Universidad de La Salle
Ciencia Unisalle

Ingeniería en Automatización | Facultad de Ingeniería

6-14-2017

Desarrollo de un simulador digital en tiempo real basado en xPC Target

Jeimmy Johana Romero Melo
Universidad de La Salle, Bogotá

Esteban Gustavo Rodríguez Pinzón
Universidad de La Salle, Bogotá

Follow this and additional works at: https://ciencia.lasalle.edu.co/ing_automatizacion

Part of the Other Engineering Commons

Citación recomendada

Romero Melo, J. J., & Rodríguez Pinzón, E. G. (2017). Desarrollo de un simulador digital en tiempo real basado en xPC Target. Retrieved from https://ciencia.lasalle.edu.co/ing_automatizacion/20

This Trabajo de grado - Pregrado is brought to you for free and open access by the Facultad de Ingeniería at Ciencia Unisalle. It has been accepted for inclusion in Ingeniería en Automatización by an authorized administrator of Ciencia Unisalle. For more information, please contact ciencia@lasalle.edu.co.

DESARROLLO DE UN SIMULADOR DIGITAL EN TIEMPO REAL BASADO EN xPC TARGET

JEIMMY JOHANA ROMERO MELO

ESTEBAN GUSTAVO RODRÍGUEZ PINZÓN

UNIVERSIDAD DE LA SALLE
FACULTAD DE INGENIERÍA
INGENIERÍA EN AUTOMATIZACIÓN
BOGOTÁ
14 de junio de 2017

DESARROLLO DE UN SIMULADOR DIGITAL EN TIEMPO REAL BASADO EN xPC TARGET

JEIMMY JOHANA ROMERO MELO

ESTEBAN GUSTAVO RODRÍGUEZ PINZÓN

TRABAJO DE GRADO PARA OPTAR AL TÍTULO DE INGENIERO EN AUTOMATIZACIÓN

Director
EFRAIN BERNAL ALZATE

UNIVERSIDAD DE LA SALLE
FACULTAD DE INGENIERÍA
INGENIERÍA EN AUTOMATIZACIÓN
BOGOTÁ
14 de junio de 2017

Nota de aceptación:

_____

_____

_____

_____

_____

_____

_____

_____

Director de Trabajo de Grado

_____

Jurado

_____

Jurado

Bogotá, Colombia, 14 de junio de 2017

AGRADECIMIENTOS

A mi familia por apoyarme incondicionalmente y por brindarme los recursos necesarios para estudiar. A nuestro director de tesis Ing. Efrain Bernal por su asesoria y dedicación en el proyecto.
Gracias

Johana Romero.

Todo un conjunto de personas, momentos, conocimiento y motivación merecen ser nombrados en los agradecimientos, hacen parte del proyecto y su realización pero además es la clausura de una etapa, un pregrado y todo lo que conlleva(R.I.P a mis ganas de estudiar por un tiempo).
Especialmente: la participación incomparable de mi familia, papá con su particular gusto por mi carrera, mi madre espantando cualquier intento de desmotivación, mi hermanita y su bullying incondicional, a mis abuelos con la dosis de amor, también, a nuestro director Efrain Bernal que basado en su temperamento hostil y en su experiencia permite un trabajo sólido. Finalmente agradezco a la ULS por permitirme disfrutar la mejor fucking y agradable experiencia.

Esteban Rodríguez Pinzón.

CONTENIDO

pág.

1. SIMULADOR DIGITAL EN TIEMPO REAL (RTDS) 16
1.1. TECNOLOGÍAS PARA LA SIMULACIÓN 16
1.1.1. Rapid Control Prototyping (RCP) 16
1.1.2. Hardware in the Loop (HIL) 16
1.1.3. Software in the Loop (SIL) 16

2. TECNOLOGÍA xPC 17
2.1. CONFIGURACIÓN xPC 18
2.1.1. Ajustes en el Target PC 18
2.1.2. Conexión y arranque del Target PC 18
2.1.3. Configuración del entorno de Simulink para la simulación en tiempo real . . 19

3. SELECCIÓN DEL CASO DE ESTUDIO 22
3.1. MARCO TEÓRICO 22
3.1.1. Identificación de sistemas de primer orden 22
3.1.2. Root Mean Square Error (RMSE) 23
3.2. PÉNDULO INVERTIDO 23
3.2.1. Modelamiento teórico 25
3.2.2. Identificación de las variables Alpha y Theta 26
3.2.3. Modelo embebido en el xPC 28
3.2.4. Análisis de las identificaciones del péndulo 30
3.3. SMART GRID (SG) INTEGRADA A UN SISTEMA FOTOVOLTAICO 32
3.3.1. Modelo a Embeber de la SG 34
3.3.2. Simulación en tiempo real de la SG 34
3.4. UNIDAD DE ENTRENAMIENTO EN AUTOMATIZACIÓN (UEA) 35
3.4.1. Métodos de obtención de modelos 37
3.4.1.1. Modelamiento teórico 37
3.4.2. Identificación de las variables Flujo y Presión 42
3.4.3. Condiciones iniciales 44
3.5. ANÁLISIS DE RESULTADOS 45

4. IMPLEMENTACIÓN DE LA UEA EN EL RTDS 46
4.1. MARCO TEÓRICO 46
4.1.1. Conjunto de protocolos TCP/IP 46
4.2. ARQUITECTURA 47
4.2.1. Descripción 47
4.2.2. Simulador xPC vs UEA 47
4.3. MODELOS PARA LAS VARIABLES FLUJO Y PRESIÓN 50
4.3.1. Discretización Funciones de Transferencia 50
4.3.2. Implementación MATLAB/Simulink® 50

4.4. VALIDACIÓN DEL MODELO 52
4.4.1. Flujo con una válvula 53
4.4.2. Flujo con dos válvulas 54
4.4.3. Presión con una válvula 55
4.4.4. Presión con dos válvulas 57

5. ANÁLISIS DE LA RESPUESTA DEL CASO DE ESTUDIO CON UN CONTROLADOR 59
5.1. MARCO TEÓRICO 59
5.1.1. Sistema de control 59
5.1.2. Discretización del control 61
5.1.3. Programación de PLCs 61
5.1.4. Open Platform Communications (OPC) 62
5.2. MODELO FINAL A EMBEBER 62
5.3. DISEÑO DEL CONTROL 63
5.3.1. Simulaciones del control de las variables Flujo y Presión 66
5.4. PROGRAMACIÓN DEL PLC 69
5.4.1. Comunicación con el Target PC 69
5.4.2. Implementación del control 73
5.5. Adquisición de datos por OPC 74
5.6. ANÁLISIS DEL CASO DE ESTUDIO IMPLEMENTADO EN EL RTDS CON UN CONTROL 75
5.6.1. Flujo con una válvula: variaciones en un rango de 200 L/h a 430 L/h. 76
5.6.2. Flujo con dos válvula: variaciones en un rango de 300 L/h a 600 L/h. 77
5.6.3. Presión con una válvula: variaciones en un rango de 1 psi a 9 psi. 78
5.6.4. Presión con dos válvula: variaciones en un rango de 0.5 psi a 4.5 psi. 80

6. CONCLUSIONES 82
BIBLIOGRAFÍA 83

LISTA DE FIGURAS

pág.

1. xPC Operation Principle . . . 17
2. xpcexplr Interface . . . 19
3. Solver . . . 20
4. Code Generation . . . 20

5. Physical Pendulum . . . 24
6. WorkStation Components Pendulum . . . 24
7. Model Identification Pendulum . . . 27
8. Voltage In for Pendulum . . . 27
9. Pendulum Signals . . . 28
10. Pendulum Model . . . 29
11. Real Time Simulation of Pendulum . . . 30
12. Alpha Identification Vs. Alpha Real . . . 30
13. Alpha RMSE . . . 31
14. Theta Identification Vs. Theta Real . . . 31
15. Theta RMSE . . . 32
16. Smart Grid . . . 33
17. Structure of On-Grid Photovoltaic System . . . 33
18. Smart Grid Model . . . 34
19. xPC Smart Grid . . . 35
20. P&ID UEA . . . 36
21. Fluid System . . . 37
22. Fluid System . . . 39
23. Balance de energía . . . 40
24. Firts Orden Comparison . . . 42
25. Flow Stair Response One Transfer Function-Real Plant . . . 43
26. Flow Stair Response Error One Transfer Function-Real Plant . . . 44

27. Simulator Architecture . . . 47
28. Physical Plant . . . 48
29. Physical xPC . . . 49
30. Flow One Valve Model . . . 51
31. xPC Flow Screenshot . . . 52
32. Flow Open Loop UEA - xPC Comparison with One Valve . . . 53
33. Flow Error Open Loop UEA - xPC Comparison with One Valve . . . 54
34. Flow Open Loop UEA - xPC Comparison with Two Valves . . . 54
35. Flow Error Open Loop UEA - xPC Comparison with Two Valves . . . 55
36. Pressure Open Loop UEA - xPC Comparison with One Valve . . . 56
37. Pressure Error Open Loop UEA - xPC Comparison with One Valve . . . 56
38. Pressure Open Loop UEA - xPC Comparison with Two Valves . . . 57
39. Pressure Error Open Loop UEA - xPC Comparison with Two Valves . . . 57

40. Flow One Valve Model with Communication 63
41. PI Simulation Model 67
42. Flow Control Response 67
43. Root Locus Flow Comparison 68
44. Pressure Control Response 68
45. Root Locus Pressure Comparison 69
46. Scale Flow Signal 70
47. PLC Inputs Programming 71
48. Word Division 72
49. UDP Configuration Blocks 72
50. Control Implementation 73
51. Control Activation 74
52. OPC Configuration 75
53. PI Flow Control UEA - xPC Comparison with One Valve 76
54. PI Flow Control Error UEA - xPC Comparison with One Valve 77
55. PI Flow Control UEA - xPC Comparison with Two Valves 77
56. PI Flow Control Error UEA - xPC Comparison with Two Valves 78
57. PI Pressure Control UEA - xPC Comparison with One Valve 79
58. PI Pressure Control Error UEA - xPC Comparison with One Valve 79
59. PI Pressure Control UEA - xPC Comparison with Two Valves 80
60. PI Pressure Control Error UEA - xPC Comparison with Two Valves 81

LISTA DE TABLAS

pág.

1. UEA Components. 36
2. Selection Criteria. 45

3. xPC software and hardware Requeriments. 49

4. Flow One Valve Transfer Functions. 87
5. Flow Two Valves Transfer Functions. 88
6. Pressure One Valve Transfer Functions. 89
7. Pressure Two Valves Transfer Functions. 90

RESUMEN

Este proyecto consiste en el diseño e implementación de un Simulador Digital en Tiempo Real (RTDS) utilizando la tecnología xPC de MATLAB®. Para el desarrollo del RTDS, se realiza el análisis de 3 casos de estudio: primero, el péndulo invertido, segundo, una Smart Grid (SG) y tercero, la Unidad de Entrenamiento en Automatización (UEA), en cada caso se considera la representación matemática, el controlador y la etapa de potencia. Se elige para el desarrollo del simulador la UEA, ya que cumple con todos los criterios de selección.

El desarrollo del simulador se ejecuta en varias etapas: configuración de software y hardware para el computador maestro y esclavo, implementación de la representación matemática en MATLAB/Simulink®, diseño de la arquitectura del simulador, configuración de comunicaciones con el controlador, diseño del control, programación del PLC e integración de las etapas. Posteriormente, se realiza la validación del simulador mediante métodos estadísticos que requieren estimular al simulador y al caso de estudio seleccionado en condiciones equivalentes. Los resultados exponen que el RTDS tiene un RMSE máximo de 0.9 L/h para la variable de flujo y 0.09 psi para la variable de presión.

Se concluye que el RTDS desarrollado, permite validar la efectividad de controles diseñados para las variables de flujo y presión de la UEA, antes de ser aplicados al sistema real, además de ofrecer una predicción de la respuesta de la planta en puntos críticos de operación, este procedimiento permite evitar el deterioro de la unidad en caso de que el control exija a los elementos físicos acciones que estos no puedan realizar y por lo tanto causar el daño de los mismos.

INTRODUCCIÓN

En la industria cuando se desea reemplazar un actuador o una estrategia de control en un sistema, se genera el riesgo de perjudicar la estructura física, ya que no se puede estimar el comportamiento del sistema al realizar la sustitución. La manera adecuada de ejecutar dicha sustitución es evaluando la sección afectada sin detener el proceso de producción, pues si éste se suspende se generan pérdidas económicas. Una forma de mitigar estas pérdidas, es llevar a cabo la simulación del controlador y de la planta en un software especializado como MATLAB®, sin embargo para sistemas complejos la respuesta del simulador es muy lenta, por lo tanto, se hace necesario el uso de bancos de prueba y simuladores en tiempo real, que permitan conocer la respuesta del sistema a las nuevas configuraciones sin afectar ningún elemento físico.

La implementación de un banco de prueba requiere gran espacio, sistemas de acople y *drivers* para cada actuador, por otra parte, los simuladores en tiempo real comerciales tienen un costo elevado lo que genera que en ocasiones su adquisición no sea posible, por estos motivos, la opción más viable para ejecutar simulaciones en tiempo real con fines académicos es el desarrollo de un RTDS implementado en un hardware de fácil obtención.

Para una simulación en tiempo real se requiere en primer lugar, un computador maestro (Host PC) donde se parametriza el modelo y en segundo lugar, un computador esclavo (Target PC) encargado de ejecutar la simulación. Es preciso resaltar que después de configurar el computador esclavo, éste puede funcionar de forma independiente al computador maestro. Adicionalmente, el computador esclavo cuenta con una tarjeta de adquisición para las señales análogas y digitales, que permite interactuar con el Controlador Lógico Programable (PLC).

El presente proyecto desarrolla un RTDS para un proceso industrial, el cual permite evaluar estrategias de control. El documento está organizado por capítulos y dichos capítulos describen: la Tecnología xPC, las características de los casos de estudio, el diseño de los controles, la implementación en el RTDS, la validación del simulador y las conclusiones.

OBJETIVOS

GENERAL

Desarrollar un Simulador Digital en Tiempo Real basado en xPC Target.

ESPECÍFICOS

- Analizar y seleccionar el caso de estudio.
- Analizar y diseñar la arquitectura del software y hardware.
- Implementar el modelo de la planta en el sistema embebido.
- Analizar y validar el simulador con el caso de estudio.

ANTECEDENTES

Los primeros avances de RTDS fueron presentados en el trabajo de (McLaren, Kuffel, Wierckx, Giesbrecht, y Arendt, 1992), el cual describe la estructura y el rendimiento de un RTDS empleado para la prueba de relés. El RTDS utiliza un procesamiento en paralelo basada en un Procesador de Señal Digital (DSP) para ejecutar simulaciones en tiempo real de sistemas de potencia con un tiempo de muestreo de 50-100 us. Los dispositivos físicos tales como relés, monitores de energía para MOV o estabilizadores del sistema de potencia pueden alimentarse con las señales del RTDS y la salida del dispositivo físico puede ser realimentada al simulador.

Con el fin de determinar la arquitectura del RTDS, (Kuffel, Giesbrecht, Maguire, Wierckx, y McLaren, 1995) expone que el simulador en tiempo real es una combinación de hardware y software especializado y es implementado en la solución de sistemas electromecánicos de potencia, a demás, se ejecuta la simulación en una interfaz gráfica instalada en el computador maestro, la conexión entre el RTDS y la estación de trabajo se realiza por medio de una red de área local, para la adquisición de señales externas se utiliza un dispositivo de entradas y salidas conectado al computador esclavo, finalmente se concluye que RTDS puede ser usado en pruebas de relés para protección y equipos de control.

Con la integración de los microcontroladores en los RTDS, (Kezunovic y cols., 1996) realiza una implementación del diseño de un simulador digital para la pruebas de relés, con un sistema de software y hardware de bajo costo utilizando un microprocesador RISC Y 4 DSP, adicionalmente cuenta con una Interfaz Gráfica de Usuario (GUI). La arquitectura estaba formada por tres terminales: reconstrucción, amplificación y adquisición de las señales de entrada y salida, cada una con su propio hardware. Los resultados mostraron que el tiempo de respuesta del simulador estaba entre 50 y 100 us, por lo que podían ser utilizado en la simulación de redes complejas y modelos detallados.

Años más tarde, se realiza la implementación de PLCs en simuladores de tiempo real, como se muestra en el trabajo de (G. Randolf, 2005) donde se describe el diseño de un sistema de prueba dinámica para una pila de combustible. El simulador en tiempo real está compuesto por: el banco de pruebas PC (que controla el funcionamiento del puesto de ensayo) y el PLC, para las tareas de seguridad y control de bajo nivel, demostrando la rápida respuesta del *Hardware in the loop* (HIL) integrado, siendo una solución eficiente para aplicaciones en tiempo real.

El sistema de energía radial trifásico trabajado en (Uriarte y Butler-purry, 2006) modela y simula un sistema en tres simuladores de tiempo real diferentes: GRT, xPC de MathWorks y EST de National Instruments; el primero solo requiere un computador Host, no cuenta con *Transmission Control Protocol/Internet Protocol* (TCP-IP), es limitado a modelos de complejidad simple, se ejecuta bajo el sistema operativo de Windows, su mayor ventaja es la rápida implementación para prototipos pequeños. Por otra parte xPC, es un gestor de arranque en tiempo real, que utiliza un computador Host y un Target, una de sus ventajas es el determinismo en modelos complejos aunque desactiva las comunicaciones mientras

está en tiempo real. Por último ETS, necesita un disco duro adicional para registrar los datos y recibir los archivos desde el Host PC. También requiere un software adicional para fines de comunicación, una de sus principales ventajas es la de permitir utilizar la interfaz de LabVIEW para interactuar con la simulación en tiempo real. Se concluye que los simuladores que utilizan ordenadores para su implementación están en la capacidad de ejecutar sistemas de energía de menor tamaño en tiempo real.

Una de las aplicaciones donde más se ha visto implementado el HIL es en los controladores de la electrónica de potencia, el trabajo de (Lu, Wu, Figueroa, y Monti, 2007) aplica simulación en tiempo real y HIL en un conversor y un puente H con sus respectivos sistemas de control, utilizando la plataforma de Banco de Pruebas Virtual con un extensión en Tiempo Real (VTB-RT); la cual permite la integración de simulaciones en diferentes ambientes, distribución de tareas en varios ordenadores, modificaciones ON-LINE y alto nivel de simulación. Para la implementación de esta plataforma se utilizaron componentes como: LINUX, Aplicación de Interfaz en Tiempo Real (RTAI), *freeware* COMEDI para el desarrollo de una plataforma abierta para los *drivers* de diferentes tarjetas de adquisición, al visualizar los resultados se pudo concluir que la aplicación de HIL permite eliminar la necesidad de utilizar un hardware costoso, ya que es capaz de producir un rendimiento comparable con los software de simulación en tiempo real comerciales, sin embargo las características del hardware deben ser evaluadas teniendo en cuenta el ancho de banda del sistema que se desea simular.

Por otra parte, (Shi, Miao, Zou, y Jiao, 2009) implementó controladores para *Real Time Windows Target* (RTWT) y xPC, se examinó el principio de la RTWT y xPC como simuladores, el simulador se desarrolló para una PCI-8360A. Más adelante, (Paquin, 2010) presentó las características más relevantes de la simulación en tiempo real: seleccionar el simulador en tiempo real que cumpla con las necesidades que exige el proceso; la capacidad, el tamaño y el costo son determinados por: la máxima frecuencia a la que va a ser llevado el simulador, la complejidad o el tamaño del sistema a simular, el número de canales de entradas y salidas, entre otras. El uso más frecuente de los simuladores en tiempo real se muestra en aplicaciones de Prototipado de Control Rápido (RCP) donde el simulador es conectado a la planta física y HIL donde el controlador físico es conectado a una planta virtual ejecutada en un simulador en tiempo real en lugar de la planta real. Una de las ventajas de HIL frente a RCP es que permite ensayar el controlador cuando la planta física no está disponible. Adicionalmente el uso de una planta virtual tiene un costo menor y provee repetitividad en los resultados. Según el estudio, las aplicaciones más relevantes de los simuladores en tiempo real son: generación de energía, automovilismo, redes eléctricas y educación.

