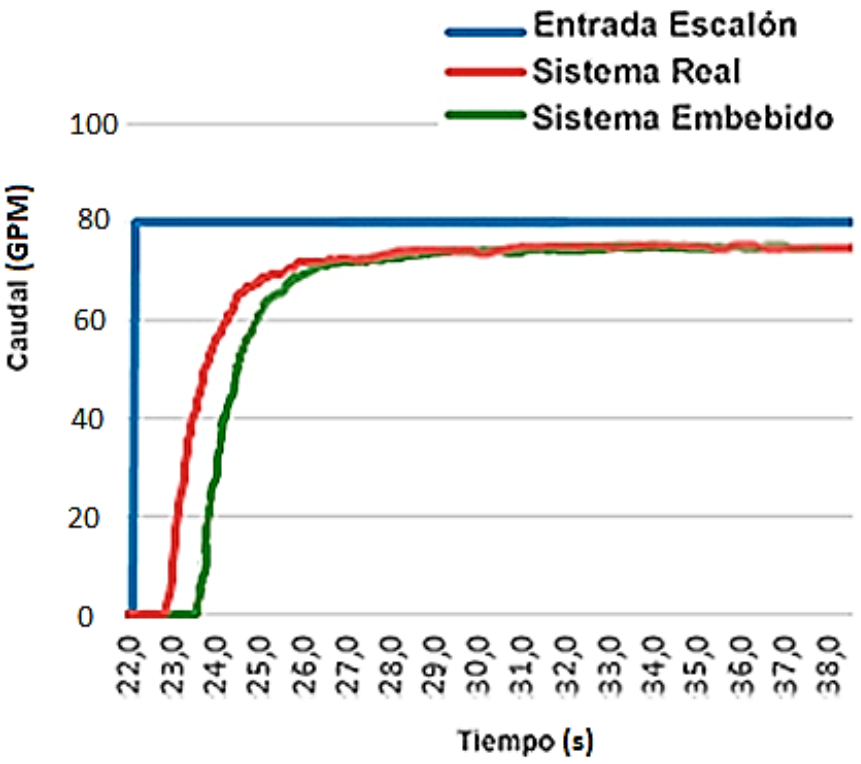

Figura 67. Respuesta Sistema Real y Sistema Embebido de Caudal

Fuente: Microsoft Excel 2013

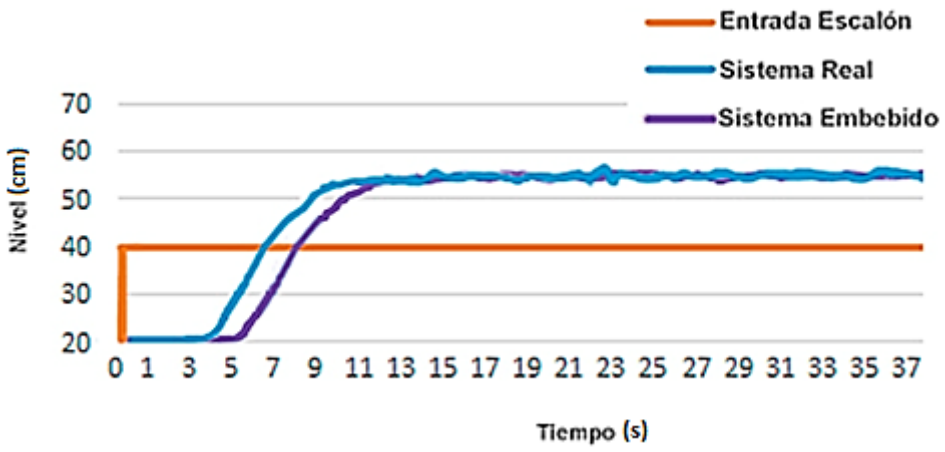

Figura 68. Respuesta Sistema Real y Sistema Embebido de Nivel

Fuente: Microsoft Excel 2013

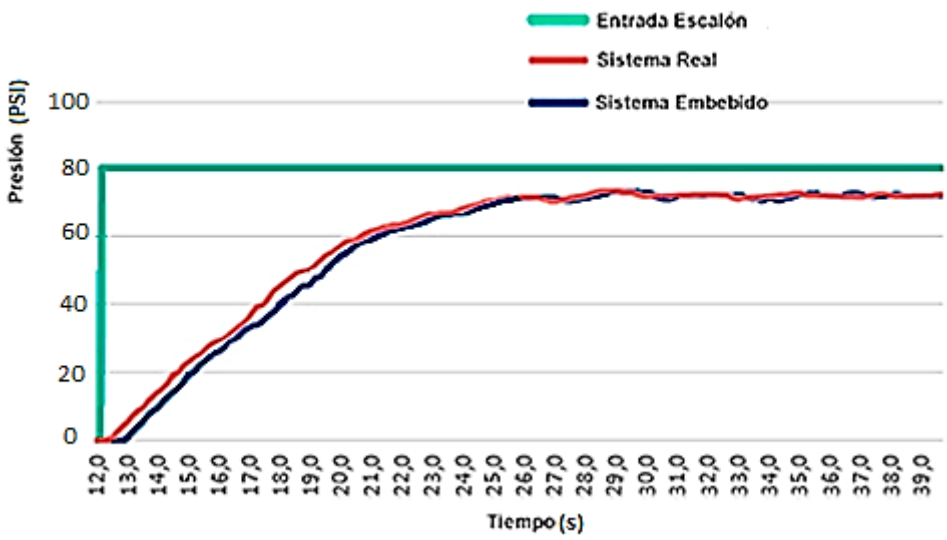

Figura 69. Respuesta Sistema Real y Sistema Embebido de Presión

Fuente: Microsoft Excel 2013

- **Caudal -** Para el Sistema de Control de Caudal es posible obtener la función de transferencia del Sistema embebido a través de la herramienta ident de Matlab y el resultado se presenta a continuación:

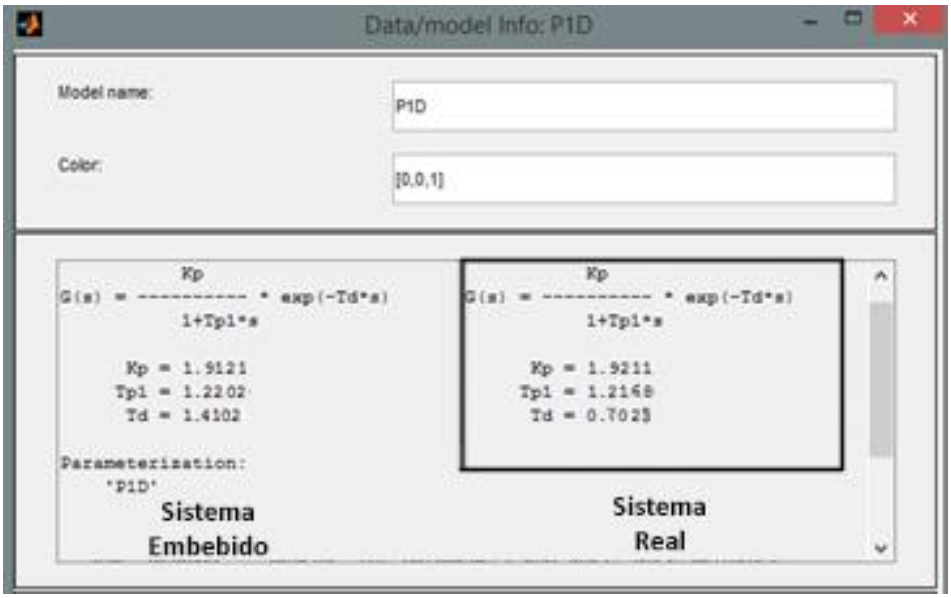

Figura 70.Función de Transferencia Sistema Embebido y Sistema Real de Caudal

Fuente: Matlab 2012a

Las Funciones de transferencia del Sistema Real y el Sistema Embebido de Caudal son las siguientes:

$$G(s)_{real} = \frac{1.9211}{1.2168s+1} e^{-0.7023s} \tag{5.1}$$

$$G(s)_{embebido} = \frac{1.9121}{1.2202s+1} e^{-1.4102s} \tag{5.2}$$

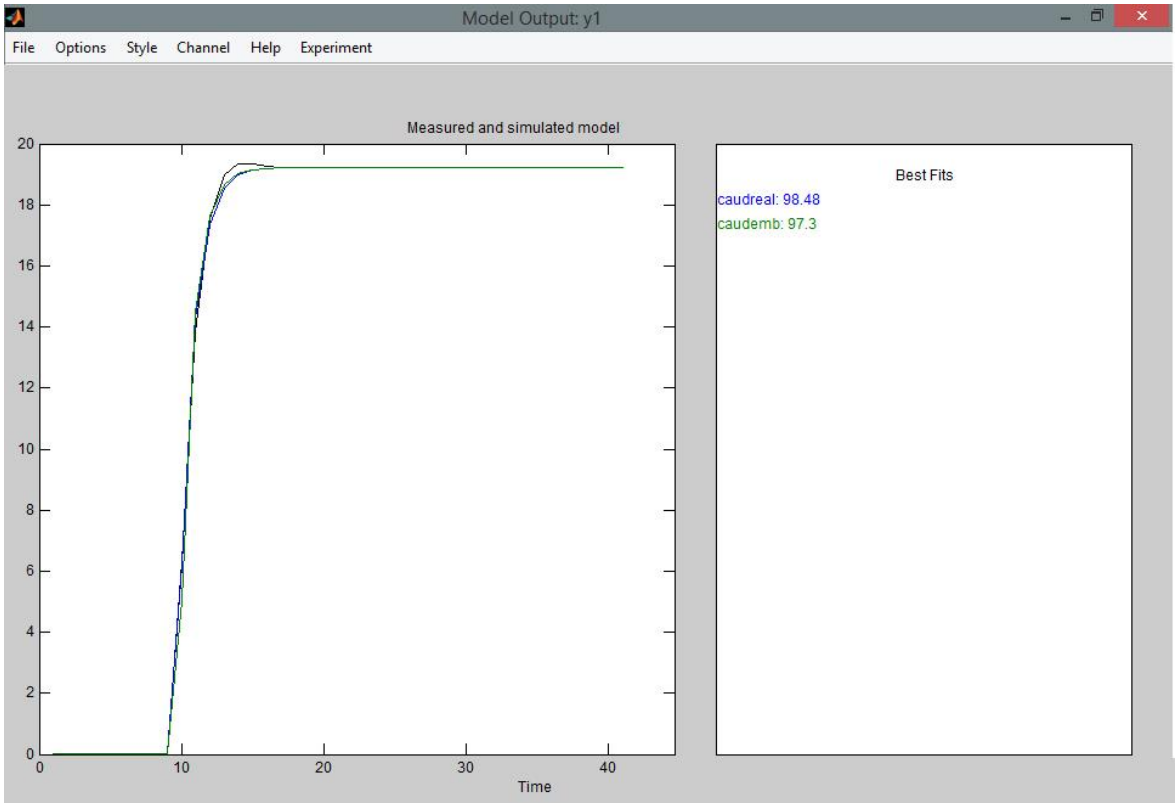

Figura 71. Porcentaje de aceptación de las funciones de transferencia del Sistema Real y el Sistema Embebido de Caudal

Fuente: Matlab 2012a

La Función de transferencia del Sistema Real presenta un porcentaje de aceptación del 98.48% mientras que el Sistema embebido presenta una función de transferencia con un porcentaje de aceptación del 97.3% lo que hace posible definir que es válido y puede ser utilizado en el entorno HIL.

Para el Sistema de Control de Nivel no es posible encontrar la función de transferencia a través de la herramienta ident por lo cual el procedimiento es de forma manual y gracias a los parámetros que pueden ser extraídos de la gráfica de respuesta ante una entrada escalón. Las Funciones de transferencia del Sistema Real y el Sistema Embebido de Nivel se presentan a continuación:

$$G(s)_{real} = \frac{0.04997}{s} e^{-4.2s} \tag{5.3}$$

$$G(s)_{embebido} = \frac{0.07}{s} e^{-3.6s} \tag{5.4}$$

Mediante sus funciones de transferencia es posible determinar que el error entre las funciones de transferencia del Sistema Real y el Sistema Embebido es muy bajo y se concluye que el Sistema Embebido es válido.