Otro avance en el desarrollo de RTDS se presenta en el trabajo de (Abdullah Mohamed, 2013) donde se crea un Sistema Operativo (OS) en tiempo real, basado en xPC Target Kernel, se construyen dos modelos, un transmisor Target-Host con UDP y un analizador de espectro; el primero para probar la capacidad del OS cuando se trata de sistemas de red y el segundo para poner a prueba la capacidad de trabajar con sistemas de alta velocidad.

Los resultados muestran que el tiempo de ejecución de la tarea en el OS para el primer modelo fue de 0.636 us y para el segundo modelo de 0.311 us.

(Zhang y Cheng, 2015) realizó un sistema de pruebas en tiempo real para una turbina de viento utilizando el software xPC de MATLAB®, ésta solución con la que cuenta MATLAB/Simulink® puede convertir el modelo de la simulación construido en el Host PC, en un código mediante el *workshop* de tiempo real, el cual puede ser descargado en el xPC Target por medio de ethernet o RS232. El xPC Target es el encargado de correr la simulación y permitir la observación del comportamiento de las señales. Se utilizó una tarjeta de adquisición de señales Adlink PCI-6208v, que se agregó a la simulación por medio de los *drivers* con los que cuenta la librería de Simulink. El controlador de la turbina es conectada con el xPC Target utilizando una tarjeta de señales que se conectada a la PCI-6208v y luego al Host PC logrando una configuración de HIL. Los resultados experimentales muestran que la simulación en tiempo real puede ser aplicada a investigaciones y desarrollo de tecnologías de energía eólica, especialmente en el diseño de equipos de verificación bajo condiciones extremas.

Los simuladores comenzaron a ser implementados en el ensayo de controladores en el trabajo de (Islam, Muyeen, Al-Durra, y Hasanien, 2016), ya que propone un nuevo enfoque para la dinámica de ensayo y aspectos de control de un sistema a gran escala fotovoltaica (PV) en tiempo real, junto con la resolución de obstáculos de diseño de los parámetros del controlador utilizado en el RTDS. Para comprobar la viabilidad del simulador se desarrollaron diferentes técnicas y estrategias de control para el sistema de captación como las perturbaciones en la red, optimizando el tiempo de recuperación. Se concluyó que la implementación del simulador permite la optimización en el control para sistemas fotovoltaicos.

Con el fin de evaluar la efectividad de los RTDS, (Liu, Goorts, Ashasi-Sorkhabi, Mercan, y Narasimhan, 2016) presentó una comparación de entornos (MATLAB y LabVIEW) para la implementación de un simulador en tiempo real híbrido, compuesto por el modelo físico de la planta y un algoritmo numérico (Parte digital) con el fin de determinar predicciones y ajustes a la planta, finalmente, concluye que es un método efectivo y permite ser desarrollado con gran robustez.

1. SIMULADOR DIGITAL EN TIEMPO REAL (RTDS)

Un RTDS se define como una combinación del hardware de un computador y un software diseñado para simulaciones en tiempo real. Uno de los aspectos más relevantes de un RTDS, es la capacidad de conectar las entradas y salidas del sistema de ensayo con el Target PC ejecutándose en tiempo real. Este método de simulación es llamado HIL, por medio de esta simulación, el intercambio de información entre el simulador y el dispositivo de testeo se encuentran en un lazo cerrado.

La simulación digital en tiempo real, está basada en tiempos discretos, donde el simulador secuencialmente resuelve las ecuaciones que representan el modelo del sistema. La simulación está limitada por el *Task Ejecution Time* (TET) del hardware que a su vez es el mínimo *sample time*, si el modelo no requiere el mínimo *sample time* deberá determinarse uno que represente con precisión la frecuencia requerida del sistema.

1.1. TECNOLOGÍAS PARA LA SIMULACIÓN

Los simuladores digitales en tiempo real se pueden usar típicamente en diferentes categorías de aplicación: Rapid Control Prototyping con una planta física, *Hardware in the Loop* y *Software in the Loop* (Paquin, 2010).

1.1.1. Rapid Control Prototyping (RCP)

En las aplicaciones de RCP, un controlador de una planta es implementado usando un simulador en tiempo real y éste se conecta a una planta real. RCP posee ventajas en la implementación para el prototipo del controlador. Este controlador en el simulador es más flexible y su desarrollo e implementación son sencillos.

1.1.2. Hardware in the Loop (HIL)

Para las aplicaciones con HIL, un controlador físico es conectado a una planta simulada ejecutándose en tiempo real en vez de una planta física. La simulación con HIL permite el ensayo de controladores cuando no es posible realizar pruebas del control en la planta real, otra ventaja es el costo de la planta simulada, además de tener la posibilidad de realizar cambios y ensayos en puntos críticos sin un costo adicional.

1.1.3. Software in the Loop (SIL)

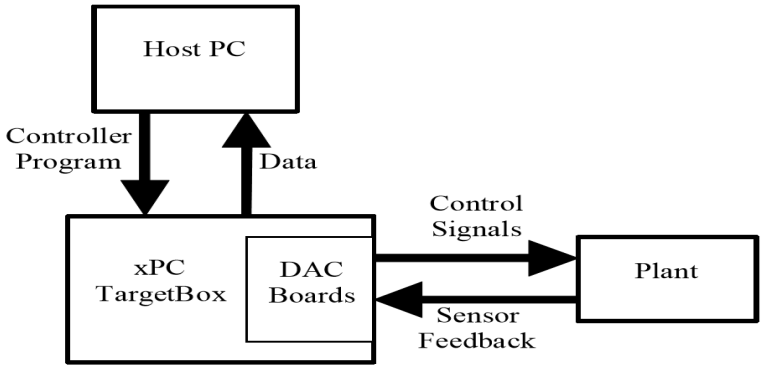
SIL, permite realizar la simulación en tiempo real de la planta y el simulador en el mismo dispositivo. SIL presenta ventajas sobre RCP y HIL ya que no es necesario el uso de entradas y salidas preservando las características de la señal.

2. TECNOLOGÍA xPC

Teniendo en cuenta las definiciones del capítulo anterior, se selecciona la categoría HIL que es construida empleando el *toolbox* xPC Target de MATLAB®.

Este *toolbox* permite convertir la simulación de un modelo de MATLAB/Simulink® en un código para ser embebido en el Target PC y generar una simulación en tiempo real. xPC utiliza el *kernel* y hardware del computador, el *kernel* no está basado en Windows, lo que significa que el RTDS puede funcionar en cualquier OS, obteniendo una máquina xPC ideal y rentable. Los componentes de cualquier sistema xPC Target son: Host PC, Target PC y un elemento de ensayo; el principio de operación del xPC y sus componentes se muestran en la Figura 1.

Figura 1: xPC Operation Principle



Fuente: Caterpillar Mechatronics Laboratory.

El procedimiento que se realiza para ejecutar una simulación en tiempo real dentro del RTDS se presenta en el Algoritmo 1.

Algorithm 1 Load a Model Sequence

1: Start MATLAB® software on the Host PC.
2: Establish a communication channel via Ethernet with the Target PC.
3: Simulink Model SetUp and parameters.
4: Define Sample Time.
5: Generate Simulink model code.
6: Download code to the xPC Target.

Para poder efectuar dicho procedimiento y ejecutar una simulación en tiempo real, debe realizarse la configuración xPC (Target PC y Host PC) que se explica en la siguiente sección.

2.1. CONFIGURACIÓN xPC

2.1.1. Ajustes en el Target PC

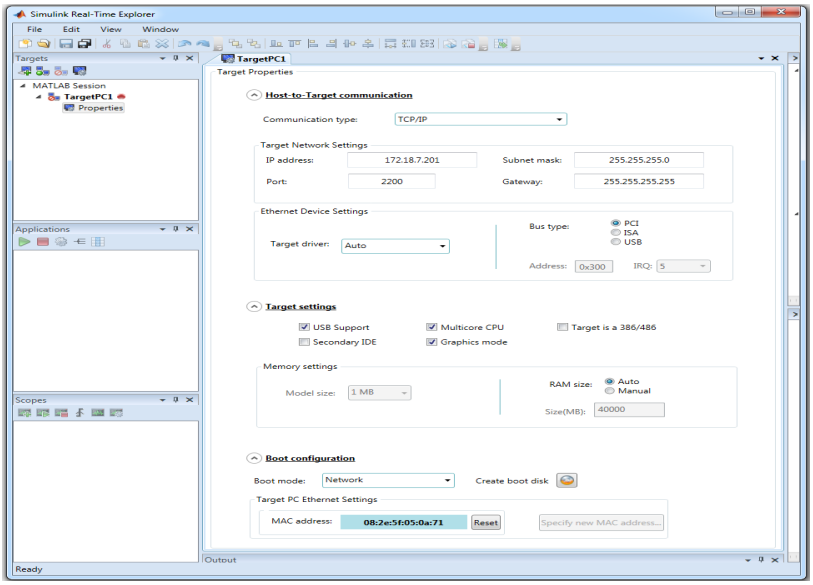
Teniendo en cuenta, que el Target PC no necesita ejecutar el OS para realizar simulaciones en tiempo real, se configura *Basic Input/Output System* (BIOS) de la siguiente manera:

- Deshabilitar Plug-and-Play (PnP): deshabilitar el Sistema operativo PnP permite que PCI BIOS configure las tarjetas PCI conectadas.
- Deshabilitar el modo ahorro de energía.
- Deshabilitar la detección de placa PCI 0xff: permite que la BIOS detecte y configure las tarjetas PCI.
- Deshabilitar el hyper-threading: con el fin de evitar que el rendimiento de la ejecución del simulador en tiempo real en el Target PC sea afectada.
- Habilitar el soporte de procesador Multicore.
- Habilitar el arranque Ethernet.
- Configurar el orden de arranque: se define el siguiente orden.
  - Boot disquete CD / DVD de arranque de sólo lectura de memoria (ROM).
  - Red de arranque dedicado.
  - Unidad flash disco duro extraíble.
  - Disco duro de arranque.

2.1.2. Conexión y arranque del Target PC

Para cargar el modelo en el Target PC, se ejecuta el comando "xpcexplr" en la ventana de comandos de MATLAB® del Host PC, obteniendo la interfaz mostrada en la Figura 2, en la parte superior izquierda, se pueden agregar los Target PC y en la parte derecha configurar propiedades como dirección, puerto y tipo de comunicación con el Target PC, entre otros. Adicionalmente, en la zona inferior izquierda se pueden modificar configuraciones de las pantallas mostradas en el Target PC.

Figura 2: xpcexplr Interface



Fuente: MATLAB®.

Se agrega un Target PC y se configuran los siguientes paneles:

- Host-to-Target comunication, habilitando la comunicación por el protocolo TCP/IP, también debe indicarse parámetros del Target PC como: dirección IP, Mascara de subred, puerto, adicionalmente, se selecciona el tipo de bus de salida PCI.
- Target Settings, permite seleccionar configuraciones adicionales, en este caso, se habilita el soporte USB, CPU Multicore y el modo gráfico.
- Boot configuration, se selecciona el modo de inicio por medio de la red y se escribe la dirección MAC del Target PC.

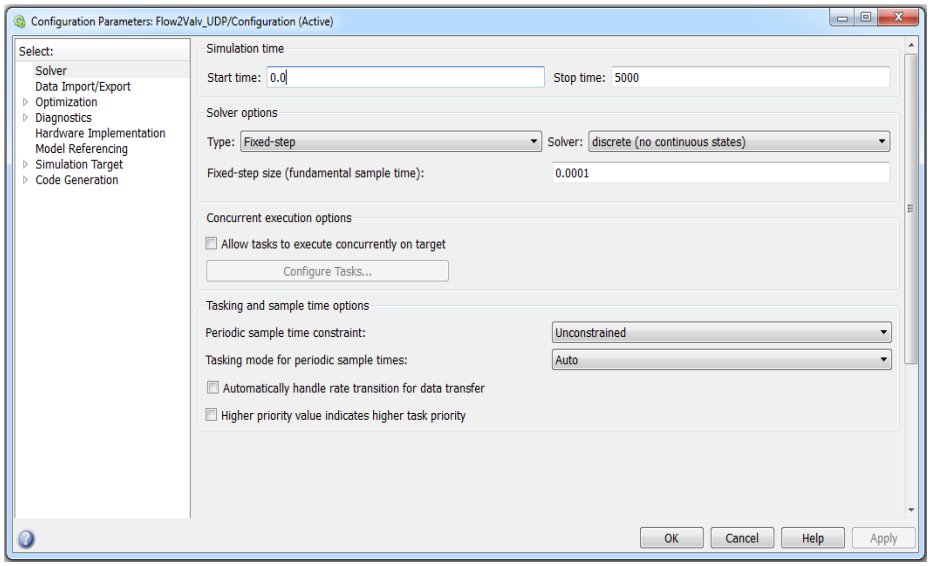
El código generado por el modelo en Simulink®, puede cargarse en el Target PC de dos maneras: primero, construyendo el modelo en Simulink® generando el código y cargándolo en el Target PC directamente, segundo, cargando el código con la extensión ".dlm" de un modelo existente desde la interfaz "xpcexplr".

2.1.3. Configuración del entorno de Simulink para la simulación en tiempo real

Después de elaborar el diseño del modelo en MATLAB/Simulink®, los parámetros del entorno deben configurarse en la caja de herramientas "Model Configuration Parameters" de Simulink® para crear el código ejecutable y poderlo descargar en el simulador en tiempo real. Los dos parámetros más importantes a configurar son el "Solver" y el "Code Generation"; La configuración recomendada para realizar una simulación en tiempo real se describe a continuación.

La configuración del "Solver" se muestra en la Figura 3, la sección de la izquierda corresponde a todos los parámetros del modelo, se selecciona el parámetro *solver* y se escoge la opción *fixed-step*.

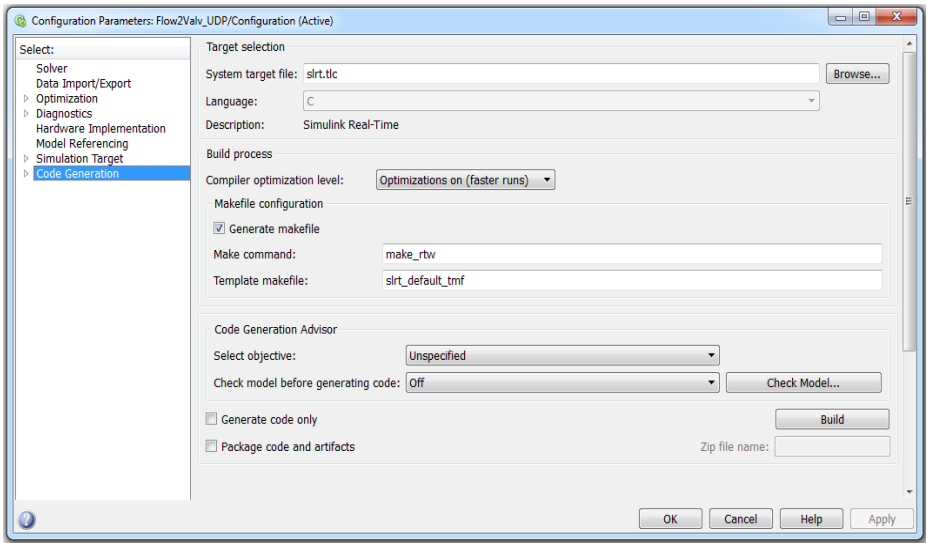
Figura 3: Solver



Fuente: MATLAB®.

La configuración del "Code Generation" se muestra en la Figura 4, en la opción *System target file* debe seleccionarse "grt.tlc".

Figura 4: Code Generation



Fuente: MATLAB®.

Luego de diseñar el modelo en MATLAB/Simulink® y realizar la configuración apropiada, se procede a compilar dicho modelo y el código generado se ejecuta en el Target PC con el fin de correr simulaciones en tiempo real.

Configurado el xPC Target, se procede a realizar la selección del caso de estudio a implementar en el RTDS, como se expone en el siguiente capítulo.

3. SELECCIÓN DEL CASO DE ESTUDIO

Se realiza la evaluación de tres casos de estudio para el desarrollo de un RTDS: primero, "Péndulo invertido", segundo, "Smart Grid (SG) integrada a un sistema fotovoltaico" y por último, "Unidad de Entrenamiento en Automatización (UEA)".

En la sección 3.1 se presentan los conceptos necesarios para tener un criterio en la selección del caso de estudio, en la sección 3.2 se encuentra el análisis del péndulo invertido, en la sección 3.3 el análisis de una SG integrada a un sistema fotovoltaico y en la sección 3.4 el análisis de la UEA, cada sección contiene la representación matemática del caso de estudio y los resultados al embeber el modelo en el simulador, posteriormente se determina la exactitud de la identificación del sistema y por último, en la sección 3.5 se presenta una tabla con los criterios para seleccionar el caso de estudio que será aplicado al RTDS.

3.1. MARCO TEÓRICO

3.1.1. Identificación de sistemas de primer orden

La representación de un sistema puede obtenerse mediante el desarrollo de modelos usando los balances de energía, balances de masa o leyes físicas y químicas, sin embargo, estos complicados y demorados procesos pueden ser omitidos en algunos casos (Lipták, 2013). Una buena aproximación de procesos dinámicos puede ser obtenida usando métodos de cálculo simplificados, como el **método de la curva de reacción**, éste proporciona un simple y rápido procedimiento para determinar un modelo de aproximación lineal del proceso. Esta técnica experimental está basada en la aplicación de un cambio en la variable manipulada (entrada del proceso) en lazo abierto y registrar la respuesta (salida del sistema) (Liptak, 2005).

Un alto porcentaje de todos los procesos químicos pueden ser modelados usando sistemas de primer orden más tiempo muerto, siendo el tiempo muerto el resultado de un retraso en la medida ya sea por consumo computacional, comunicación de los elementos o el tiempo de transporte requerido para que un fluido circule a través de una tubería (Corriou, 2013).

La Ecuación 3.1, es la representación de un sistema de primer orden más tiempo muerto.

$$\frac{Ke^{T_d S}}{TS+1} \tag{3.1}$$

Siendo:

- K : Ganancia en estado estacionario
- T : Constante de tiempo
- T_d : Tiempo muerto

Considerando que no se cuenta con una representación matemática del péndulo invertido y de la UEA, se implementa el método de la curva de reacción para hallar dicha representación.

3.1.2. Root Mean Square Error (RMSE)

El RMSE también conocido como error estándar de la regresión o error estándar de la estimación (Barreto y Howland, 2006), es la raíz cuadrada de la varianza de los residuos y mide la dispersión de los datos alrededor de la línea de regresión (Albert, 2003), este error es utilizado para evaluar el ajuste de un modelo utilizando los residuos del mismo (Dennison y Roberts, 2003) y tiene la propiedad de estar en las mismas unidades que la variable de respuesta. El RMSE es representado por la Ecuación 3.2.

$$\sqrt{\Sigma \frac{(Y_{pred} - Y_{ref})^2}{N}} \tag{3.2}$$

Siendo:

- Y_{pred} : Valor de predicción
- Y_{ref} : Valor real
- N : Numero de muestras

Teniendo en cuenta que se diseñan los modelos para el péndulo invertido y la UEA, se utiliza el RMSE para evaluar el ajuste de dichos modelos a los sistemas reales.

3.2. PÉNDULO INVERTIDO

El péndulo invertido ha sido una herramienta clásica en los laboratorios de control desde 1950, usado originalmente para ilustrar ideas de control lineal como la estabilidad de los sistemas (Cannon, 1967). Por su naturaleza no lineal, en la actualidad es usado para representar algunas de las ideas en el campo del control no lineal (K.J.Astrom y K.Furuta, 1996).