Para el Sistema de Control de Presión si es viable utilizar la función ident de Matlab y los resultados de sus funciones de transferencia se presentan a continuación:

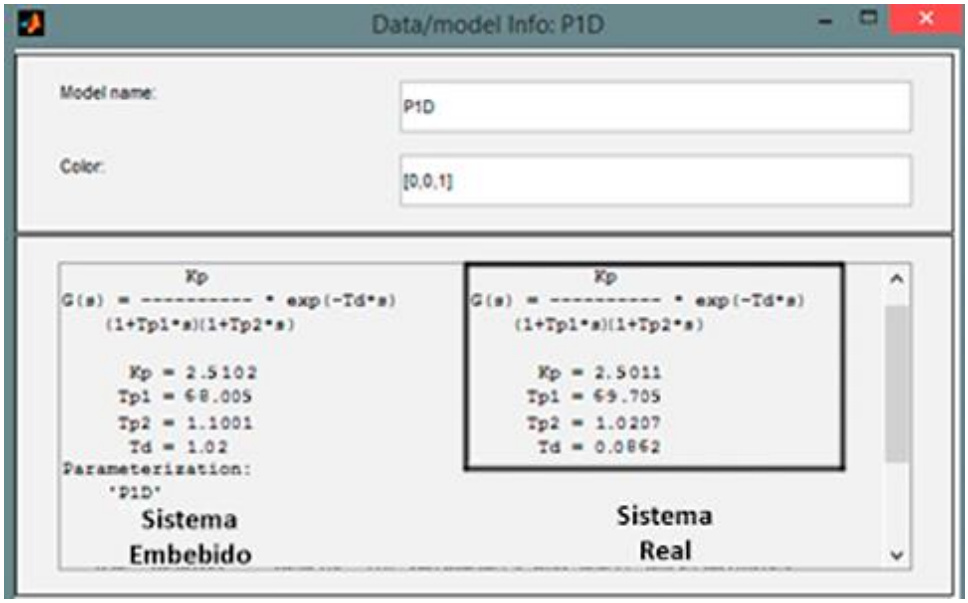

Figura 72. Función de Transferencia Sistema Embebido y Sistema Real de Presión

Fuente Matlab 2012a

Las Funciones de transferencia del Sistema Real y el Sistema Embebido de Presión son las siguientes:

$$G(s)_{real} = \frac{25.011}{(71.1478s^2+9.96069s+1)} e^{-0.686s} \tag{5.5}$$

$$G(s)_{embebido} = \frac{25.102}{(70.8452s^2+8.96069s+1)} e^{-1.02} \tag{5.6}$$

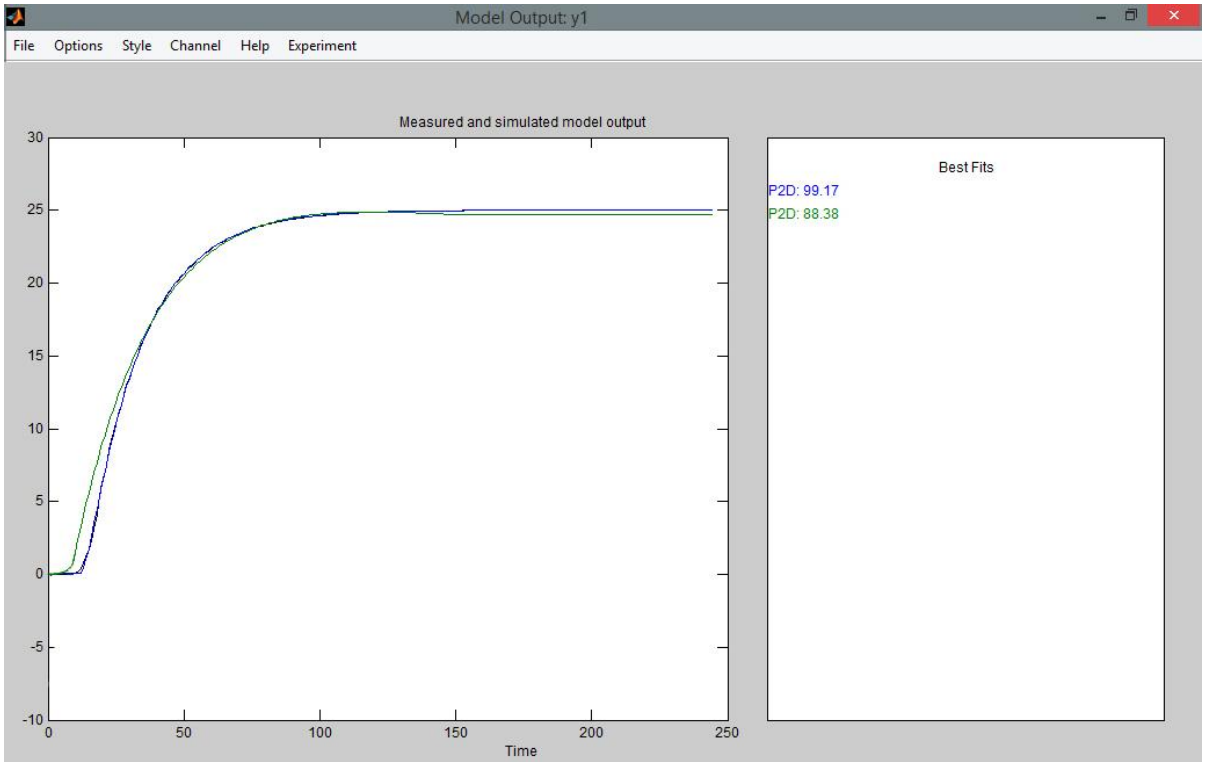

Figura 73. Porcentaje de aceptación de las funciones de transferencia del Sistema Real y el Sistema Embebido de Presión
Fuente: Matlab 2012a

La Función de transferencia del Sistema Real presenta un porcentaje de aceptación del 99.17% mientras que el Sistema embebido presenta una función de transferencia con un porcentaje de aceptación del 88.38% lo que hace posible definir que es válido.

5.2. Análisis de los Controladores

Para determinar el mejor desempeño del controlador PID se encontró sus parámetros mediante tres diferentes métodos analíticos Ziegler-Nichols, Lambda y Cohen-Coon.