El péndulo invertido rotatorio, es uno de los experimentos más populares en el ámbito académico, ya que introduce conceptos intermedios de control y teorías relevantes para los retos que enfrentan áreas mecánicas y aeroespaciales en la vida real. El péndulo físico es presentado en la Figura 5, dicho péndulo está compuesto por un brazo plano con un pivote en un extremo y un eje metálico en el otro. El extremo del pivote está montado en la parte superior del eje del engranaje ubicado en la base rotatoria y es empleada para equilibrar una varilla en posición vertical girando o cambiando el ángulo de la misma. En cada servomotor, se encuentra acoplado un *encoder* de alta resolución encargado de medir los ángulos del brazo y la varilla.

Figura 5: Physical Pendulum

Fuente: Fabricante Quanser.

Las características principales son:

- Compatibilidad con MATLAB/Simulink®.
- Tres modos de operación: classical, grantry crane, self-erecting.
- Diseño de arquitectura abierta.

Figura 6: WorkStation Components Pendulum

Component	Description
Plant	• Servo Base Unit (SRV02) • Rotary Inverted Pendulum module
Controller Design Environment[1]	• Quanser QUARC® add-on for MATLAB®/Simulink® • Quanser RCP Toolkit add-on for NI LabVIEW™
Documentation	• ABET-aligned* instructor workbook** • ABET-aligned* student workbook** • User Manual** • Quick Start Guide
Real-Time Targets[1]	• Microsoft Windows®
Data Acquisition Board[2]	• Quanser Q2-USB
Amplifier	• Quanser VoltPAQ-X1
Others	• Complete dynamic model • Simulink® pre-designed controllers • LabVIEW™ pre-designed controllers

Fuente: Fabricante Quanser.

En la Figura 6 se detallan los componentes del péndulo y su descripción. Como se observa, el ambiente para el diseño del control del péndulo invertido está vinculado con MATLAB/Simulink® y las librerías Quarc®, permitiendo exportar y procesar los datos del péndulo invertido.

3.2.1. Modelamiento teórico

Para la descripción matemática del péndulo invertido, el fabricante propone el desarrollo del mismo por medio de las ecuaciones de movimiento, en específico el método de Lagrange. A continuación se representa el modelamiento.

Ecuación Euler-Lagrange

$$\frac{\partial^2 L}{\partial t \partial \dot{q}_i} - \frac{\partial^2 L}{\partial q_i} = Q_i \tag{3.3}$$

Las variables q_i son las coordinadas generales, entonces:

$$q(t)^T = [\Theta(t) \quad \alpha(t)] \tag{3.4}$$

Por tanto las velocidades son:

$$\dot{q(t)}^T = [\frac{\partial \Theta(t)}{\partial t} \quad \frac{\partial \alpha(t)}{\partial t}] \tag{3.5}$$

Con las coordinadas definidas, las ecuaciones de Euler-Lagrange quedan:

$$\frac{\partial^2 L}{\partial t \partial \dot{\Theta}} - \frac{\partial^2 L}{\partial \Theta} = Q_1 \tag{3.6}$$

$$\frac{\partial^2 L}{\partial t \partial \dot{\alpha}} - \frac{\partial^2 L}{\partial \alpha} = Q_2 \tag{3.7}$$

El Lagraniano del sistema se describe como:

$$L = T - V \tag{3.8}$$

Dónde T esla energía cinética del sistema, V la energía potencial. La fuerza que actúa sobre la sección rotatoria es:

$$Q_1 = \tau - B_r \dot{\Theta} \tag{3.9}$$

Mientras que la fuerza en el brazo del péndulo:

$$Q_2 = -B_r \dot{\alpha} \tag{3.10}$$

Encontrando el equivalente de las fuerzas, reemplazamos y las ecuaciones de movimiento se describen:

$$(m_p L_r^2 + \frac{1}{4} m_p L_p^2 \cos(\alpha)^2 + J_r)\ddot{\Theta} - (\frac{1}{2} m_p L_p L_r \cos(\alpha))\ddot{\alpha} + \frac{1}{2_p} L_p^2 \sin(\alpha)\cos(\alpha))\dot{\Theta}\dot{\alpha}$$

$$+(\frac{1}{2} m_p L_p L_r \sin(\alpha))\dot{\alpha}^2 = \tau - B_r \dot{\Theta} \tag{3.11}$$

$$-\frac{1}{2} m_p L_p L_r \cos(\alpha)\ddot{\Theta} + (J_p + \frac{1}{4} m_p L_p^2)\ddot{\alpha} - \frac{1}{4} m_p L_p^2 \cos(\alpha)\sin(\alpha)\dot{\Theta}^2$$

$$\frac{1}{2_p} L_p g \sin(\alpha) = -B_r \dot{\alpha} \tag{3.12}$$

La linealización de las ecuaciones se realiza utilizando el Jacobiano, teniendo el sistema linealizado el sistema se plantea el espacio de estados:

$$\dot{x} = Ax + Bu \tag{3.13}$$

Y

$$y = Cx + Du \tag{3.14}$$

Para el péndulo se define:

$$x^T = [\Theta \quad \alpha \quad \dot{\Theta} \quad \dot{alpha}] \tag{3.15}$$

Y

$$y^T = [x1 \quad x2] \tag{3.16}$$

Las salidas son las medidas en el ángulo del servo Theta Θ y el del brazo alpha α

$$\mathrm{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

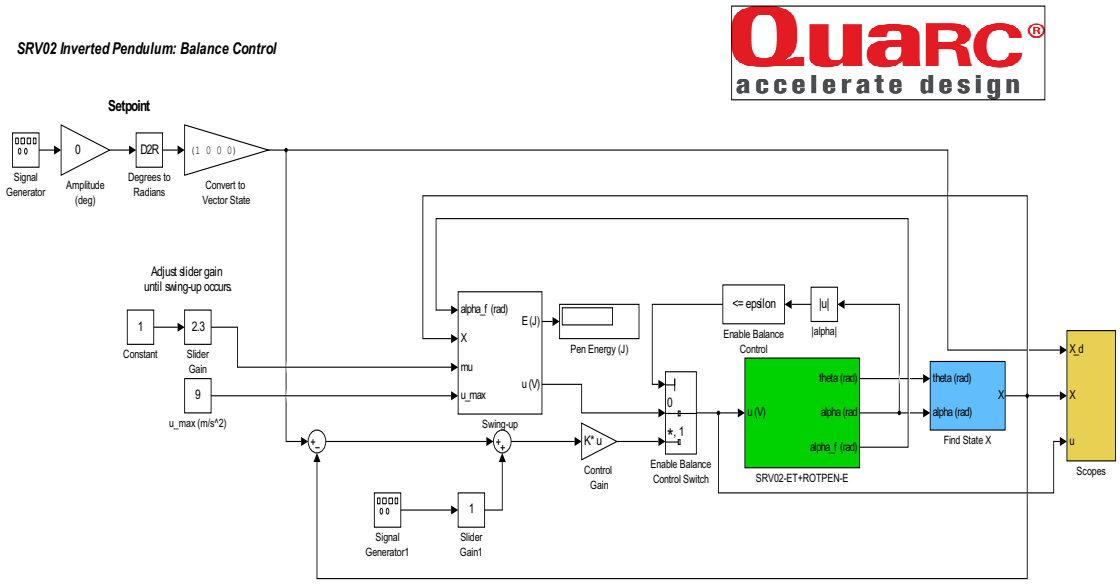
$$\mathrm{D} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Las velocidades se pueden obtener mediante filtros derivativos u observadores de estados.

3.2.2. Identificación de las variables Alpha y Theta

Para el manejo y posterior identificación del péndulo se usa un modelo suministrado por Quanser® implementado en MATLAB/Simulink®, dicho modelo permite la adquisición de datos desde el entorno de Simulink® y puede ser visualizado en la Figura 7.

Figura 7: Model Identification Pendulum

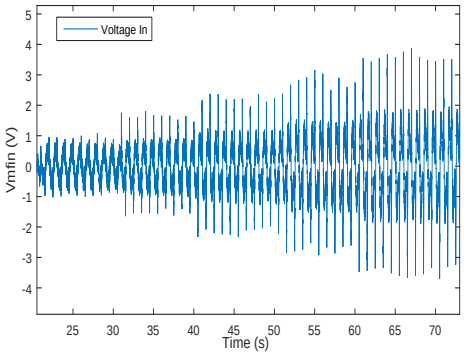


Fuente: Fabricante Quanser.

El modelo incluye dos técnicas de control: la primera, realiza un balanceo para llevar el péndulo a su punto de equilibrio y la segunda, estando en su punto de equilibrio lo mantiene en un rango de operación dado en grados. La excitación del sistema se realiza aplicando un patrón de perturbación mientras el péndulo invertido opera dentro del rango de equilibrio.

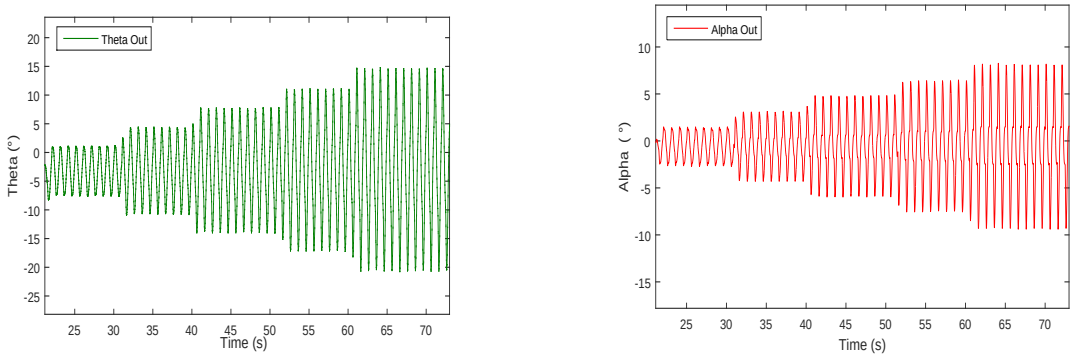
Se registran los valores de la señal de entrada al sistema (Voltaje) como se contemplan en la Figura 8. Adicionalmente, las variables del péndulo a medir como salidas son Alpha y Theta, visualizadas en las Figuras 9a y 9b, las cuales corresponden a los ángulos del pivote y la varilla respectivamente.

Figura 8: Voltage In for Pendulum



Fuente: Autores.

Figura 9: Pendulum Signals



(a) Theta **(b)** Alpha

Fuente: Autores.

Se utiliza el *toolbox* de identificación "ident" de MATLAB®, para hallar una representación matemática de las señales Alpha y Theta, obteniendo las funciones de transferencia apreciadas en las Ecuaciones 3.17 y 3.18.

$$TF_{Alpha} = \frac{-12{,}34S^4 + 899{,}9S^3 - 648{,}2S^2 + 3{,}719e^4S - 1{,}323e^4}{S^5 + 16{,}39S^4 + 118{,}5S^3 + 874{,}8S^2 + 2792S + 7551} \tag{3.17}$$

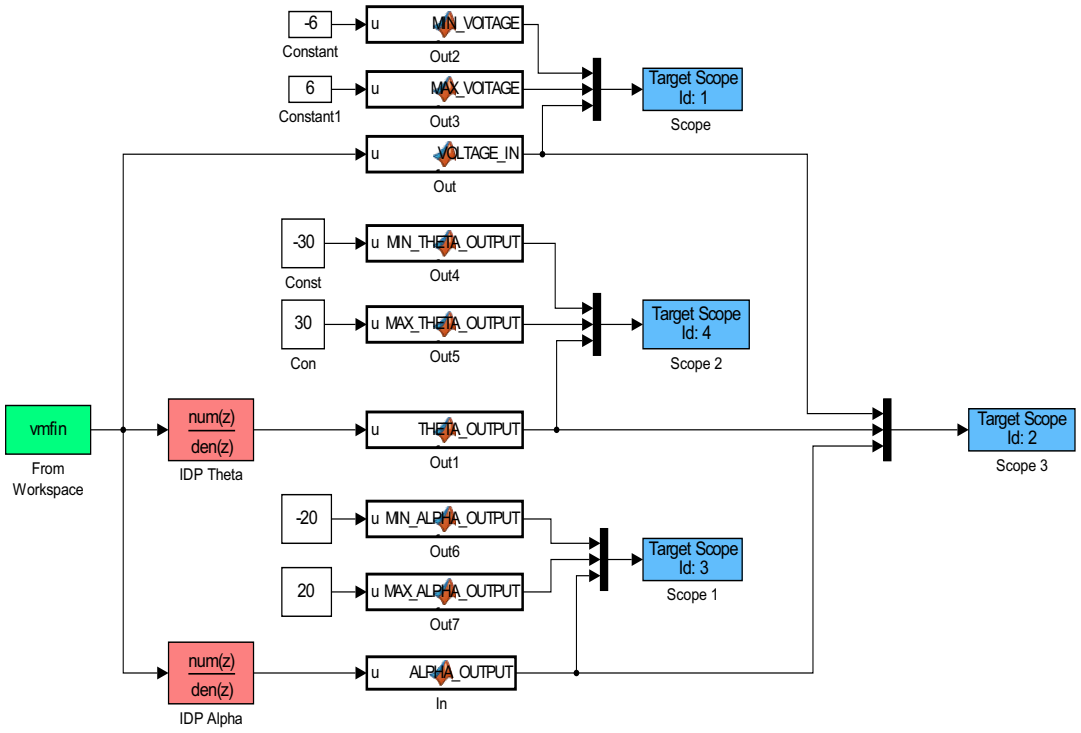
$$TF_{Theta} = \frac{962S^3 + 2{,}748e^4S^2 + 1{,}58e05S + 518{,}6}{S^5 + 37{,}3S^4 + 537{,}4S^3 + 2156S^2 + 2311S + 668{,}9} \tag{3.18}$$

Estas funciones de transferencia se implementan en el modelo de Simulink® para ser embebido en el Target PC.

3.2.3. Modelo embebido en el xPC

Para realizar la simulación en tiempo real del péndulo se diseña un modelo en Simulink® donde se importa la señal de entrada que alimenta a las funciones de transferencia "IDP Theta" e "IDP Alpha". El modelo a embeber en el simulador se expone en la Figura 10.

Figura 10: Pendulum Model



Fuente: Autores.

Como se menciona anteriormente se embebe el modelo que representa el comportamiento del péndulo y los resultados obtenidos se visualizan en la interfaz del xPC, el cual es configurado con 3 pantallas que muestran la entrada y las dos salidas en tiempo real, esta interfaz está disponible para el usuario y es mostrada en la Figura 11.

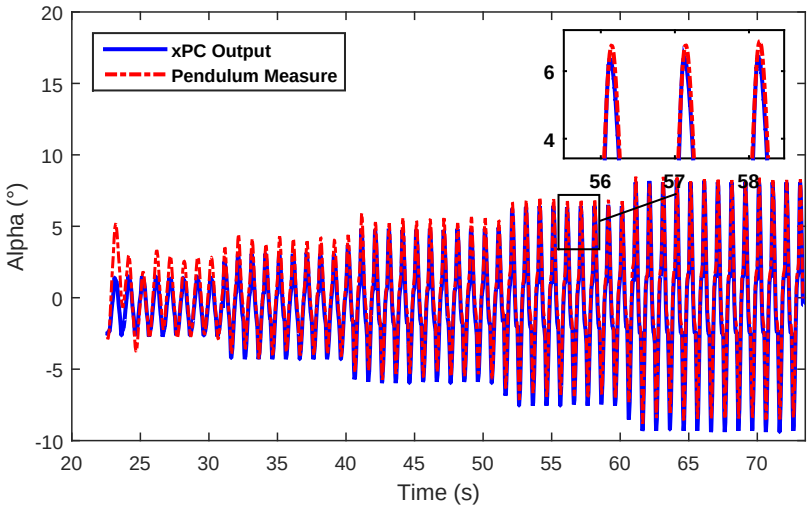
Figura 11: Real Time Simulation of Pendulum

Fuente: Autores.

3.2.4. Análisis de las identificaciones del péndulo

En la Figura 12, se detalla la respuesta de la identificación de la variable Alpha y la respuesta real del sistema, aquí se evidencia que al comenzar la excitación la variable real presenta un desorden en el patrón que dura aproximadamente 3 s y luego se estabiliza, en el recuadro de *zoom*, se visualiza que en los instantes positivos la identificación coincide con el comportamiento real de la variable, mientras que en los negativos existe un error.

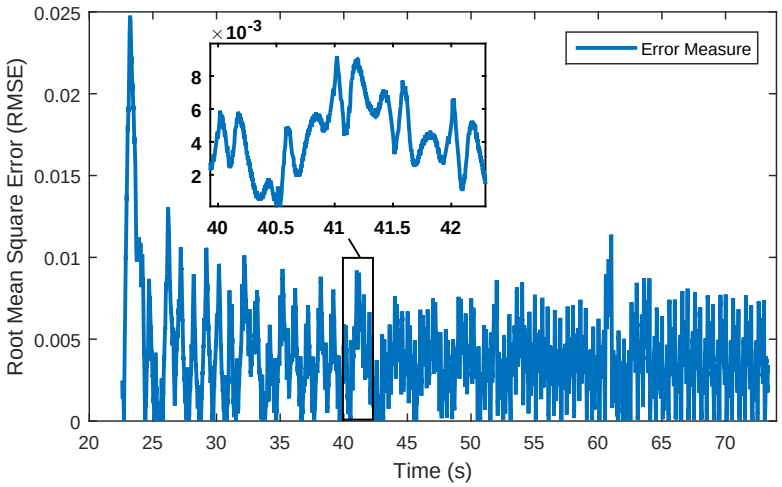
Figura 12: Alpha Identification Vs. Alpha Real



Fuente: Autores.

El RMSE de la identificación para la variable Alpha se expone en la Figura 13, se aprecia que la identificación no presenta un error constante, es decir, para diferentes puntos de operación cambia su rango. En el segundo 40 el RMSE presenta un valor de $\pm$ 0.001°, en el segundo 41.2 el RMSE arroja un valor de $\pm$0.008°, para una simulación ésta variación en el error no hace la medida confiable.

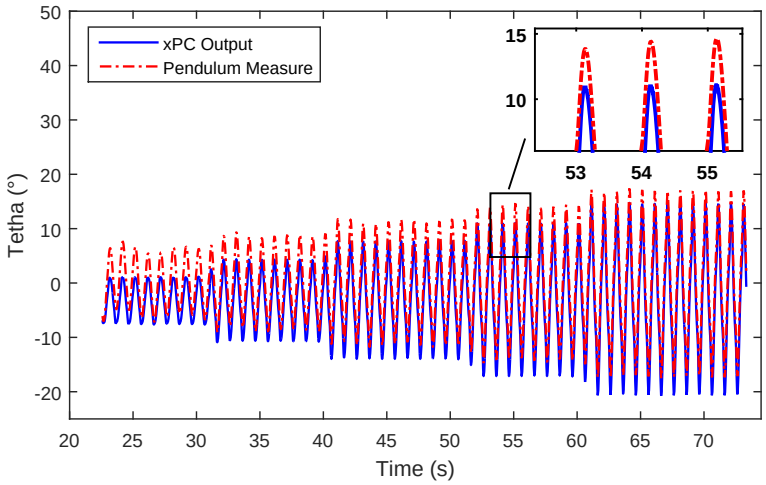
Figura 13: Alpha RMSE



Fuente: Autores.

La comparación entre el comportamiento real de la variable Theta y su identificación se expone en la Figura 14, se puede analizar que la función de transferencia no tiene las mismas amplitudes que la variable real a pesar de que si coincide con la frecuencia de respuesta.

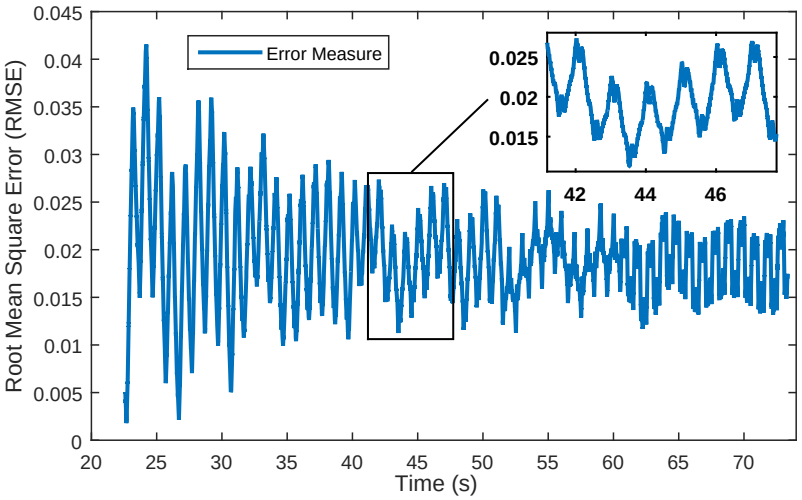
Figura 14: Theta Identification Vs. Theta Real



Fuente: Autores.

El RMSE para la variable Theta es presentado en la Figura 15, se visualiza que posee un offset de 0.02° en todo el tiempo de muestreo, para la simulación del péndulo invertido se requiere un error más cercano a cero ya que es un sistema con alta resolución.

Figura 15: Theta RMSE



Fuente: Autores.

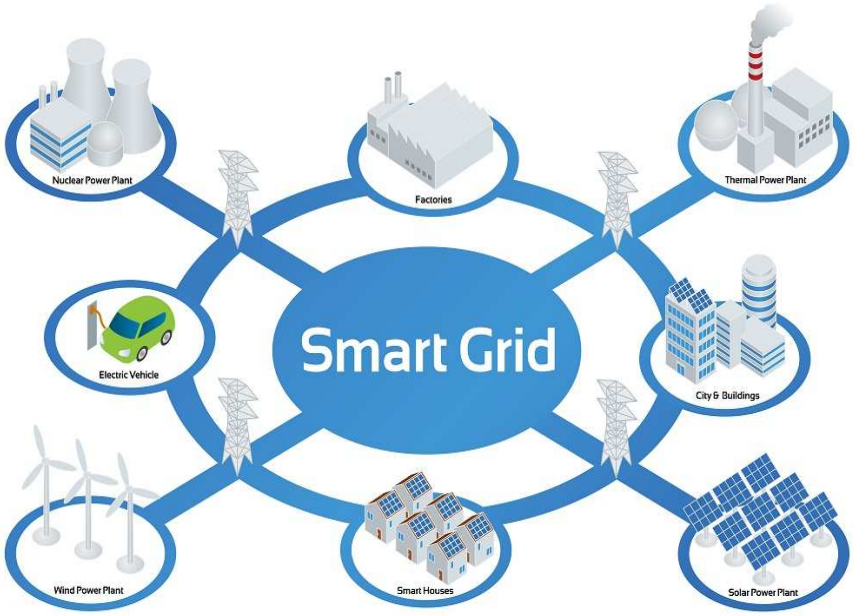
Teniendo como referencia el modelamiento realizado en la sección 3.2.1, se puede concluir que la identificación de las variables correspondientes al angulo del pivote (Alpha) y al angulo de la varilla (Theta), no representan de manera acertada la dinámica del sistema, ya que las variables mencionadas no se encuentran interrelaciones como se visualiza en el modelo mostrado en la Figura 10.

3.3. SMART GRID (SG) INTEGRADA A UN SISTEMA FOTOVOLTAICO

Una SG es una red eléctrica destinada a entregar energía de manera inteligente desde los puntos de generación a los consumidores, quienes hacen parte activa de la SG mediante la lectura de su consumo y la configuración a sus necesidades(Sevilla y Fernández, 2013). Las SG incluyen una entrega de energía bidireccional, es decir que en caso de que un consumidor tenga un medio de generación como energía eólica o energía solar puede aportar a la red de distribución (Siano, 2014).

La eficiencia de la red es soportada por un sistema de información integrado, trabajando en conjunto con elementos de medición automatizados y comunicados. En la Figura 16 se observa que las islas de consumo y generación interactúan de manera dinámica y bidireccional.

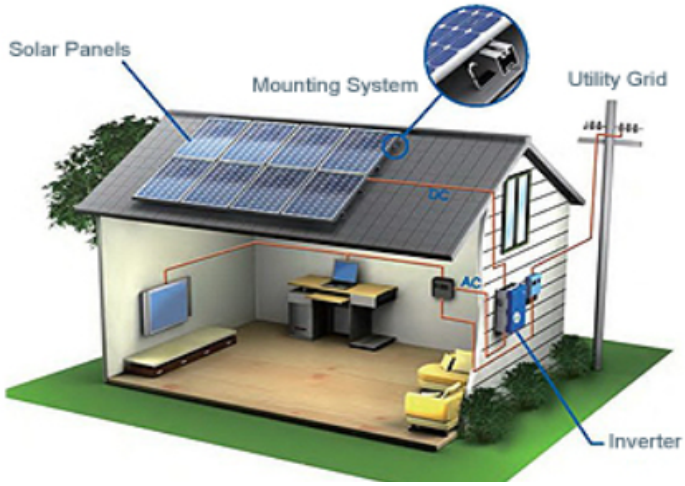
Figura 16: Smart Grid



Fuente: Equipo ENISA.

Un sistema fotovoltaico convierte energía solar en energía eléctrica, estos sistemas pueden ser de dos tipos dependiendo de su propósito: los sistemas locales que almacenan la energía para su uso y los sistemas *On-Grid* que permiten enviar la energía eléctrica hacia la red de distribución (Dunlop, 2004).

Figura 17: Structure of On-Grid Photovoltaic System



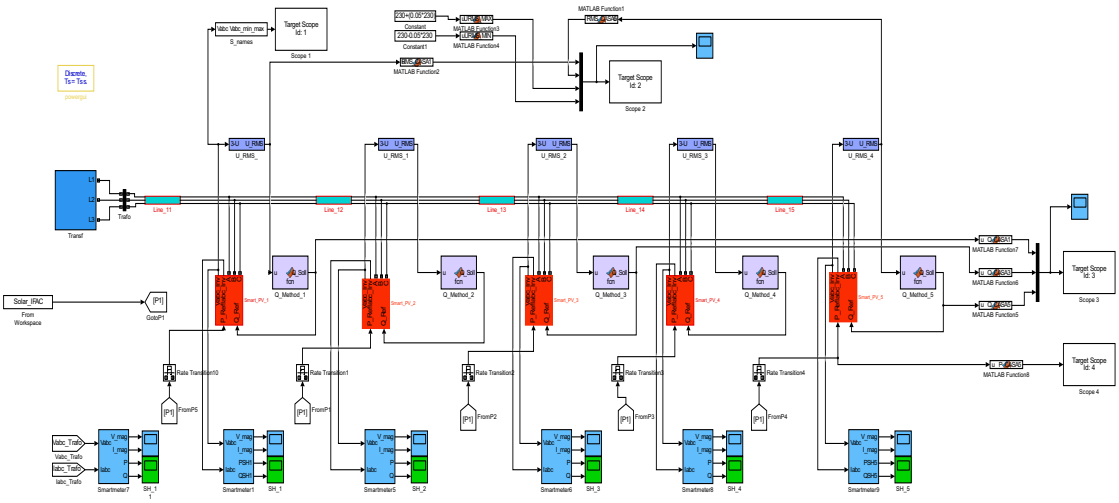
Fuente: Empresa Assawsaneh.

En la Figura 17 se detalla la estructura típica de un sistema fotovoltaico integrado a la red eléctrica. Este medio de energía renovable ofrece varias ventajas económicas y ambientales aunque propone retos en la integración de sistemas, debido a que la distribución de energía desde el sistema fotovoltaico a la red requiere un acondicionamiento.

3.3.1. Modelo a Embeber de la SG

El modelo definido para este caso de estudio se compone de una SG de 5 casas, cada una cuenta con la lectura de la celda fotovoltaica y un inversor de potencia ya que la señal que emite el inversor no es AC, cabe resaltar que cada inversor cuenta con una estrategia de control para no tener sobrecargas a la red, esta configuración se muestra en la Figura 18.

Figura 18: Smart Grid Model



Fuente: Autores.

3.3.2. Simulación en tiempo real de la SG

Después de implementar el modelo, se realiza la simulación en tiempo real, es requerido como señal de entrada al simulador un perfil del sol que estimula el modelo, este perfil también es simulado.

Como se observa en la Figura 19 la pantalla 1 visualiza la señal de entrada, la pantalla 2 el voltaje RMS de las 5 casas, donde se puede detallar el efecto de la celda en la red, en la pantalla 3 se contempla el voltaje en cada fase de la casa y en la pantalla 4 se identifica la pérdida de potencia.

Figura 19: xPC Smart Grid

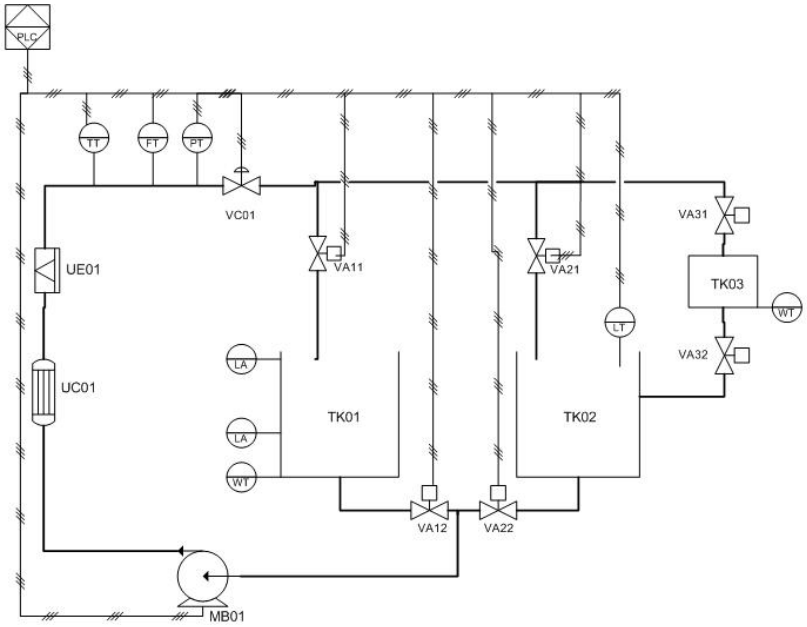
Fuente: Autores.

3.4. UNIDAD DE ENTRENAMIENTO EN AUTOMATIZACIÓN (UEA)

La UEA es una implementación que representa un proceso industrial y posee un enfoque académico, este prototipo cuenta con un bombeo hidráulico en circuito cerrado cuyos componentes de instrumentación son: sensores, actuadores y controladores. Los elementos que componen la UEA permiten la calibración de sensores, configuración de actuadores y la manipulación de variables como flujo, presión, temperatura, peso y nivel. Para el análisis, se toman específicamente las variables de flujo y presión ya que operan en la UEA de manera correcta.

La Figura 20 es el diagrama de tubería e instrumentación P&ID de la UEA, cuenta con dos tanques en los cuales recircula el fluido. El llenado y vaciado de los mismos se controla con cuatro válvulas *On/Off*, las válvulas "VA11" y "VA21" permiten el llenado, las válvulas "VA12" y "VA22" permiten el vaciado de los tanques "TK01" y "TK02" respectivamente.

Figura 20: P&ID UEA



Fuente: Autores.

Los elementos y características que componen la UEA se explican en la Tabla 1.

Tabla 1: UEA Components.

Elemento	Características
Sensores	**Flujo:** Sensor SITRANS FM MAG 5100W. **Presión:** Sensor SITRANS P220
Actuadores	**Variador de frecuencia:** SINAMICS V20 1AC200-240V, 47-63HZ, 0,75KW/ 0,75HP. **Motor:** Trifásico 3600RPM, 0,75 HP, corriente 2,4 / 1,2 A 220 /440. **Valvulas:** 4 unidades On-Off.
Controlador	**PLC:** SIEMENS, 2 Analog/Inputs, 1 Analog/Output, 4 Digital/Outputs.

El actuador que genera el desplazamiento del fluido en la UEA es una bomba hidráulica "MB01". Para controlar las variables de flujo y presión es necesaria la realimentación de sus medidas, por medio de un sensor para cada una, la medida es leída y procesada por un PLC que envía la señal de control al actuador.

3.4.1. Métodos de obtención de modelos

Existen dos métodos principales para obtener el modelo de un sistema: **Modelamiento teórico** e **Identificación**. El modelamiento teórico tiene un campo de aplicación restringido a procesos muy sencillos o sistemas que no requieren alto nivel de exactitud, por otra parte, la estructura del modelo obtenido a partir de las leyes físicas posee un conjunto de parámetros desconocidos y solo pueden ser determinados experimentalmente (J. Schoukens, 1991). Es por esta razón que surge la necesidad de recurrir al método de identificación de sistemas.

3.4.1.1. Modelamiento teórico

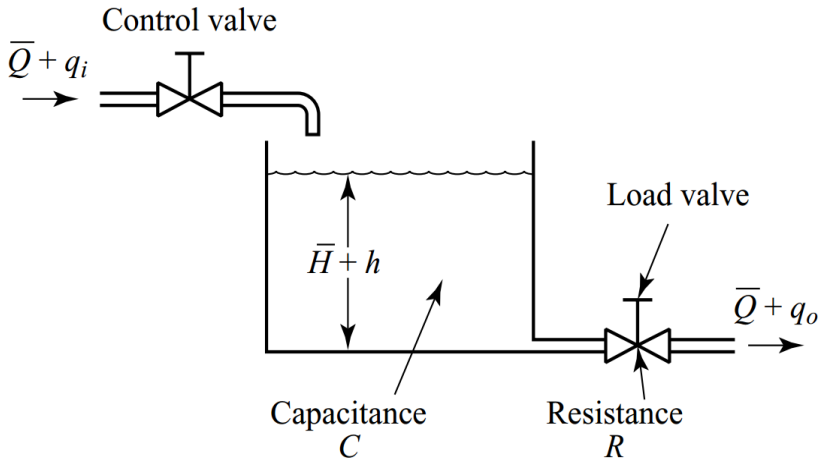
Este modelamiento es un método analítico, en el que se recurre a leyes físicas básicas para describir el comportamiento dinámico de un proceso.

A continuación, se realizara el modelamiento matemático para las variables flujo y presión de la UEA.

Sistema de fluidos

Un proceso típico de almacenamiento de líquidos es mostrado en la Figura 21, donde q_i y q_0 son caudales volumetricos de entrada y salida respectivamente, las válvulas son dispositivos de restricción y proporcionan una resistencia hidraulica R (Seborg, 2004).

Figura 21: Fluid System



Fuente: (Ogata y Yang, 2010)

Existen dos tipos de flujo en los sistemas líquidos: laminar y turbulento. El flujo laminar se produce cuando la velocidad del fluido es relativamente lenta y el liquido fluye en capaz, generalmente se produce solo con materiales muy viscosos (Hughes, 2002). El flujo turbulento ocurre cuando la velocidad del fluido es relativamente alta y la velocidad del

líquido en cualquier punto varía irregularmente, el caudal de estado estacionario esta dado por la Ecuación 3.19 (Sharma, 2013).

$$Q = K\sqrt{H} \tag{3.19}$$

Donde:
Q = Flujo en estado estacionario, m^3/sec
K = Coeficiente de flujo, $m^{2,5}/sec$
H = Cabeza de estado estacionario, m
h = desviación de la cabeza de estado estacionario, m

La resistencia R para el flujo turbulento es obtenido por medio de la Ecuación 3.20.

$$R = \frac{dH}{dQ} \tag{3.20}$$

A partir de la Ecuación 3.19 es posible obtener:

$$dQ = \frac{K}{2\sqrt{H}}dH \tag{3.21}$$

$$\frac{dH}{dQ} = \frac{2\sqrt{H}}{K} = \frac{2\sqrt{H}\sqrt{H}}{Q} = \frac{2H}{Q} \tag{3.22}$$

$$R = \frac{2H}{Q} \tag{3.23}$$

En muchos casos prácticos, no se conoce el valor del coeficiente K en la ecuación 3.19, que depende del coeficiente de flujo y del área de restricción.

Es posible determinar la resistencia mediante el trazado de la relación entre la cabeza y el flujo, La curva de velocidad basada en datos experimentales y midiendo la pendiente de la curva en la condición operativa, definiendo una pequeña desviación de la cabeza del valor de estado estacionario como h y el pequeño cambio correspondiente del caudal como q, se puede obtener la Ecuación 3.24 (Ogata y Yang, 2010).

$$R = \frac{h}{q_0} \tag{3.24}$$

La capacitancia C de un tanque se define como el cambio en la cantidad de líquido almacenado necesario para causar un cambio de unidad en el potencial. (El potencial es la cantidad que indica el nivel de energía del sistema).

Un sistema puede considerarse lineal si el flujo es laminar. Incluso si el flujo es turbulento, el sistema puede ser linealizado si los cambios en las variables se mantienen pequeños

(Ogata y Yang, 2010). Basándose en el supuesto de que el sistema es lineal o linealizado, la ecuación diferencial de este sistema puede obtenerse de la siguiente manera: Dado que el flujo de entrada menos el flujo durante el intervalo de tiempo pequeño dt es igual a la cantidad adicional almacenada en el tanque como se evidencia en la Ecuación 3.25.

$$Cdh = (q_i - q_0)dt \tag{3.25}$$

Reemplazando la Ecuación 3.24 en la Ecuación 3.25.

$$RC\frac{dh}{dt} + h = Rq_i \tag{3.26}$$

Sabiendo que RC es la constante de tiempo del sistema. Tomando las transformadas de Laplace de ambos lados de la Ecuación 3.26, asumiendo la condición inicial cero, se obtiene:

$$(Rcs + 1)H(s) = RQ_i(s)) \tag{3.27}$$

$$\frac{H(s)}{Q_i(s)} = \frac{R}{RCs + 1} \tag{3.28}$$

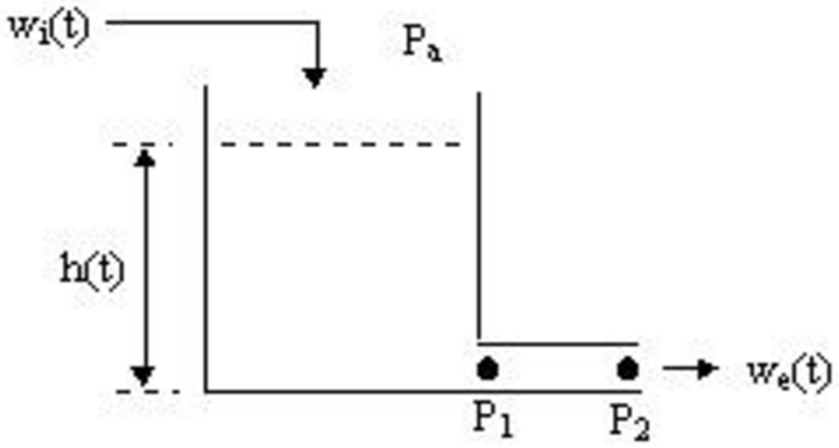
Considerando que la entrada al sistema es Q_i y la salida es Q_0, la función de transferencia que describe el comportamiento del sistema esta dada por la Ecuación 3.29.

$$\frac{Q_0(s)}{Q_i(s)} = \frac{1}{RCs + 1} \tag{3.29}$$

Sistema de presión

Considerando la dinámica de un tanque de retención, donde el caudal de salida se determina a partir de la presión del fluido en el tanque, como se muestra en la Figura 22.