En la Figura 81 se puede observar la respuesta del Sistema de Control de Caudal cuando se ha aplicado los distintos métodos de sintonía. En la Tabla 8 se realiza una comparación de los datos más sobresalientes al aplicar cada método. Es importante mencionar que de los valores de constantes obtenidos existió la necesidad de realizar una sintonía fina.

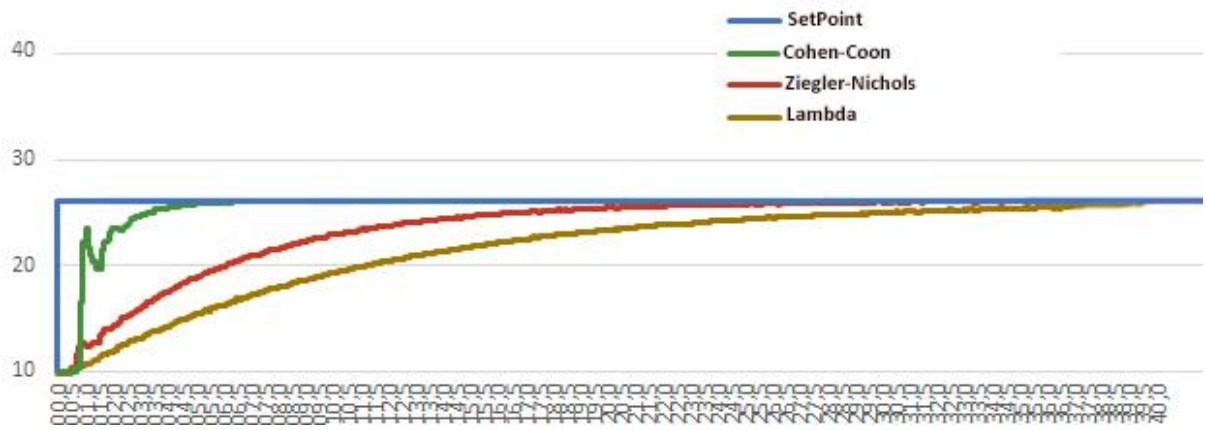

Figura 74. Métodos de Sintonía aplicados al Sistema de Control de Caudal

Tabla 9.
Comparación de los Métodos de sintonía para el Sistema de Control de Caudal

Método de Sintonía	Kp	Ti	Td	SobreIpulso	TS(s)
Ziegler-Nichols	0.82597	4.1346	0	0	30
Lambda	0.2199	2.84	0.6147	0	46
Cohen-Coon	0.5479	0.6538	0	0	8

De acuerdo a los resultados mostrados en la tabla el método de sintonía que mejor respondió ante este sistema es el método de Cohen-Coon pero que mostraba una perturbación antes de llegar al valor de consigna por lo que se tomó los parámetros.

En la Figura 82 se puede observar la respuesta del Sistema de Control de Nivel aplicados los distintos métodos de sintonía. En la Tabla 9 se realiza una comparación de los datos más sobresalientes al aplicar cada método. Es importante mencionar que de los valores de constantes obtenidos existió la necesidad de realizar una sintonía fina.

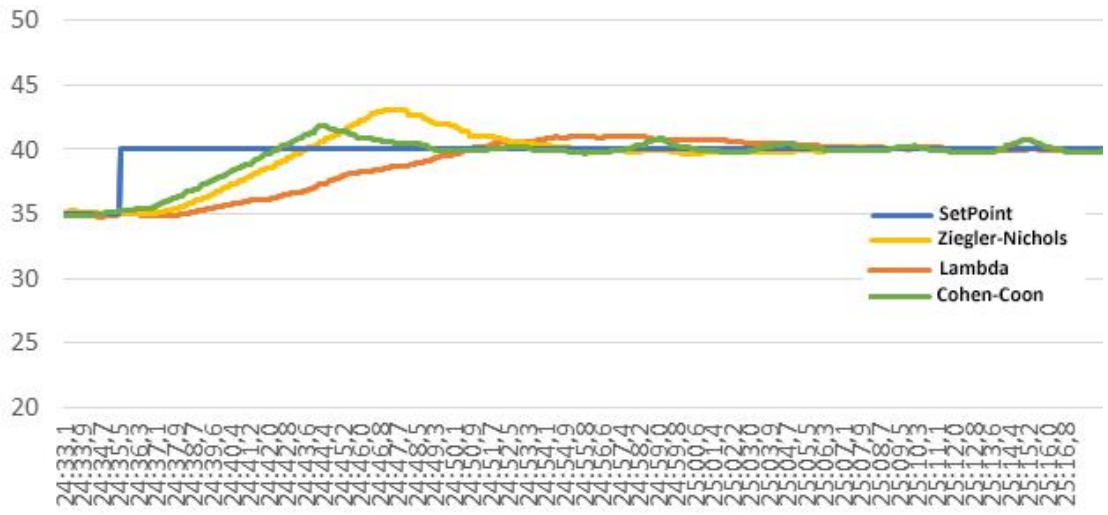

Figura 75. Métodos de Sintonía aplicados al Sistema de Control de Nivel

Tabla 10.
Comparación de los Métodos de Sintonía para el Sistema de Control de Nivel

Método de Sintonía	Kp	Ti	Td	SobreImpulso	TS(s)
Ziegler-Nichols	7.0519	8.4	1.554	0.077	19
Chien, Hrones y Reswick	0.2638	8.64	1.512	0.016	39
Cohen-Coon	23.989	0.10602	0	0.024	-

Para el Sistema de Control de Presión que es de segundo orden el proceso se tornó más complejo puesto que aquí se debió investigar sobre la sintonía de los sistemas de segundo orden, es posible aproximar el sistema de segundo orden a uno de primero y aplicar los tres métodos antes aplicados, pero esto implica que se pierda una cantidad de datos por lo que finalmente se trabajó con la función de segundo orden y solo se aplicó el método de Ziegler-Nichols por ser menos complejo. Este Sistema de Control de Proceso fue más sensible, que al cambio mínimo de la acción derivativa generó la oscilación del sistema por lo que se optó por manejarlo como un control Proporcional-Integral.