Figura 22: Fluid System



Fuente: (White, 1997)

Donde la presión de salida es P_a y el caudal de entrada es la función de fuerza de entrada. La variable de estado es la altura del agua en el tanque $h(t)$ y la salida de interés es el caudal másico de salida w_e, a partir de un balance de masas se obtiene la Ecuación 3.30.

$$\frac{d}{dt}m = w_i - w_e \tag{3.30}$$

La Ecuación 3.31 representa la formula de la masa, donde ρ es la densidad, Ah el volumen del tanque.

$$m = \rho A h \tag{3.31}$$

$$\frac{d}{dt}m = \rho A\frac{d}{dt}h + Ah\frac{d}{dt}\rho = \rho A\frac{d}{dt}h \tag{3.32}$$

El caudal másico de salida puede ser escrito en términos de w_e, siendo v_e la velocidad de salida como se visualiza en la Ecuación 3.33.

$$w_e = \rho A_e v_e \tag{3.33}$$

Realizando un balance de energía, Figura 23.

Figura 23: Balance de energía

$$\begin{bmatrix}\text{change in}\\ \text{energy}\\ \text{storage}\end{bmatrix} = \begin{bmatrix}\text{change in}\\ \text{internal}\\ \text{energy}\end{bmatrix} + \begin{bmatrix}\text{change in}\\ \text{kinetic}\\ \text{energy}\end{bmatrix} + \begin{bmatrix}\text{change in}\\ \text{potential}\\ \text{energy}\end{bmatrix} + \begin{bmatrix}\text{change in}\\ \text{flow}\\ \text{work}\end{bmatrix}$$

Fuente: (White, 1997)

Se obtiene la Ecuación 3.34.

$$\frac{d}{dt}(me) = w(u_1 - u2) + \frac{w}{2}(v_1^2 - v_2^2) + wg(z_1 - z_2) + \frac{w}{\rho}(p_1 - p_2) \tag{3.34}$$

Asumiendo cambio insignificantes en la energía, la ecuación de conservación de la energía indica (White, 1997):

$$v_e = \sqrt{2gh} \tag{3.35}$$

Reemplazando en la Ecuación 3.30.

$$\rho A\frac{d}{dt}h = -\rho A_e\sqrt{2gh} + w_i \tag{3.36}$$

La Ecuación 3.36, define que este sistema de primer orden es no lineal. Esta expresión dice

que la velocidad de cambio de masa dentro del tanque es igual al caudal másico de entrada menos el caudal másico de salida, donde la masa en el tanque y el caudal de salida están escritos tanto en términos de la altura del fluido en el tanque en el instante t.

La restricción al paso del flujo R, se define como el cambio en la diferencia de nivel dividido en el cambio del caudal, como se evidencia en la Ecuación 3.37.

$$R(h) = \frac{h}{\rho_e A_e \sqrt{2gh}} \tag{3.37}$$

La capacitancia C es la cantidad de liquido almacenado necesario para causar un cambio de unidad en el potencial y es el cambio en el liquido almacenado dividido en el cambio en la cabeza, Ecuación 3.38.

$$C = \frac{\rho A h}{h} = \rho A \tag{3.38}$$

Sustituyendo la Ecuación 3.37 y 3.38 en la Ecuación 3.36, se obtiene.

$$C\frac{d}{dt}x(t) = \frac{-1}{R}x(t) + u(t) \tag{3.39}$$

$$\frac{d}{dt}x(t) = \frac{-1}{RC}x(t) + \frac{1}{C}u(t) \tag{3.40}$$

Donde $x(t) = h(t)$ y $u(t) = w_i(t)$.

Este sistema no lineal, puede ser linealizado introduciendo las siguientes variaciones:

$$h(t) = h_o(t) + \delta h(t) \tag{3.41}$$

$$w_i(t) = w_{io}(t) + \delta w_i(t) \tag{3.42}$$

La Ecuación 3.43 representa a el sistema linealizado.

$$\frac{d}{dt}\delta h = -\frac{A_e}{A}\frac{g}{\sqrt{2gh_o}}\delta h + \frac{1}{\rho A}\delta w_i \tag{3.43}$$

Siendo $x(t) = \delta h(t)$, $u(t) = \delta w_i$, y $R = \frac{\sqrt{2gh-0}}{\rho g A_e}$, $C = \rho A$.

Reemplazando en la Ecuación 3.40.

$$C\frac{d}{dt}\delta h = -\frac{1}{R}\delta h + \delta w_i \tag{3.44}$$

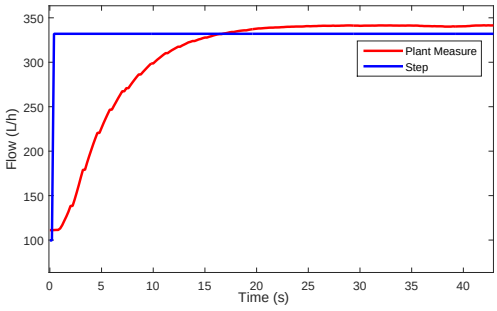
Las Ecuaciones 3.44 y 3.29 representa las variables flujo y presión de la UEA, sin embargo no incluyen parámetros del sistema como: propiedades del fluido y la tubería, por este motivo es necesario aplicar el método de identificación.

3.4.2. Identificación de las variables Flujo y Presión

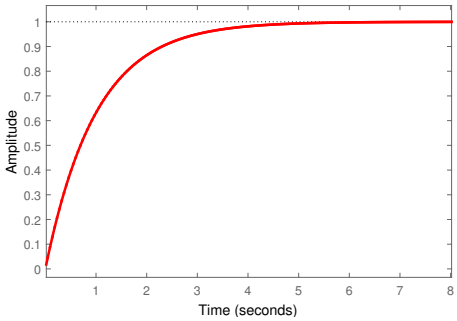
Considerando que no se cuenta con la representación matemática de la UEA, se hace necesario hallar una para implementar en el simulador, la teoría define que los sistemas físicos se pueden representar de diferentes maneras como: ecuaciones diferenciales, espacio de estados o función de transferencia. La complejidad matemática para determinar un espacio de estados o modelar una ecuación diferencial que describa de manera acertada el comportamiento del sistema, deja como mejor opción en este caso identificar una función de transferencia, siendo ésta una relación entre la salida y la entrada del sistema por medio de la transformada de Laplace (Bemporad, 2011).

Como se menciona anteriormente para realizar la identificación se requiere tener los valores de la entrada y de la salida del sistema, por tanto, se estimula el sistema con una entrada tipo escalón al 60 % del punto máximo de operación del actuador, con el fin de registrar el comportamiento de la salida. Comparando la Figura 24a y la Figura 24b se evidencia que el sistema se comporta como un modelo de primer orden. Debido a la ubicación de los sensores y las dimensiones de la tubería, se halla un retardo respecto a la señal de entrada, por ende, se identifica el sistema con el método de aproximación a un modelo de primer orden más tiempo muerto (Corriou, 2013).

Figura 24: Firts Orden Comparison



(a) Flow step Response



(b) Step response of a first-order system

Fuente: Autores.

Cuando se tiene el registro de las señales de entrada y salida, se aplica el método de identificación de primer orden más tiempo muerto, descrito en el Algoritmo 2 (Alfaro, 2007).

Algorithm 2 Firt-Order with Dead-Time identification

```
procedure DATA(dy, du)
    dy25 = 0,25 * dy
    dy75 = 0,75 * dy
    Kp = dy/du
    t = 0,9102 * (t75 − t25)
    tm = (1,2620 * t25) − (0,2620 * t75)
    Gs = (kp)/(t * s + 1)
    [N, D] = pade(tm, 1)
    Gr = tf(N, D)
    TF = Gs * Gr
    return TF
```

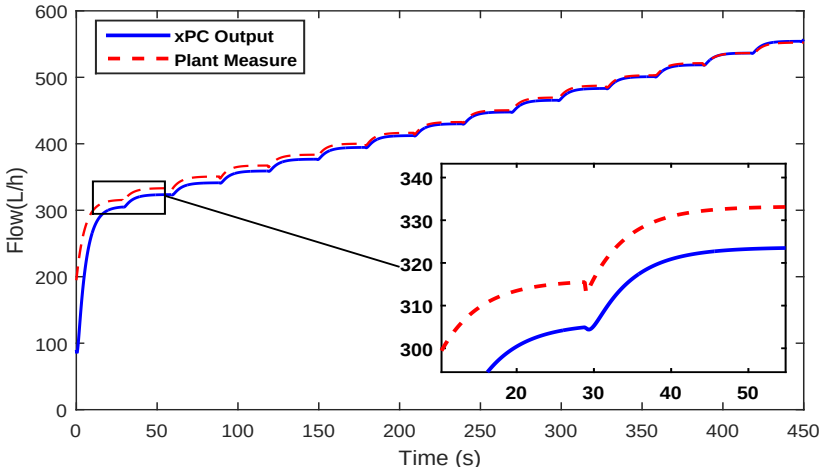
Como resultado del método de identificación se obtiene la ecuación 3.45 que describe la función de transferencia del sistema para la variable de flujo en el 80 % de la potencia del actuador.

$$TF80\,\% = \frac{-0{,}01774S + 0{,}02621}{5{,}121S^2 + 8{,}566S + 1{,}477} \tag{3.45}$$

Las limitantes físicas del sistemas como tubería, posición de los sensores, vibración, desgaste, cambios en las propiedades del fluido, se ven reflejadas en el funcionamiento de la planta, presentando un comportamiento distinto entre puntos de operación.

Se aplica a la planta real y la función de transferencia *TF80 %*, una escalera con rangos de 12 % al 94 % de la potencia del actuador, obteniendo los resultados de la Figura 25.

Figura 25: Flow Stair Response One Transfer Function-Real Plant

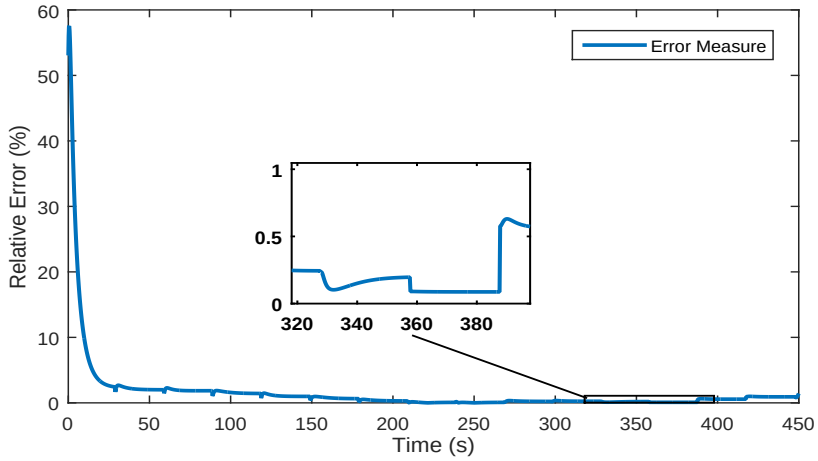


Fuente: Autores.

En la Figura 26 se detalla que la respuesta de la función de transferencia presenta error en todos los rangos excepto en el perteneciente al 80 %, a medida que se aleja del rango

identificado el error incrementa, por lo que se hace necesario realizar la identificación en todos los rangos de operación.

Figura 26: Flow Stair Response Error One Transfer Function-Real Plant



Fuente: Autores.

La UEA puede recircular el fluido en dos configuraciones: con una válvula de llenado abierta o con las dos válvulas de llenado abiertas, para cada configuración las variables se comportan de manera distinta, por lo que se realiza una identificación en las siguientes configuraciones:

- Flujo con 1 válvula de llenado.
- Flujo con 2 válvulas de llenado.
- Presión con 1 válvula de llenado.
- Presión con 2 válvulas de llenado.

3.4.3. Condiciones iniciales

Con el fin de asegurar que el comportamiento del simulador corresponda con el comportamiento real de la planta UEA, se deben cumplir las siguientes condiciones iniciales:

- Nivel total en el sistema: 14 L.
- Apertura de la válvula proporcional en el 100 %.
- Mínimo una válvula de vaciado abierta.
- Mínimo una válvula de llenado abierta.
- La consigna mínima de la bomba para que haya flujo en el sistema es del 12 %.
- No exceder una consigna para la bomba del 94 % ya que el fluido se comporta de manera diferente (Espuma).

3.5. ANÁLISIS DE RESULTADOS

Para realizar la selección del caso de estudio a implementar en el simulador, se definen los criterios que se muestran en la Tabla 2, evidenciando que en el caso de estudio del péndulo invertido no se logro obtener una representación acertada de la dinámica del sistema. Por otra parte, la SG requiere más entradas y salidas de las disponibles, lo que genera la necesidad de obtener una *Data Acquisition* (DAQ) que se ajuste a las *In/Out* necesarias, además de la implementación de una etapa de potencia de alto costo. Por tal motivo, se escoge como caso de estudio a implementar en el RTDS la UEA, puesto que se tienen a disposición los elementos para el desarrollo del simulador los cuales son: el controlador (PLC) y una DAQ, esta última suple los requerimientos de *In/Out*. Debido a que la potencia de salida de la DAQ es suficiente para la lectura de las señales por el PLC, no es necesaria la implementación de una etapa de potencia. Cabe aclarar, que el TET en los tres casos de estudio es suficiente para realizar la simulación en tiempo real.

Tabla 2: Selection Criteria.

Modelo	In/Out DAQ	Procesamiento DAQ	Respuesta Procesador	Accesibilidad de señales	TET	Elementos externos	Análisis
Péndulo Invertido	In A: 1 Out A: 2 ✓	✓	✓	✓	$3{,}76e^{-7}$	Generador de señal de entrada X	Teniendo en cuenta que el péndulo es un sistema no lineal, se aplica un control para mantenerlo en un punto de operación, por lo que la identificación es de un sistema modificado, dicha identificación no describe el comportamiento real del sistema; adicionalmente, no se cuenta con un dispositivo capaz de generar la entrada adecuada para el sistema.
Smart Grid	In A:1 Out A: 15 X	✓	✓	X	$2{,}66e^{-5}$	Etapa de Potencia X	El Target PC corre el modelo de manera adecuada, pero la DAQ no cumple los requerimientos para representar todas las señales necesarias.
UEA	Out A: 2 ✓	✓	✓	✓	$5{,}93e^{-7}$	PLC ✓	El simulador cumple con todos los requerimientos para representar el modelo, a demas de contar con el elemento externo para interactuar con el simulador (PLC).

4. IMPLEMENTACIÓN DE LA UEA EN EL RTDS

Como se define en el capítulo anterior, el caso de estudio final es la UEA, su implementación se desarrolla en 3 etapas: diseño, modelo y validación, en este capítulo se describen las etapas de implementación. En la sección 4.1 se proporcionan los conceptos para un entendimiento claro de la implementación. Por otro lado, en la sección 4.2 se describe la arquitectura del RTDS, se especifican los protocolos, la dirección y el flujo de datos en el simulador. El modelo ejecutado por el simulador en tiempo real está descrito en la sección 4.3, la validación empleando el RMSE y análisis de funcionamiento se expone en la sección 4.4.

4.1. MARCO TEÓRICO

4.1.1. Conjunto de protocolos TCP/IP

TCP/IP nació a partir del trabajo que se llevó a cabo en la creación de la red DARPA durante 1970, el desarrollo se realizó experimentalmente con nuevos protocolos que proporcionaran confianza y permitiera la comunicación entre varias estaciones principales. En los inicios se usaba el protocolo *Network Control Program* (NCP) hasta 1983 cuando la conversión a los nuevos protocolos TCP/IP fue completada (Miller, 2009).

Este conjunto de protocolos, consiste en una interrelación de múltiples protocolos que ocupan la capa superior de las tres vistas en el modelo ARPA: capa de interfaz de red, capa de internet y capa host a host.

La capa host a host, es la responsable de la integración entre el host remitente y el host receptor independientemente del camino, la distancia y la trasmisión del mensaje, además de permitir la corrección de errores de comunicación (Deon Reynders, 2003). Esta capa cuenta con dos protocolos asociados:

- Protocolo de Datagramas de Usuario (UDP)

Es un protocolo sin conexión usado por la dirección de puerto de la capa superior, ofrece la mínima sobrecarga ya que permite la eliminación del chequeo de errores, error de conexión y corrección; transmitiendo la información referida como un datagrama UDP sin establecer inicialmente una conexión con el receptor. Para estar seguro de que la información se está transmitiendo al receptor indicado, se establece un periodo de tiempo de espera para recibir la confirmación del paquete y si no se verifica el recibido del paquete, éste puede ser reenviado o cancelado (Held, 2002).

- Protocolo de Control de Transmisión (TCP)

Es un protocolo de conexión orientado a ofrecer protección ampliamente mejorada y de control de errores. Requiere establecer una conexión entre el remitente y el receptor del

mensaje, así como un reconocimiento del estado de disponibilidad del receptor antes de enviar el mensaje (Carne, 2004).

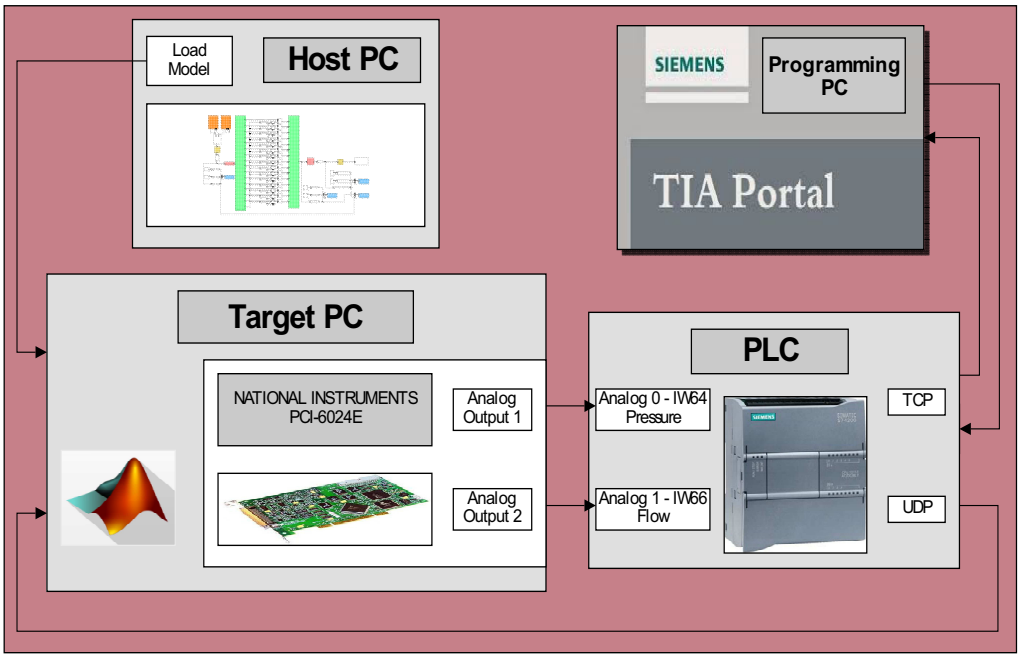
Teniendo en cuenta la descripción de los protocolos, se procede a diseñar la arquitectura del RTDS.

4.2. ARQUITECTURA

4.2.1. Descripción

La comunicación entre el Host PC y el Target PC se realiza por medio del protocolo TCP/IP. Como se visualiza en la Figura 27, el modelo parametrizado se carga en el Target PC desde el Host PC, el Target, por medio de la DAQ envía señales eléctricas (flujo-presión) al PLC, que las procesa y retorna la información empleando el protocolo UDP ya que no satura la red y permite enviar la señal de cada variable por un puerto específico, logrando una implementación de HIL. El Programming PC se comunica bidireccionalmente con el PLC, de manera opcional, desde el Programming PC se puede manipular la lógica cargada en el PLC.

Figura 27: Simulator Architecture



Fuente: Autores.