5.3. Pruebas Experimentales mediante la configuración Hardware in the Loop

Para el análisis del sistema se justifican los objetivos con la total integración del sistema de HIL. Además se explica el manejo del entorno para ser manipulado por el usuario en cuanto a la interfaz. El proyecto requiere mostrar los tres procesos, de forma que se realizó cada proceso por separado y se pudo observar que cada programa ocupa aproximadamente la mitad del espacio de la memoria interna del dispositivo, por lo cual se puede guardar un máximo de dos procesos por cada NI myRIO. Creamos un proyecto myRIO donde el Main.vi contiene las dos estaciones y utiliza 2 entradas analógicas, 8 entradas digitales y 2 salidas analógicas que pueden ser conectadas al PLC gracias a una tarjeta conectada al Puerto MXP de la myRIO.

Después de cargar en el PLC el programa creado del controlador PID y aplicar las constantes de sintonía, se creó una interfaz para poder enviar datos como el Setpoint al proceso y visualizar la respuesta del mismo, se conectó al sistema embebido mediante las entradas analógicas, digitales y salidas analógicas. Se ingresó las constantes obtenidas con los métodos de sintonía y se dio inicio a la generación de los datos del PID en el bloque del TIA Portal teniendo como resultados lo que se puede ver en las siguientes figuras:

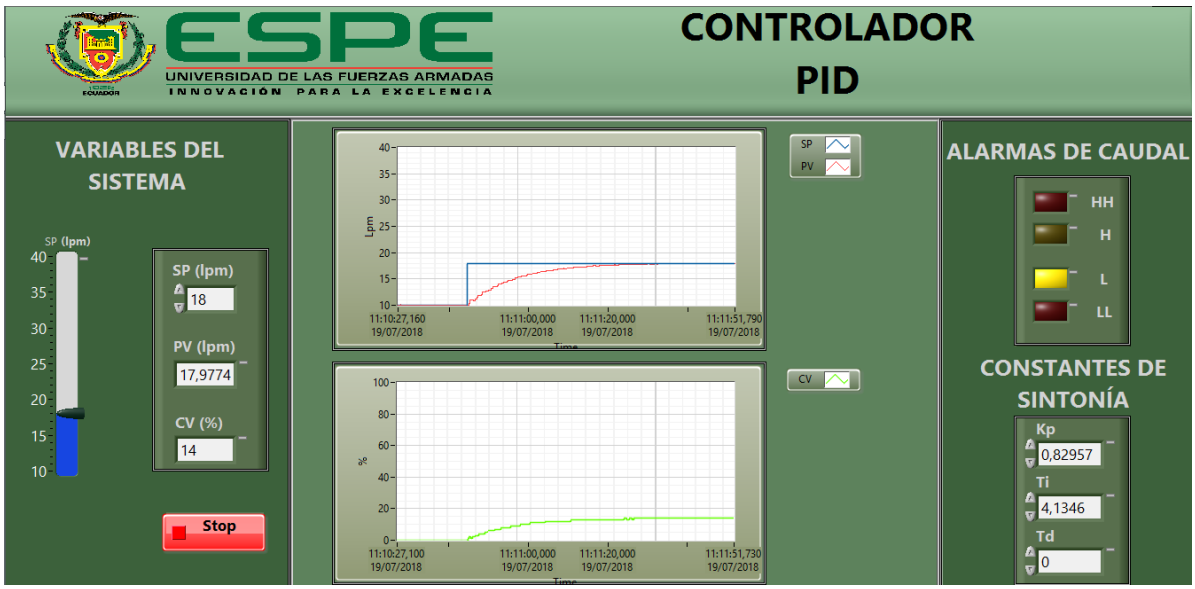

Figura 76. Respuesta del Sistema de Caudal con el Controlador PID
Fuente: LabVIEW2012

Para este proceso se asignó mediante la Interfaz Humano Máquina un valor de consigna de 18 litros por minuto (lpm), se ingresó igualmente sus constantes de

sintonía y se pudo observar la respuesta del sistema la misma que mostró un valor de aproximadamente 17.9777± 0.1 lpm.

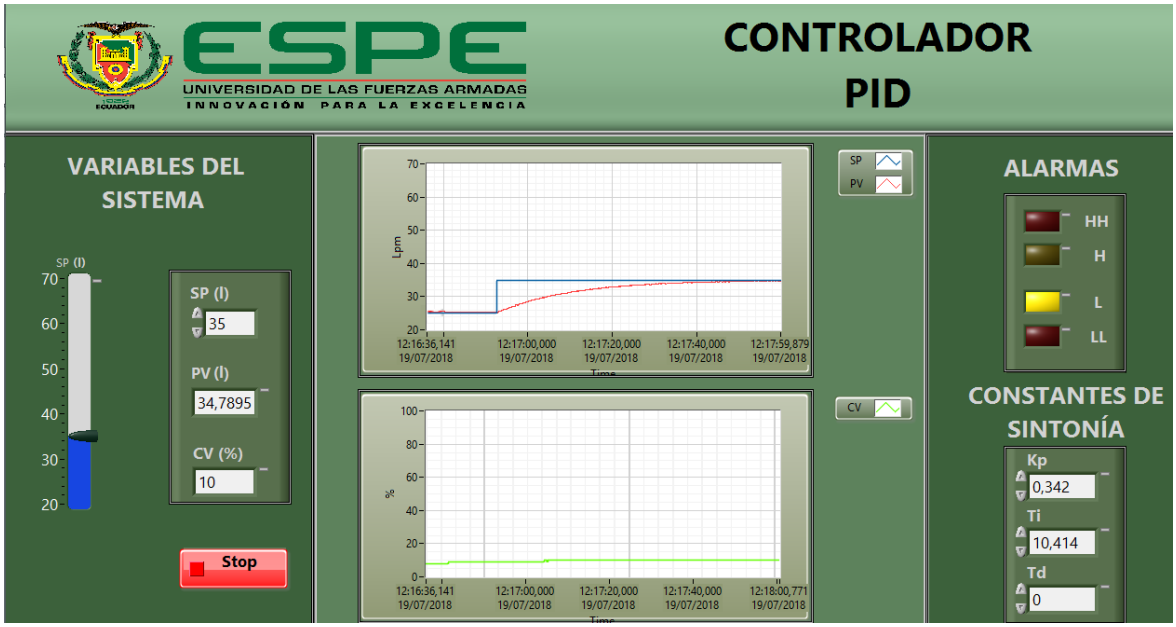

Figura 77. Respuesta del Sistema de Nivel con el Controlador PID
Fuente: LabVIEW2012

Para el control de Nivel se envió un valor de consigna de 35 litros (lt), se ingresó igualmente sus constantes de sintonía y la respuesta de este sistema fue aproximadamente 34.7895± 0.1 lt. Igualmente el funcionamiento del elemento de control final es aceptable y se determina la validez del controlador.

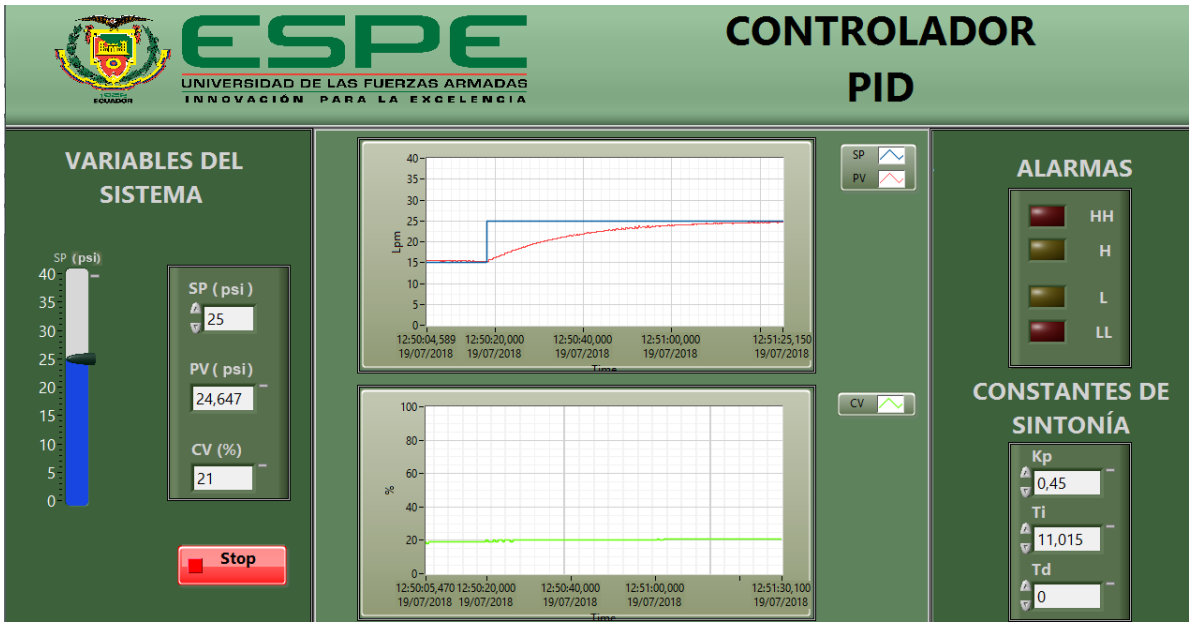

Figura 78. Respuesta del Sistema de Presión con el Controlador PID
Fuente: LabVIEW2012

Para el control de Presión se envió un valor de consigna de 25 PSI, se ingresó igualmente sus constantes de sintonía y la respuesta de este sistema fue aproximadamente 24.647 $\pm$ 0.4 psi y la respuesta de su elemento de control final es aceptable y se determina la validez del controlador.

El sistema de control de presión es un sistema de tipo subamortiguado, se puede ver su respuesta al escalón en la Figura 79. Tiene una relación directa con la función de transferencia de la forma:

$$G(s)=\frac{K{\omega_n}^2}{s^2+2\delta\omega_n s+{\omega_n}^2} \tag{5.7}$$

Donde δ es el valor del sobreimpulso que muestra la respuesta del sistema y que puede ser cambiado su valor para cambiar su sobreimpulso.

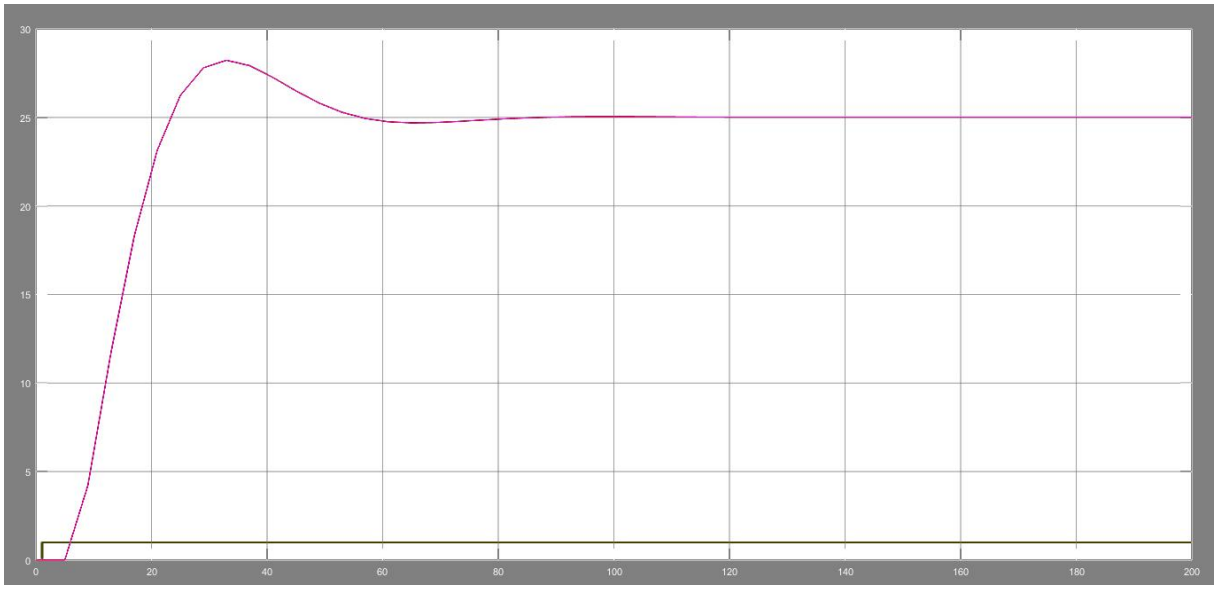

Figura 79. Proceso de Presión Subamortiguado
Fuente: LabVIEW2012

Cada proceso es distinto y para lograr el efecto visual es necesario utilizar imágenes tridimensionales que presenten profundidad. Para las plantas de caudal y nivel tienen Leds, tuberías, tanques, bombas, transmisor y un PLC. En la estación de presión contiene las mangueras, acoples de aire, transmisor, PLC, válvula de diafragma, conversor de I/P, indicadores de presión, tanques y descargador de aire.