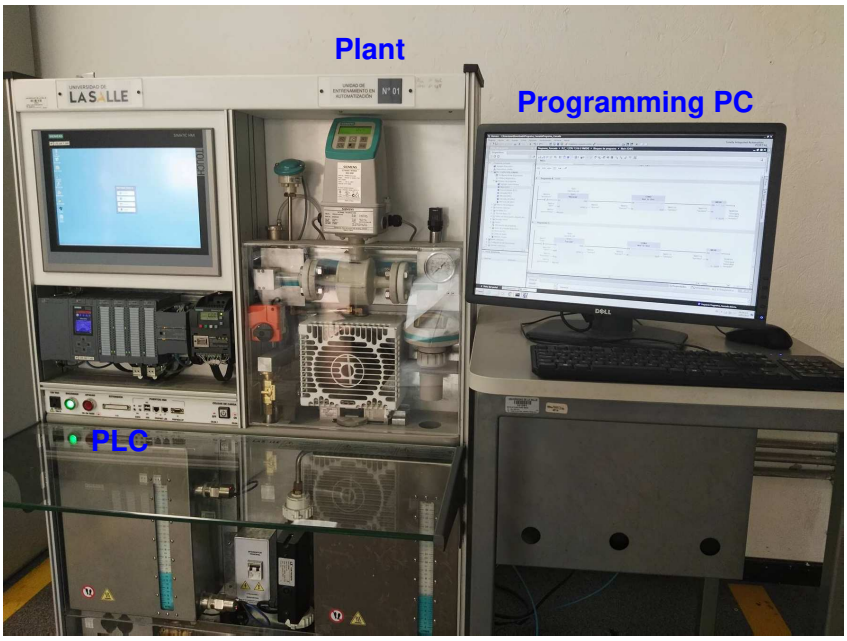
4.2.2. Simulador xPC vs UEA

El software de manejo y programación en la UEA es TIA Portal-Siemens, éste permite cargar la lógica de control en un PLC Siemens que a la vez recibe las señales de los sensores

de flujo-presión y envía la señal de consigna a la bomba.

En la Figura 28 se expone la estructura física de la planta real (UEA), compuesta por un panel *Human machine Interface* (HMI), un PLC Siemens; la Planta hace referencia a: la bomba, tubería y sensores de flujo-presión y por último, el programming PC desde el cual se programa el PLC y se adquieren datos de los sensores y el actuador por medio de MATLAB/Simulink®.

Figura 28: Physical Plant



Fuente: Autores.

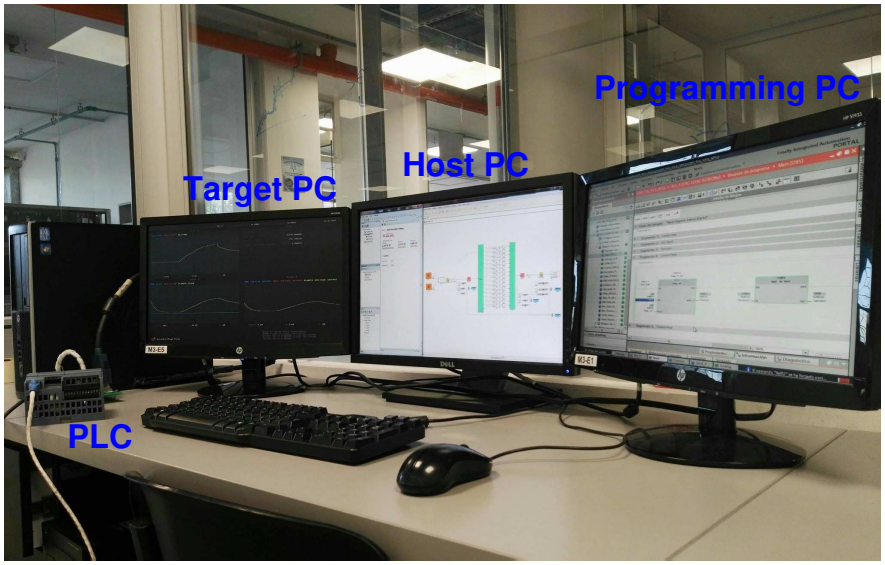
Basándose en los componentes de la UEA apreciados en la Tabla 1 y con el fin de lograr que el RTDS tenga un comportamiento equivalente a la planta real, los requerimientos que debe cumplir el simulador se explican en la Tabla 3.

Tabla 3: xPC software and hardware Requeriments.

Elemento	Requerimientos
Host PC	**Software:** MATLAB/Simulink®. **Tarjeta de Red.**
Target PC	**Puerto PCI** **Procesador Intel 3,1GHz** **RAM:** 4GB **Tarjeta de Red.**
DAQ	**NI serie E** **Maxima Rata de Muestreo:** 200KS/s **Voltaje de salida:** ±10 V **No. salidas análogas:** 2
PLC	**No. entradas análogas:** 2 **Tarjeta de Red.**
Programming PC	**Software:** TIA PORTAL. **Tarjeta de Red.**

Teniendo en cuenta los requerimientos descritos en la tabla anterior, la Figura 29 detalla la estructura física del simulador xPC, compuesta por un PLC Siemens, el Target PC con una DAQ, el cual representan la planta, el Host PC, donde se carga el modelo y el Programming PC donde se desarrolla la lógica del control diseñado para el PLC.

Figura 29: Physical xPC



Fuente: Autores.

Para el usuario del simulador se propone un entorno de programación (TIA Portal-Siemens) y un controlador (PLC Siemens) equivalente al de la planta real.

Por otro lado, para el Target PC, MATLAB/Simulink® ofrece un *toolbox* con *drivers* para diferentes DAQ que pueden ser usados para aplicaciones en tiempo real y una interfaz para la visualización de las señales. Cabe resaltar que el Programming PC puede ser el mismo Host PC, solamente si cuenta con el software TIA Portal-Siemens.

4.3. MODELOS PARA LAS VARIABLES FLUJO Y PRESIÓN

Con la arquitectura y las comunicaciones definidas, se desarrollan los modelos que han de ser embebidos en el Target PC. Experimentalmente se determina que para cubrir todos los puntos de operación es necesario dividir el rango de funcionamiento de la UEA en 15 identificaciones, iniciando en el 12 % aumentando proporcionalmente el 5.8 % hasta llegar al 94 % de la potencia del actuador, no se consideran los valores inferiores al 12 % porque el actuador no genera flujo en la tubería y la medida no es confiable. Para valores superiores al 94 % el fluido presenta un cambio de propiedades como espuma generando errores en la lectura del sensor. Siguiendo la recomendación de (MathWorks, 2014b) para la ejecución de un modelo en tiempo real basado en la tecnología xPC Target, se trabajan las identificaciones en tiempo discreto.

4.3.1. Discretización Funciones de Transferencia

El procedimiento empleado para discretizar las funciones de transferencia se presenta en el Algoritmo 3, utilizando la aproximación bilineal "tustin" ejecutada en MATLAB®, para este procedimiento se define un tiempo de muestreo de 0.001 s teniendo en cuenta que corresponde con el ciclo de operación del PLC.

Algorithm 3 Discretization procedure

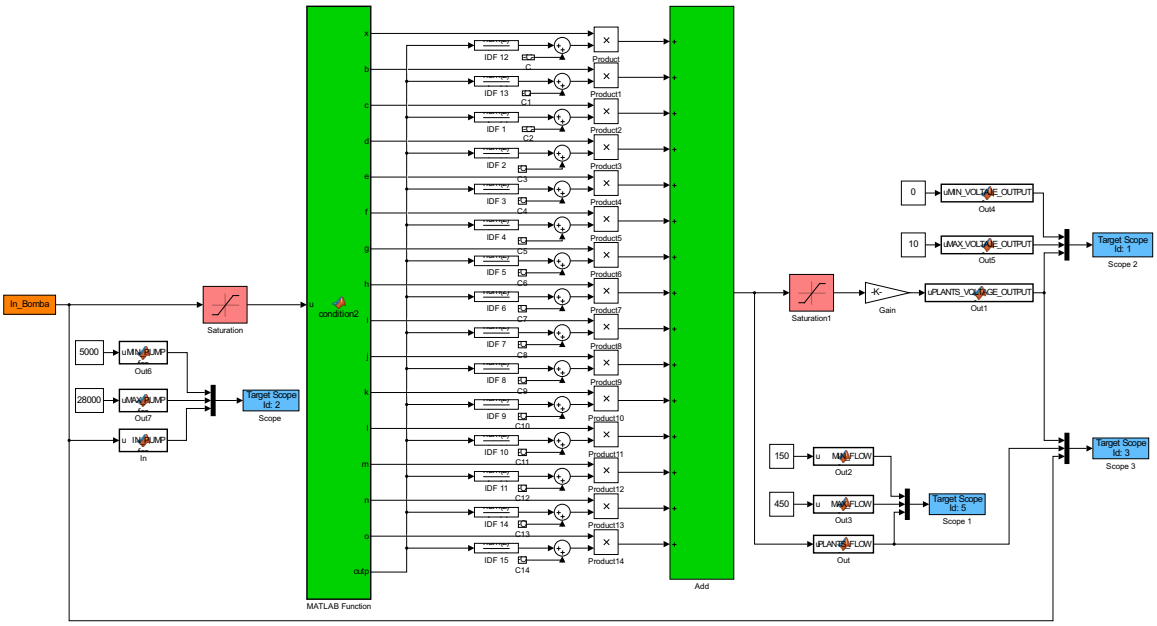
```
1: procedure In Contionuous transfer function(Y(s), U(s))
2:     Choose Sample Time.
3:     Choose discretization method.
4:     MATLAB® c2d function.
5:     return Discrete transfer function
```

Las funciones de transferencia para las cuatro configuraciones identificadas y discretizadas se muestran en la sección de "Anexos".

4.3.2. Implementación MATLAB/Simulink®

Con la discretización de las funciones de tranferencia realizada, se diseña el modelo de MATLAB/Simulink® para embeber en el Target PC, en la Figura 30 se visualiza el diseño del modelo para la variable de flujo operando con una válvula de llenado abierta.

Figura 30: Flow One Valve Model

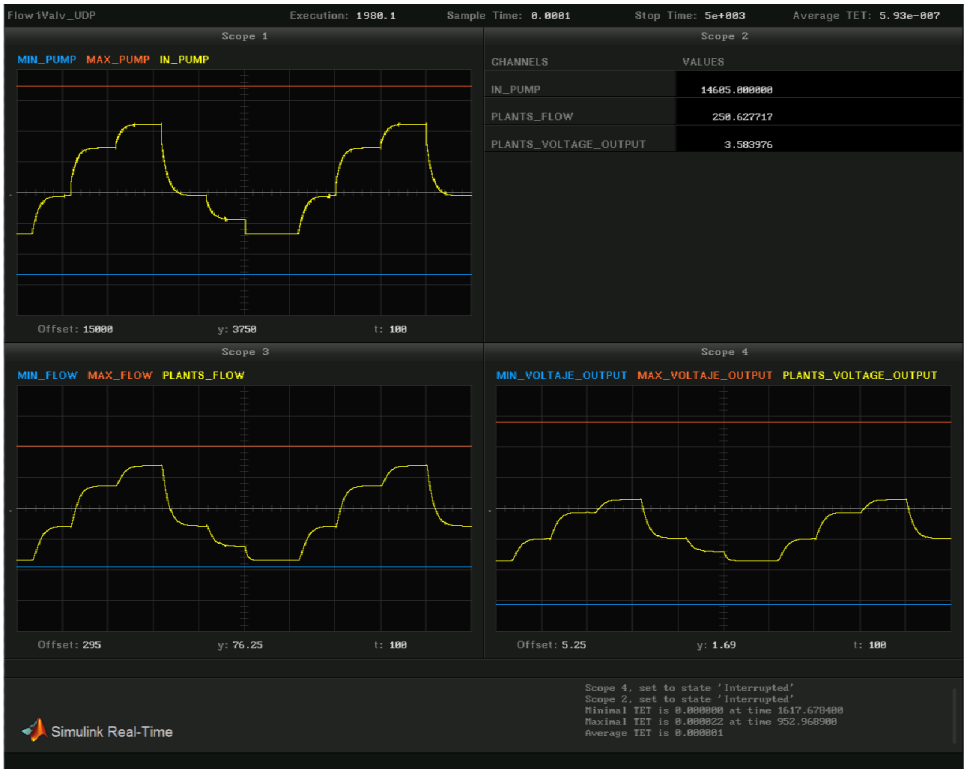


Fuente: Autores.

En el modelo para la configuración de flujo con una válvula de llenado, el bloque "MATLAB Function-condition2" es un condicional que funciona como selector, considera el valor de entrada y dependiendo de éste activa su respectiva salida. Los bloques "IDF" contienen las funciones de transferencia discretas, el número que los acompaña corresponde a su rango de operación. Cada punto de operación cuenta con un *offset* en su señal, por lo que se implementan los bloques de adición con las constantes, y los bloques de multiplicación eliminan la acción de las funciones de transferencia que no correspondan al punto de operación. El bloque "Add" permite el cambio entre funciones de transferencia sin afectar la señal de salida, es decir, en el cambio de rango de operación la salida solamente toma un valor.

Para la visualización de las variables se dispone de 4 pantallas, una pantalla numérica y tres gráficas con las variables de: consigna bomba (entrada), flujo o presión de salida y voltaje de salida, como se muestra en la Figura 31.

Figura 31: xPC Flow Screenshot



Fuente: Autores.

Como se menciona en la sección "Identificación de las variables Flujo y Presión" la UEA opera en 4 configuraciones diferentes:

- Flujo con 1 válvula de llenado.
- Flujo con 2 válvulas de llenado.
- Presión con 1 válvula de llenado.
- Presión con 2 válvulas de llenado.

Para cada configuración hay un modelo equivalente al mostrado en la Figura 30 con sus respectivas funciones de transferencia.

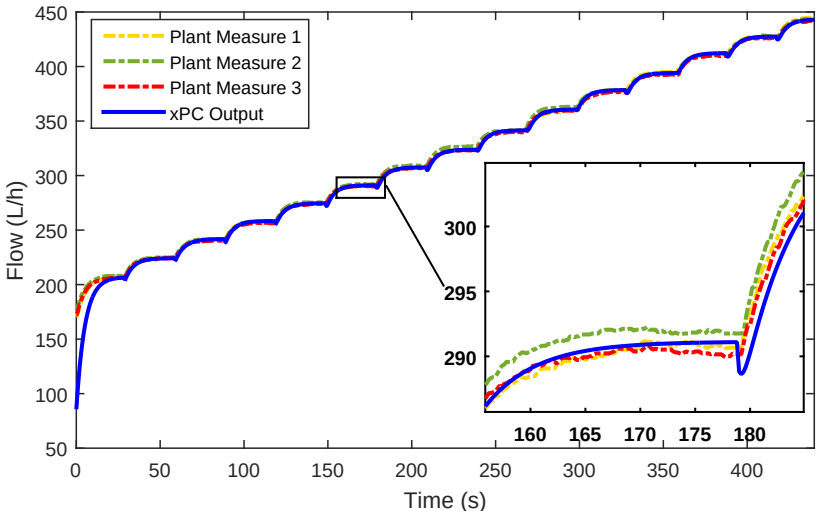
4.4. VALIDACIÓN DEL MODELO

Posterior al diseño de los modelos de MATLAB/Simulink® se procede a embeberlos en el Target PC, después de verificar su correcta ejecución se realiza la validación de los mismos estimulando el Target PC con la misma entrada que la UEA. Para evaluar todos los puntos de operación se propone como entrada una escalera con valores de 12 % al 94 % de la potencia del actuador.

4.4.1. Flujo con una válvula

En la Figura 32, se observa la comparación de la variable de flujo entre el simulador xPC y la UEA operando con una válvula de llenado abierta, ésta se estimula 3 veces con la misma entrada. Se puede evidenciar que la respuesta de la planta UEA tiene una variación de $\pm 5\ L/h$ ante la misma entrada, el simulador responde dentro del rango de variación con la que cuenta la UEA.

Figura 32: Flow Open Loop UEA - xPC Comparison with One Valve

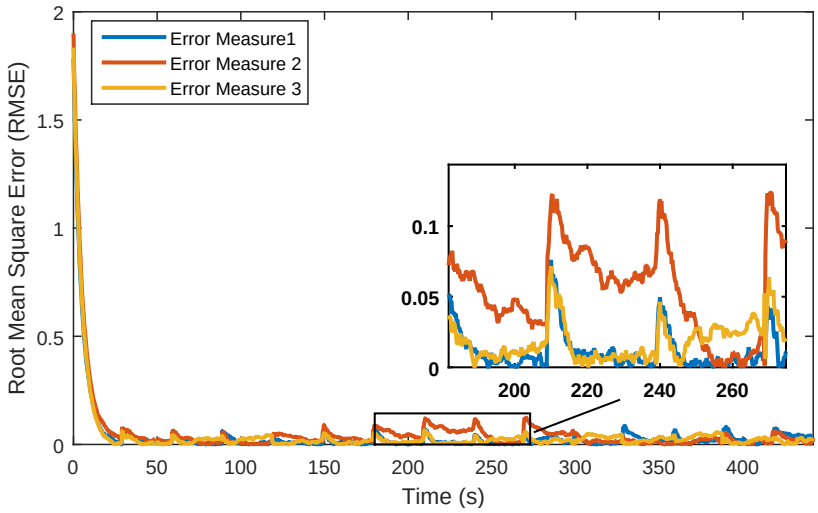


Fuente: Autores.

En el acercamiento de la Figura 32, se puede detectar que en el segundo 177 se presenta un *undershoot* de $\pm 3\ L/h$, esto se debe a que en el cambio de una función de transferencia a otra, el modelo de la Figura 30 vuelve 0 la señal en la que se encuentra operando y enseguida retoma la señal de la siguiente función de transferencia.

En la Figura 33 se aprecia el RMSE, se realiza entre la señal del simulador y las 3 muestras de la UEA. Como se menciona anteriormente, en el simulador xPC se presenta un error en los cambios de rango de operación, es por esto que se observan los picos visualizados en el acercamiento de la figura, también se observa que después del pico el error tiende a estabilizarse, es decir, el simulador representa el comportamiento de la planta en todos sus rangos de operación. El error presenta aumentos en los límites de operación de cada función de transferencia, debido a que el modelo requiere un tiempo para multiplexar la función de transferencia en la que se encuentra operando.

Figura 33: Flow Error Open Loop UEA - xPC Comparison with One Valve

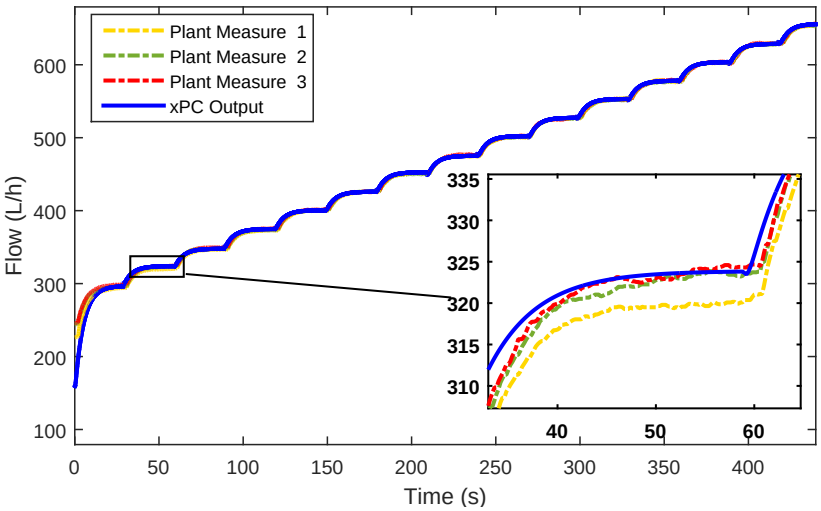


Fuente: Autores.