A continuación observamos las parte principales de la interfaz. Una vez encendido el sistema con el interruptor “Power” podemos tener varias acciones como el manejo de flujo de datos con un interruptor con las opciones automático y manual, cada una con un Led indicador. Si escogemos la opción automático maneja los datos enviados por la unidad de control electrónico (ECU) para nuestro caso será del PLC S7 – 1200 los dos Leds junto al Autómata Programable indican la entrada y salida de datos, cuando esta de color verde se están enviando y recibiendo datos en el PLC, caso contrario se encienden de color rojo que indican que no se recibe ni envía ningún dato. Con la opción manual se carga los datos desde la misma interfaz a través de una perrilla virtual que envía datos deseados por el usuario en el rango que se encuentra la salida de la planta además se enciende un Led junto a la perrilla cuando el valor esta al máximo, para indicar que existe un sobrecarga en el sistema. Se maneja un cuadro de indicadores (focos) que representan las alarmas del proceso, estas se encuentran en la Interfaz Humano Máquina del proceso y del controlador. Cada proceso maneja un perrilla con denominación “V – 3” y representa a una válvula que genera una perturbación al sistema, se enciende un Led cuando la carga esta al máximo generando una gran perturbación al proceso. El elemento de control final está presente en cada proceso para nivel y caudal son representados con el variador, en presión está representado por la válvula de diafragma. Finalmente se manejan cuadros de texto que presenta información de advertencia o de gran relevancia, porque indica que el proceso está fuera de sus parámetros normales.

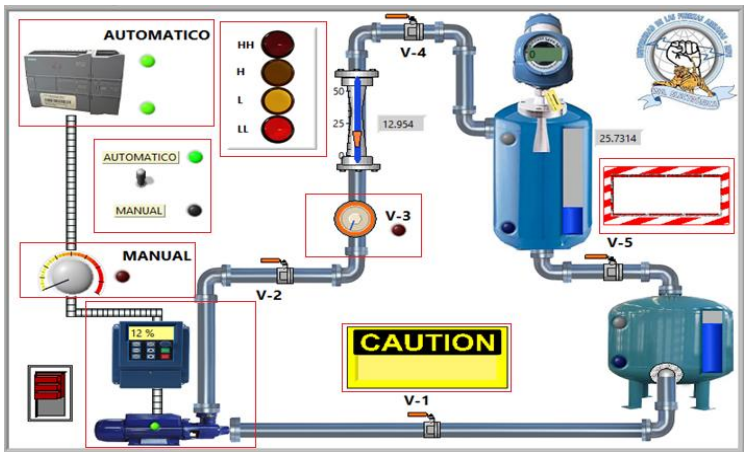

Figura 80. Indicadores fundamentales de la Interfaz Humano Máquina

Fuente Matlab 2012a

El resultado de manejar un instrumento virtual es la incorporación de cuadros que presenten mayor información de todos los sistemas, el proyecto tiene 4 áreas que indican información adicional del estado de las variables del proceso. Adicional se añade un área con leds indicadores que presentan el estado de las válvulas para el correcto funcionamiento de la interfaz.

Figura 81. Indicadores extra para mostrar el funcionamiento del Sistema

Fuente Matlab 2012a

CAPÍTULO VI

6. CONCLUSIONES Y RECOMENDACIONES

6.1. Conclusiones

- La obtención del comportamiento dinámico de los Sistemas (Plantas) de Control de Procesos se basa en las características de los elementos que lo conforman y su funcionamiento, su principal diferencia se establece en el tipo de Sistema y su configuración, lo que determina los métodos para la obtención de la función de transferencia.
- La tarjeta NI myRIO–1900 es un dispositivo con tecnología FPGA de alta velocidad que permite la programación en el entorno LabVIEW facilitando la creación del proyecto en tiempo real. Los tres bloques o frames que se ocuparon contribuyen en el procesamiento y sincronización de datos entre dispositivos, permitiendo emular buena integración con otros equipos.
- La validación de los Sistemas de Control de Procesos Embebidos hacen que el funcionamiento del entorno a través de Hardware in the Loop sea óptimo, presentando un alto grado de similitud de su respuesta con la obtenida a través de la configuración tradicional con los Sistemas Reales.
- La Interfaz Humano Máquina que presenta el diagrama del Sistema brinda una gran ventaja al utilizar el módulo Real-Time, permitiendo anular cualquier tiempo de retardo del Sistema, mostrando toda la información instantáneamente. Lo contrario a la Interfaz que muestra las gráficas de respuesta del Controlador donde el servidor OPC establece por defecto un tiempo de retardo de un segundo el mismo que puede ser configurado por el usuario.
- El módulo DCS de LabVIEW permite crear una interfaz normada, además de permitir la importación de gráficas 3D que presenten un entorno muy realista, brindando un mayor grado de inmersión al usuario con el fin de tener una mejor comprensión del Sistema y por lo tanto un correcto manejo de la información.
- El proyecto refleja datos coherentes para la sintonización del PLC Siemens S7-1200 con las plantas simuladas. Se evidencia que las curvas del controlador PID pueden corrige el error manteniendo los valores tolerables y existe una

desviación mínima con los valores de set point y los valores de la planta indicados por el transmisor de cada sistema.

6.2. Recomendaciones

- Para la programación es oportuno utilizar un proyecto propio para myRIO. Todo programa se debe realizar en el main que está dentro del dispositivo, pues la tarjeta ejecuta un programa a la vez y toma como prioridad la programación del programa principal Main.vi.
- Para nuestro esquema de programación, el segundo frame donde se realiza el tratamiento de la información debe excluir la opción de property node que trae conflicto cuando se deja al dispositivo en sistema embebido.
- En el ciclo while se debe trabajar con “Add Shift Register”, caso contrario genera un conflicto en el paso de los datos entre los bloque.
- Es importante cuando se trabaja con la función “Wait until Next ms Multiple” porque las pausas con la estructura “Timed Loop” provoca una desincronización cuando no se encuentran con los valores correctos.