4.4.2. Flujo con dos válvulas

En la Figura 34 se detalla la comparación de la variable flujo entre el simulador xPC y la UEA operando con las dos válvulas de llenado abiertas, a diferencia de la configuración con una válvula de llenado abierta el límite de flujo máximo es ± 650 L/h (200 L/h más), dado que el fluido tiene dos caminos para recircular. Como se aprecia en el acercamiento, la muestra 1 de la UEA presenta un *offset* de ± 4 L/h con respecto a las muestras 3 y 4 en el segundo 56, y se mantiene hasta el cambio de *Set Point*.

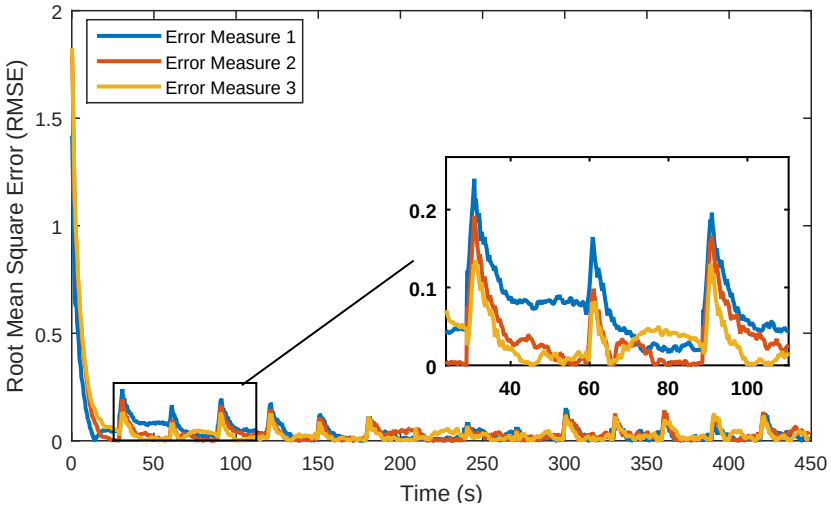
Figura 34: Flow Open Loop UEA - xPC Comparison with Two Valves



Fuente: Autores.

En la Figura 35 se visualiza el RMSE entre la señal del simulador y las 3 muestras de la UEA, se evidencia que después de los picos el error tiende a ser constante, por otro lado, el valor inicial de 1.8 L/h se debe a que la planta y el simulador no cuentan con las mismas condiciones iniciales. El *offset* presentado en la muestra 1 de la UEA del segundo 40 al 60 genera un error de 0.9 L/h.

Figura 35: Flow Error Open Loop UEA - xPC Comparison with Two Valves



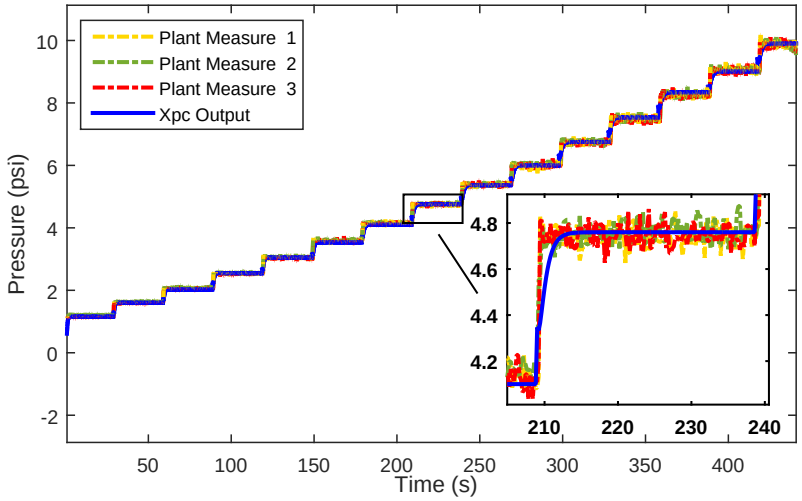
Fuente: Autores.

4.4.3. Presión con una válvula

En la Figura 36, se contempla la comparación de presión entre el simulador xPC y la UEA operando con una válvula de llenado abierta.

Las medidas de presión en la UEA tiene una oscilación de ± 0.3 *psi*, aunque esto no representa que sea un sistema de segundo orden, puesto que el comportamiento de las variables de procesos son de primer orden (Corriou, 2013). Se puede visualizar que la respuesta del simulador para la variable de presión se encuentra dentro del rango de oscilación de las medidas de la UEA.

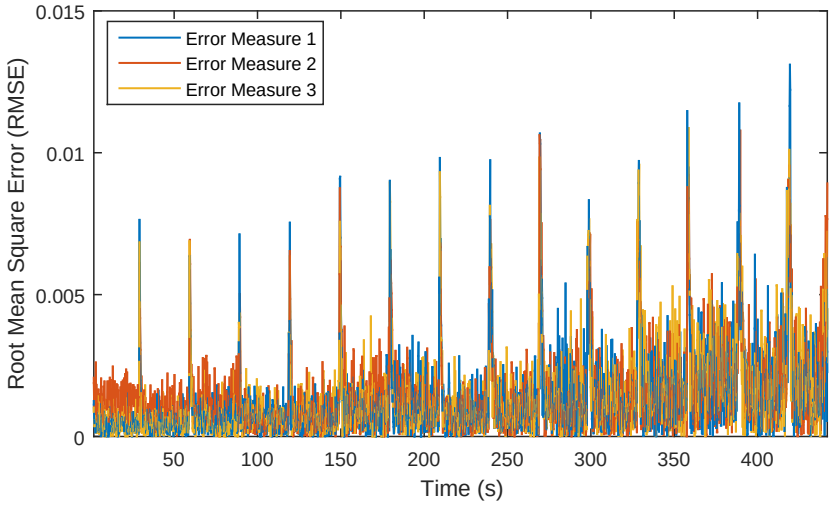
Figura 36: Pressure Open Loop UEA - xPC Comparison with One Valve



Fuente: Autores.

En la Figura 37 se visualiza el RMSE, a pesar de que el error incrementa proporcionalmente a la amplitud de las oscilaciones, conserva un patrón. Por otra parte, al igual que en la variable de flujo, el error presenta picos debido al cambio de un rango de operación a otro, también se observa que el valor del error para los cambios de rango se encuentran entre 0.0075 *psi* y 0.025 *psi*.

Figura 37: Pressure Error Open Loop UEA - xPC Comparison with One Valve

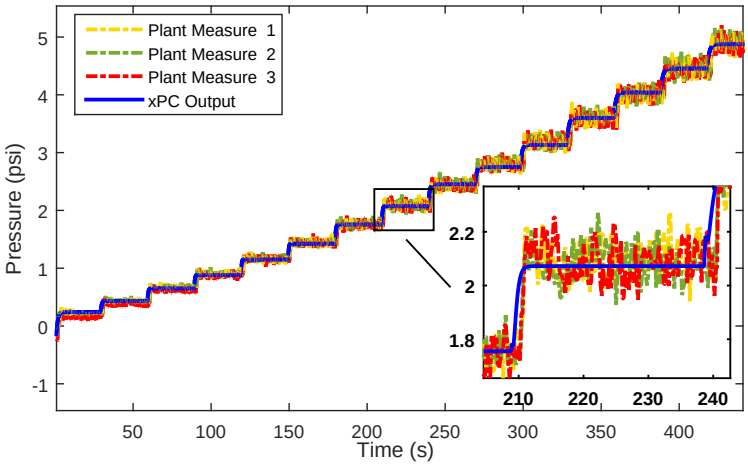


Fuente: Autores.

4.4.4. Presión con dos válvulas

En la Figura 38 se muestra la comparación de la variable de presión entre el simulador xPC y la UEA operando con las dos válvulas de llenado abiertas, ésta se estimula 3 veces con la misma entrada. Se evidencia menos presión que en la Figura 36, dado que con la configuración de dos válvulas de llenado abiertas el actuador ejerce menos fuerza sobre la tubería, generando que la presión máxima sea de 5 *psi*.

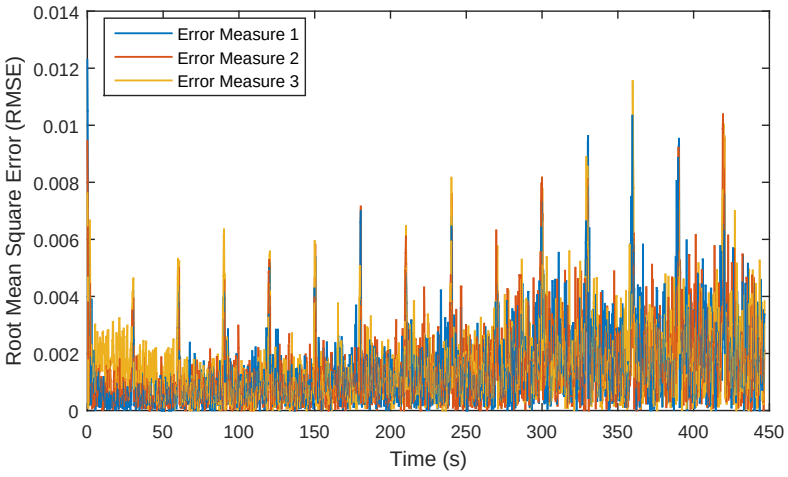
Figura 38: Pressure Open Loop UEA - xPC Comparison with Two Valves



Fuente: Autores.

Realizando una comparación entre la Figura 37 y la Figura 39, se evidencia que el RMSE se comporta de manera similar para las dos configuraciones, puesto que las oscilaciones tienen la misma amplitud máxima (0.013 *psi)*.

Figura 39: Pressure Error Open Loop UEA - xPC Comparison with Two Valves



Fuente: Autores.

Se analiza que el simulador xPC para las 4 configuraciones presenta picos en el RMSE, generado por el cambio de un rango de operación a otro, además de tener un comportamiento estable. La respuesta del simulador xPC para todos los modelos de MATLAB/Simulink® está dentro del rango de respuesta de la UEA.

5. ANÁLISIS DE LA RESPUESTA DEL CASO DE ESTUDIO CON UN CONTROLADOR

Para obtener un resultado comparativo entre la UEA y el RTDS bajo las mismas condiciones típicas de evaluación, se implementa la misma estrategia de control en el RTDS y en la UEA y se analiza el resultado del simulador en sus 4 configuraciones de operación.

La sección 5.1 introduce los conceptos necesarios para el diseño de un control y su implementación en controladores discretos. En la sección 5.2 se define el modelo final de MATLAB/Simulink® que se toma como plantilla para embeber en el Target PC, complementado con las comunicaciones del controlador. La sección 5.3 explica el diseño de dos controles PI desarrollados por la técnica de LGR y los resultados de las simulaciones al aplicar dichos controles a los sistemas. En la sección 5.4 se detalla la implementación de los controles diseñados en los PLCs y en la sección 5.5 se describe la configuración para la adquisición de los datos de la UEA y el RTDS simultáneamente mediante OPC. Los resultados de los sistemas con los controles implementados son presentados en la sección 5.6 con su respectivo análisis.

5.1. MARCO TEÓRICO

5.1.1. Sistema de control

El control automático ha desempeñado un papel vital en el avance de la ingeniería y la ciencia. Es esencial en operaciones industriales como el control de presión, temperatura, humedad, viscosidad y flujo (Ogata, 2003).

Un sistema de control se caracteriza por la presencia de una serie de elementos que permiten influir en el funcionamiento del mismo. La finalidad de un sistema de control es conseguir, mediante la manipulación de las variables de control, un dominio sobre las variables de salida, de modo que estas alcancen unos valores prefijados (Kuo, 1996).

El diseño de sistemas de control lineales se puede realizar en el dominio del tiempo o en el dominio de la frecuencia. La precisión (error en estado estable) se obtiene al aplicar una entrada tipo escalón, rampa o parabólica y el diseño para cumplir requisitos como: sobre impulso máximo, tiempo de subida o tiempo de estabilización, es más conveniente realizarlo en el dominio del tiempo (de sistemas y automática, 1996).

El diseño de sistemas de control continuo puede ser mediante: (Ogata, 2003)

- Realimentación de estados
- LQR
- Diseño polinomial

- Reguladores PID
    - Respuesta en frecuencia
    - Lugar geométrico de las raíces (LGR)

El método del **LGR**, permite ubicar en un gráfico los polos de un sistema en lazo cerrado a partir del conocimiento de los polos en lazo abierto, en función de un parámetro variable (Ogata, 2003). Se utiliza la condición de ángulo y módulo para hallar las constantes del control.

Condición de Ángulo: $faseGLA(s) = \pm 180°(2\lambda + 1); \lambda \in N$

Condición de Módulo: $|GLA(s)| = 1$

Se define el LGR como el conjunto de puntos del plano S en los que se verifica la condición del ángulo, es decir, un punto que pertenece al LGR es un posible polo del sistema en lazo cerrado; para ello únicamente es necesario validar la condición de módulo, y ésta se cumplirá para un valor determinado de la ganancia del sistema en lazo abierto. Sin embargo, un punto del plano S que no pertenezca al LGR no puede ser polo en lazo cerrado porque no verifica la condición de ángulo, aunque varíe la ganancia del sistema en lazo abierto (Spartacus Gomáriz, 2004). De este modo, variando el parámetro k, donde:

$$GLA(s) = k\frac{(s+z1)...(s+zm)}{(s+p1)....(s+pn)}; 0 \leq K \geq \infty \tag{5.1}$$

Se logra trazar el LGR que proporciona los valores de los polos en lazo cerrado en función de k.

El **control proporcional integral derivativo (PID)** es la estrategia de control más usada en las aplicaciones industriales; se estima que más del 90 % de los lazos de control utilizan control PID, dado que es una estrategia simple, efectiva y no requiere una gran fundamentación teórica para su utilización en los procesos cotidianos (Ang, 2005).

Una de las estructuras estándar de controladores PID es la estructura paralela. Esta estructura se refiere a la función de transferencia de un controlador PID, la cual se puede expresar en su forma ideal como: (María Elena Anaya Pérez, 2014)

$$G_{PID} = \frac{U(s)}{E(s)} = K_p(1 + \frac{1}{T_i S} + T_d S) \tag{5.2}$$

Donde $U(s)$ es la señal de control actuando en la señal de error $E(S)$, K_p es la ganancia proporcional, T_i es la constante de tiempo integral, T_d es la constante de tiempo derivativo, y S es el argumento de la transformada de Laplace.

5.1.2. Discretización del control

Con frecuencia, los modelos de sistemas, filtros o controladores, están dados como un modelo de tiempo continuo en forma de ecuación diferencial o función de transferencia en Laplace (Bemporad, 2011). Con el fin de obtener la ecuación en diferencia indicada para implementar en un computador, se debe discretizar el modelo en tiempo continuo (Haugen, 2010). Esta aproximación puede ser de diferentes métodos:

Métodos basados en la derivada de la ecuación diferencial

Son métodos implícitos por que usan el valor del siguiente paso para evaluar el campo vectorial. Estos métodos son mucho más estables que los métodos explícitos (Luo, 2015).

- Método de Euler hacia adelante: es comúnmente usado para el desarrollo de simuladores simples. La aproximación del área se realiza con rectángulos.

- Método de Euler hacia atrás: es usado para la discretización de filtros simples y controladores industriales (Haugen, 2010).

5.1.3. Programación de PLCs

Un PLC es un computador industrial que permite el control y monitoreo de variables en aplicaciones industriales (Lewis, 1998).

La introducción de los PLCs a la industria se hace como un medio simple para el control de entradas y salidas, luego evolucionan a sistemas basados en procesadores que ejecutan algoritmos complejos de control e incluyen funcionalidades como HMI e integración con sistemas de información. Los diferentes tipos de PLCs tienen un principio de funcionamiento discreto y cubren amplias tareas, desde una red de computadores pequeña, hasta PLCs modulares de alto rendimiento, dependiendo de sus características como: velocidad de procesamiento, capacidad de trabajo en red y módulos *In/Out* soportados (John, 2001).

Durante los últimos años, una amplia gama de técnicas de programación han sido usadas para escribir programas enfocados en las aplicaciones de control industrial, dando lugar al desarrollo de diferentes lenguajes de programación para PLCs: Lista de Instrucciones (IL), Texto estructurado (ST), Diagrama Ladder (LD), Diagrama de bloques (FBD) y Diagrama Funcional de Secuencia (SFC). El estándar IEC 61131-3 especifica el correcto uso de cada uno y de la estructura la lógica que ejecutará el PLC.

- **Diagrama Ladder:** también llamado lenguaje de contactos, ha ido ganando aceptabilidad por su facilidad de comprensión, ya que es un lenguaje de programación gráfico basado en esquemas eléctricos de control básico (Ponsa Asensio, 2006).

5.1.4. Open Platform Communications (OPC)

El análisis y transmisión de los datos de un PLC a un computador puede realizarse por diferentes medios, se han desarrollado estándares con el fin de evitar un protocolo para cada fabricante, uno de los estándar más comunes es OPC.

OPC permite el intercambio seguro y fiable de datos en el espacio de automatización industrial entre otras industrias ("OPC Foundation", 2017). Emplea un cliente-servidor enfocado al intercambio de información, encapsula la fuente de información de proceso como un dispositivo y la coloca a disposición por medio de su interfaz. Un cliente OPC se conecta en el servidor OPC para acceder a los datos (Wolfgang Mahnke, 2009).

El *toolbox* de MATLAB/Simulink®, permite acceder y leer datos originales y procesados de cualquier historiador que cumpla con el estándar de acceso a información de OPC (MathWorks, 2014a). También puede acceder a datos activos de un servidor OPC de tres maneras:

- Ejecutar las funciones del *toolbox* OPC desde la línea de comandos de MATLAB®.
- Utilizar la aplicación *Client OPC* para conectarse a los servidores.
- Utilizar los bloques de MATLAB/Simulink® para leer y escribir datos desde y hacia un servidor OPC mientras simula un sistema.

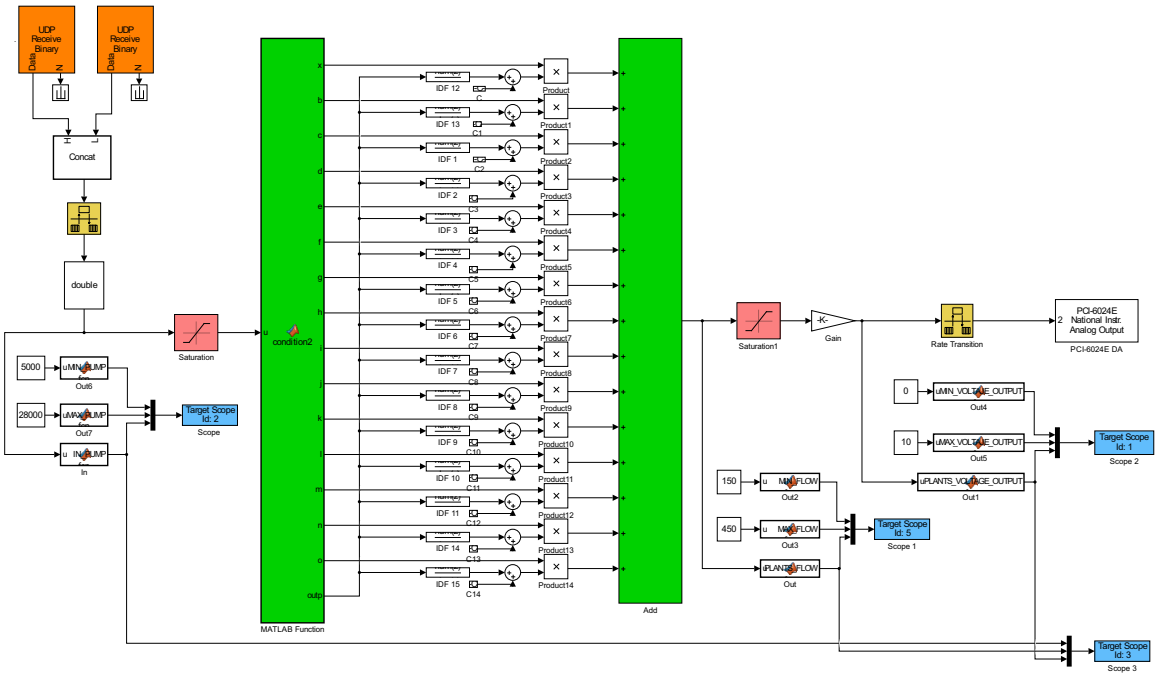
5.2. MODELO FINAL A EMBEBER

Se complementa el modelo de MATLAB/Simulink® de la Figura 30 con las comunicaciones y el acondicionamiento de las señales obteniendo como modelo final el mostrado en la Figura 40.