REFERENCIAS BIBLIOGRÁFICAS

Aguilera Martínez, P. (2002). *Programación de PLC's. Tesis Mag. Ing. Manuf. Autom.* San Nicolás de los Garza: Universidad Autónoma de Nuevo León.

Álvarez Pulido, M. (2004). *Controladores Lógicos.* Barcelona: Marcombo S.A.

Arahal, M. R., Berenguel, M., & Rodríguez, F. (2006). Técnicas de predicción con aplicaciones en Ingeniería. En M. R. Arahal, M. Berenguel, & F. Rodríguez, *Técnicas de predicción con aplicaciones en Ingeniería* (págs. 48-50). Sevilla: Universidad de Sevilla, Secretariado de Publicaciones.

Bastidas Miño, R. F., & Proaño Cevallos, L. G. (2010). *Diseño y Construcción de Módulos Didácticos para el Laboratorio de Controladores Lógicos Programables para la Escuela Politécnica del Ejército Sede Latacunga. Tesis Ing. Electrom.* Latacunga: Escuela Politécnica del Ejército Sede Latacunga.

D´Arthenay Bermúdez, D. H. (2015). *Desarrollo de un simulador de procesos industriales bajo configuración Hardware-in-the-Loop para la práctica-enseñanza de control lógico y regulatorio mediante un PLC. Tesis Mag. Ing. Aut. Ind.* Bogotá: Universidad Nacional de Colombia Facultad de Ingeniería, Departamento de Ingeniería Eléctrica y Electrónica.

Florencio Inga, P. P. (2012). *Desarrollo del Software de un Sistema SCADA para la Distribución de Agua Potable en la quebrada de Manchay.* Lima: Pontificia Universidad Católica del Perú.

Hernández Cevallos, M. I., & Ledesma Marcalla, D. A. (2010). *Desarrollo de un Sistema SCADA para la Medición de Voltajes con Sistemas Embebidos para el Laboratorio de Mecatrónica de la Facultad de Mecánica. Tesis Ing. Mec.* Riobamba: Escuela Politécnica de Chimborazo.

Instruments National. (10 de Junio de 2016). *ni.com.* Recuperado el 14 de Abril de 2018, de http://www.ni.com/pdf/manuals/376047c.pdf

Lajara Vizcaino, J. R., & Pelegrí Sebastiá, J. (2007). *LABVIEW Entorno Gráfico de Programación.* Barcelona: Marcombo Editoriales.

Lavi , J., & Kudish, J. (2004). *Systems Modeling and Requirements Specification Using ECSAM: An Analysis Method for Embedded and Computer-Based Systems* (Vol. 1). En: Innovations Syst Soft Eng.

Luna, O., Muriel, O., Pantoja , J., Giraldo, J. S., & Arrechea, K. (2017). *Reconocimiento de Controladores PLC's.* El Cauca: Universidad del Cauca.

Luque Navarro, I. (2017). *DESARROLLO DE UN SISTEMA HARDWARE-IN-THE LOOP SOBRE PLATAFORMA RASPBERRY PI PARA LA SIMULACIÓN DEL VEHÍCULO AÉREO KADETT 2400 DE GRAUPNER. Tesis Ing. Tec. Ind.* Valencia: Universitat Politécnica de Valencia.

MICRO Automación. (2000). *Automatización y Control.* México, D.F.: MICRO.

National Instruments. (20 de Febrero de 2004). *www.ni.com.* Recuperado el 21 de Abril de 2018, de http://www.ni.com/pdf/manuals/322154e.pdf

Ogata, K. (2003). *Ingeniería de Control Moderna* (Vol. 4). (I. Capella, Ed.) Madrid: PEARSON EDUCACIÓN S.A.

Pacheco Rubio, C. J., & Palencia Cadena, A. R. (2008). *Estado del Arte de la Automatización en planta 2 de Mexichem Resinas Colombia .* Cartagena: Universidad Tecnológica de Bolívar.

Siemens. (07 de Abril de 2011). *Siemens*. Recuperado el 08 de Mayo de 2018, de https://w5.siemens.com/spain/web/es/industry/automatizacion/simatic/tia-portal/tia_portal/pages/tiaportal.aspx

Sofrony, J. (5 de Agosto de 2009). *Study Resource.* Obtenido de http://studyres.es/doc/1780555/lenguajes-de-programacion---sofrony

Vallejo, H. D. (12 de Agosto de 2005). *www.editorialquark.com.er.* Recuperado el 25 de Abril de 2018, de http://www.todopic.net/utiles/plc.pdf

Velásquez, J. A. (3 de Mayo de 2012). *Slideshare.* Recuperado el 7 de Mayo de 2018, de https://es.slideshare.net/jvelasquezc/automatizacin-en-el-proceso-de-manufactura

Villajulca, J. C. (21 de Octubre de 2009). *Instrumentación y Control.net.* Recuperado el 3 de Mayo de 2018, de https://instrumentacionycontrol.net/unidad-central-de-procesos-de-un-plc/

ANEXOS

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA EN ELECTRÓNICA E INSTRUMENTACIÓN

CERTIFICACIÓN

Certifico que el trabajo de titulación, **"HARDWARE-IN-THE-LOOP PARA EL CONTROL DE LOS PROCESOS PRESIÓN, CAUDAL Y NIVEL, MEDIANTE EL SISTEMA EMBEBIDO MY RIO A REALIZARSE EN EL LABORATORIO DE REDES INDUSTRIALES Y CONTROL DE PROCESOS"** ha sido realizado por los señores, **ALULEMA GARZÓN JUAN CARLOS Y MORA NÚÑEZ NICOLÁS SEBASTIÁN** el mismo cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de las Fuerzas Armadas ESPE.

En la ciudad de Latacunga, a los 29 días del mes de agosto del 2018.

Aprobado por:

Ing. Pruna Panchi, Edwin Patricio

DIRECTORA DEL PROYECTO

Ing. Amparo Meythaler

DIRECTOR DE CARRERA

Dr. Rodrigo Vaca

SECRETARIO ACADÉMICO