Los bloques "UDP Receive" realizan la comunicación con el PLC bajo el protocolo UDP, el dato recibido es la consigna de entrada al sistema, en la UEA es la bomba. Para permitir una mayor velocidad en la transmisión de datos se propone dividir el dato en *bytes*, cada bloque recibe un *byte* de información y posteriormente se concatenan los *bytes* con el bloque "Concat". Con el fin de igualar las ratas de entrada y salida se utiliza el bloque "Rate Transition", se realiza una conversión de datos a *double* y se limita la señal por medio de un saturador con valores entre 10000 (Consigna mínima 10 %) y 26000 (Consigna máxima 94 %), terminando así el acondicionamiento para la señal de entrada.

Con el fin de simular los límites de la variable de flujo en la UEA, se utiliza un saturador con límites entre 0 L/h y 700 L/h (flujo máximo) y una ganancia de 0.0143 escaliza la señal de 0 V a 10 V ya que este es el rango de entrada al PLC, ésta señal escalizada es enviada por un canal de la DAQ configurado en el bloque "PCI-6024E DA". Para cada variable se asigna un canal.

Figura 40: Flow One Valve Model with Communication



Fuente: Autores.

5.3. DISEÑO DEL CONTROL

Después de obtener el modelo final para las variables de flujo y presión se procede a diseñar los controles.

Se propone un control para la variable flujo y otro control para la variable presión, la estrategia de diseño seleccionada es el LGR.

- **Flujo**

Para el diseño del control por LGR inicialmente se define el tiempo de establecimiento Ts y el porcentaje de *overshoot* Mp:

$$Ts = 67 \qquad Mp = 0{,}001 \tag{5.3}$$

Teniendo las características deseadas del sistema se determinan los polos de diseño(Ogata y Yang, 2010):

$$z = sqrt(log(0{,}001)^2/(log(0{,}001)^2 + \pi^2)) \tag{5.4}$$

$$w = \pi/(67 * sqrt(1 - 0{,}9103^2)) \tag{5.5}$$

$$G = 0{,}1133^2/(S^2 + 0{,}2062 * S + 0{,}1133^2) \tag{5.6}$$

El Algoritmo 4 describe el desarrollo de la técnica de LGR, siendo las entradas: polos de diseño (pd) y polos y ceros del sistema (pa, pb y c1). El algoritmo retorna un controlador PI.

$$pa = 3{,}694 \tag{5.7}$$

$$pb = 0{,}199 \tag{5.8}$$

$$c1 = pa \tag{5.9}$$

$$ks = 0{,}0045 \tag{5.10}$$

Algorithm 4 LGR

1: **In:pd, pa, pb, c1,ks**
2: *pd = polos(1,1)*
3: *i = imag(pd)*
4: *r = real(pd)*-1*
5: **if** $pa > r$ **then**
Return;
end;
6: **if** $r < pb$ **then**
$Angle = tand((-atand(i/(pa-r)) - atand(i/(pb-r)) + (180 - atand(i/(r+c1))) - (180 - atand(i/(r))) - 180) * -1)$
7: **else**
8: **if** $r > pb$ **then**
$Angle = tand((-atand(i/(pa-r)) - (180 - atand(i/(r-pb))) + (180 - atand(i/(r+c1))) - (180 - atand(i/(r))) - 180) * -1)$
end;
end;
9: *alph = (i+Angle*r)/Angle*
10: *pl1=sqrt(i^2+(r-pa)^2)*
11: *pl2=sqrt(i^2+(r-pb)^2)*
12: *pl3=sqrt(i^2+r^2)*
13: *cl1 = sqrt(i^2+(r+c1)^2)*
14: *cl1 = sqrt(i^2+(r-alph)^2)*
15: *k = (pl1*pl2*pl3)/(cl1*cl2*ks)*
16: *ci = (k*(s+alph))/s*
Return: ci

El controlador PI final para la variable de flujo es:

$$ci = \frac{2{,}889 * S + 2{,}835}{S} \tag{5.11}$$

Se define que éste control PI será implementado para la variable de flujo, siendo:

$$kp = 2{,}88 \qquad ki = 2{,}94 \tag{5.12}$$

- **Presión**

Se define el tiempo de establecimiento Ts y el porcentaje de *overshoot* Mp:

$$Ts = 14 \qquad Mp = 0{,}001 \tag{5.13}$$

Teniendo las características deseadas del sistema se determinan los polos de diseño(Ogata y Yang, 2010):

$$z = sqrt(log(0{,}001)^2/(log(0{,}001)^2 + \pi^2)) \tag{5.14}$$

$$w = \pi/(14 * sqrt(1 - 0{,}9103^2)) \tag{5.15}$$

$$G = 0{,}5490^2/(S^2 + 0{,}9868 * S + 0{,}5490^2) \tag{5.16}$$

Los parámetros de entrada al Algoritmo 4 son:

$$pa = 1{,}309 \tag{5.17}$$

$$pb = 0{,}9232 \tag{5.18}$$

$$c1 = pa \tag{5.19}$$

$$ks = 0{,}00045 \tag{5.20}$$

El controlador de salida para la variable de presión es:

$$ci = \frac{515{,}6 * S + 495{,}9}{S} \tag{5.21}$$

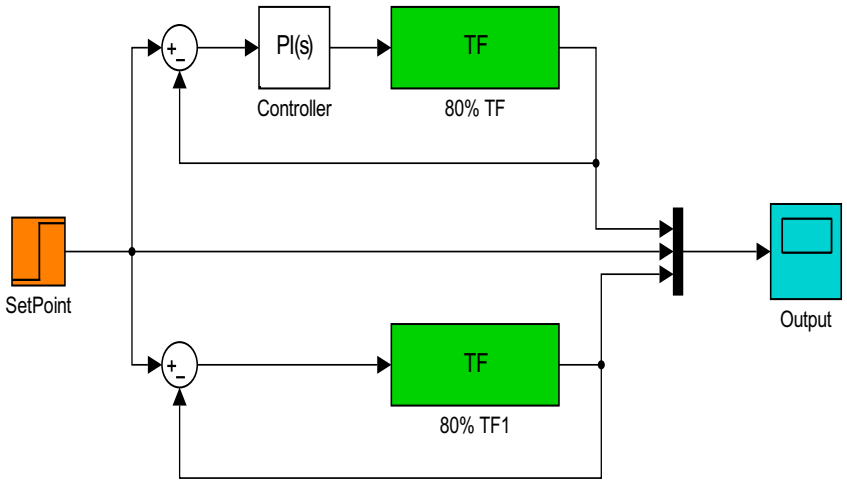
Se define que éste control PI será implementado para la variable de presión, siendo:

$$kp = 515{,}6 \qquad ki = 536{,}0782 \tag{5.22}$$

5.3.1. Simulaciones del control de las variables Flujo y Presión

Los controles diseñados para la variable flujo (Ecuación 5.11) y para la variable de presión (Ecuación 5.21) son implementados en el modelo de simulación de la Figura 41. Con el fin de evaluar la efectividad de aplicar un control a las variables de flujo y presión, se realiza una comparación de la respuesta de los sistemas en lazo cerrado sin control y con control, los dos lazos son estimulados con el mismo *Set Point*.

Figura 41: PI Simulation Model

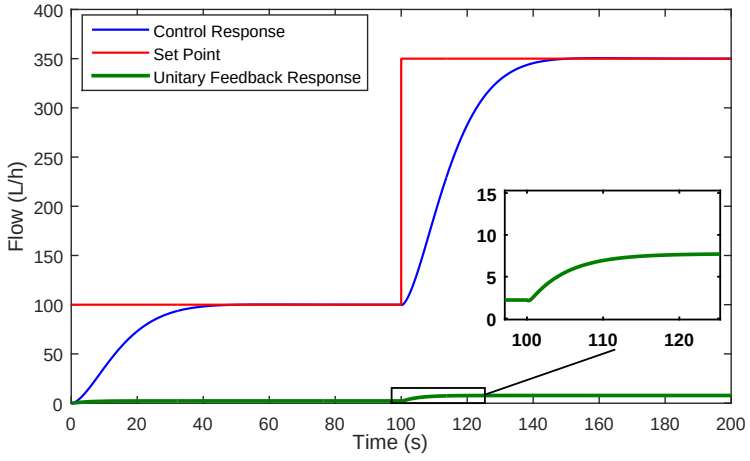


Fuente: Autores.

- **Flujo**

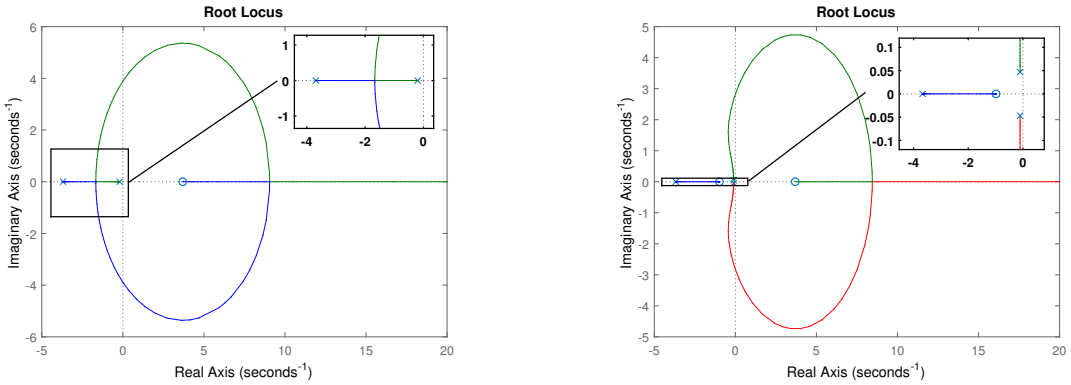
En la Figura 42, se observa como la respuesta del sistema con realimentación unitaria presenta un error de estado estable del 97.7 %, mientras que la respuesta obtenida con el control diseñado para la variable de flujo no presenta *overshoot* ni error de estado estable, lo cual indica que cumple con los requerimientos para ser implementado en la UEA y en el simulador.

Figura 42: Flow Control Response



Fuente: Autores.

Figura 43: Root Locus Flow Comparison



(a) Root Locus without Control **(b)** Root Locus with Control

Fuente: Autores.

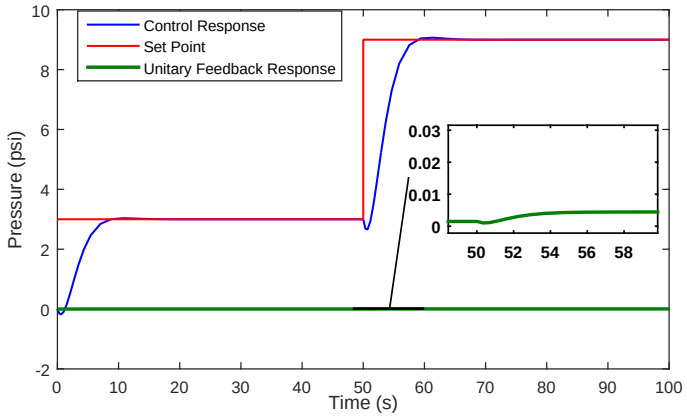
Las Figuras 43a y 43b representan el plano de LGR de la planta sin control y con control. Se evidencia el desplazamiento del sistema a la región de operación deseada ya que la ubicación de los polos dominantes coincide con los polos de diseño propuestos, Ecuación 5.23.

$$pd = -0{,}1031 \pm 0{,}0470i \tag{5.23}$$

- **Presión**

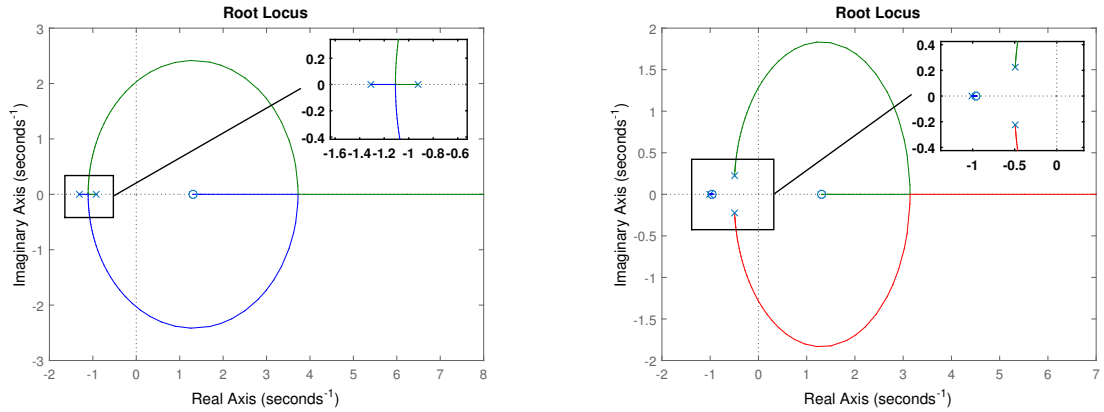
El error de estado estacionario para la variable de presión en realimentación unitaria es del 99.9 %, la Figura 44 demuestra que el control diseñado satisface los requerimientos para ser implementado: una respuesta sobreamortiguada y error de estado estable igual a 0.

Figura 44: Pressure Control Response



Fuente: Autores.

Figura 45: Root Locus Pressure Comparison



(a) Root Locus without Control

(b) Root Locus with Control

Fuente: Autores.

Las Figuras 45a y 45b representan el plano de LGR de la planta sin control y con control. Se visualiza el desplazamiento del sistema a la región de operación deseada ya que la ubicación de los polos dominantes coincide con los polos de diseño propuestos, Ecuación 5.24.

$$pd = -0{,}4934 \pm 0{,}2407i \tag{5.24}$$

5.4. PROGRAMACIÓN DEL PLC

Tanto el RTDS como la UEA operan con un PLC para el manejo de la lógica de programación, envío de información, acción de control y lectura de las señales. El desarrollo de la programación para el RTDS y la UEA incluye los controles diseñados en la sección anterior para la variable de flujo y presión. La UEA cuenta con señales de entrada de 4-20 mA proporcionadas por los sensores de flujo y presión, por otro lado, el RTDS recibe señales de 0-10 Vdc proporcionadas por la DAQ del Target PC.

5.4.1. Comunicación con el Target PC

En el funcionamiento del RTDS, el Target PC ejecuta en tiempo real el modelo de MATLAB/Simulink® que representa el comportamiento de las variables flujo y presión, utilizando como dispositivo de control el PLC. Por disposición de entradas y salidas en el PLC se propone la siguiente configuración:

Target PC → PLC: Dos señales eléctricas de 0-10 Vdc.

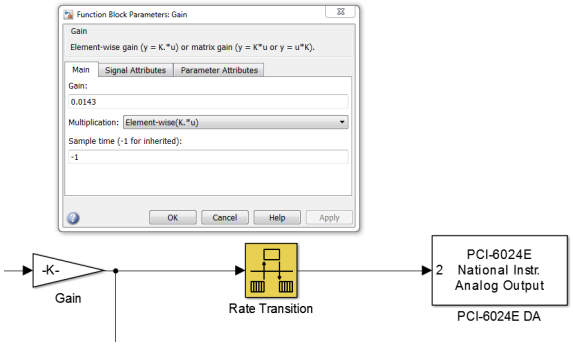
PLC → Target PC: Una señal, protocolo UDP.

Esta configuración se observa en la Figura 27 la cual describe la arquitectura del simulador.

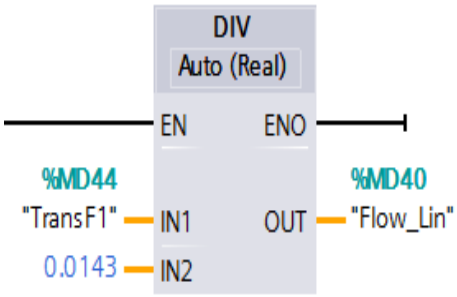
Acondicionamiento señales Target PC → PLC

Las señales eléctricas que recibe el PLC requieren una escalización a 0-10 Vdc. Como se menciona anteriormente, son dos canales de salida desde el Target PC con rangos de 0-700 L/h y 0-12 psi para las variables de flujo y presión respectivamente. En el modelo de MATLAB/Simulink® se operan las salidas con ganancias (bloque "Gain") para escalar las señales al rango deseado, y su posterior envío por el canal de la DAQ (bloque "PCI-6024E DA"), como se detalla en la Figura 46a.

Figura 46: Scale Flow Signal



(a) Scale Simulink Flow Signal OutPut



(b) Scale TIA Flow Signal InPut

Fuente: Autores.

La escalización de la señal en el PLC se realiza con la inversa de la ganancia definida en el modelo de MATLAB/Simulink®. En la Figura 46b se observa la ganancia inversa programada en el PLC.

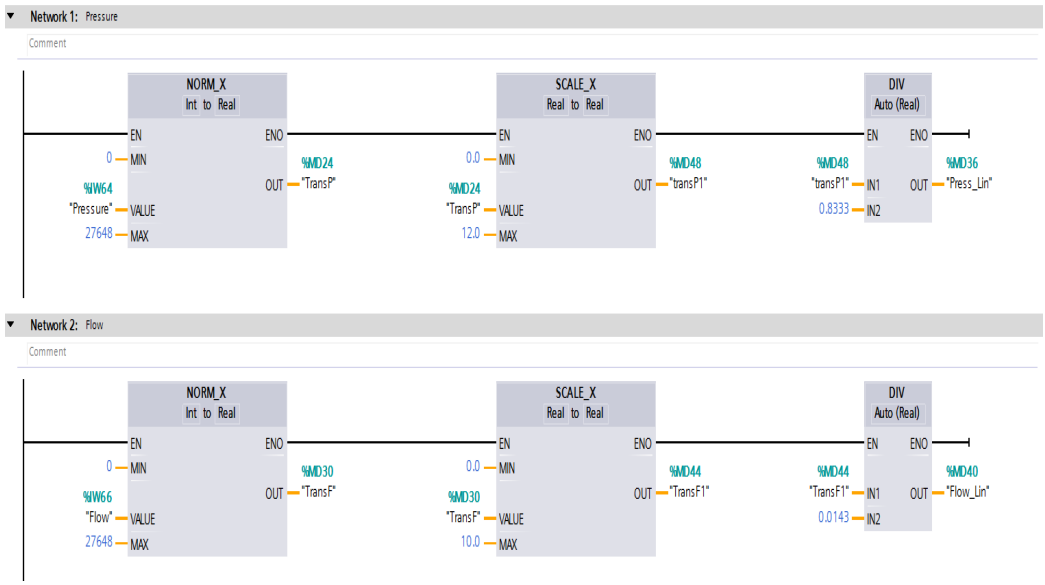
Las ganancias implementadas son:

$$Ganancia \quad Flujo = \frac{10}{700} \qquad Ganancia \quad Flujo = 0{,}0143 \tag{5.25}$$

$$Ganancia \quad Presión = \frac{10}{12} \qquad Ganancia \quad Presión = 0{,}833 \tag{5.26}$$

La programación de la ganancia en el PLC requiere una lectura previa de la señal eléctrica, se toma la guía propuesta por el fabricante, compuesta por un bloque de normalización, un bloque de escalización de la lectura desde 0-10 Vdc y la ganancia correspondiente a la variable, Ecuaciones 5.25 y 5.26. En la Figura 47 se observa la implementación final para la escalización de las variables flujo y presión en el PLC.

Figura 47: PLC Inputs Programming



Fuente: Autores.

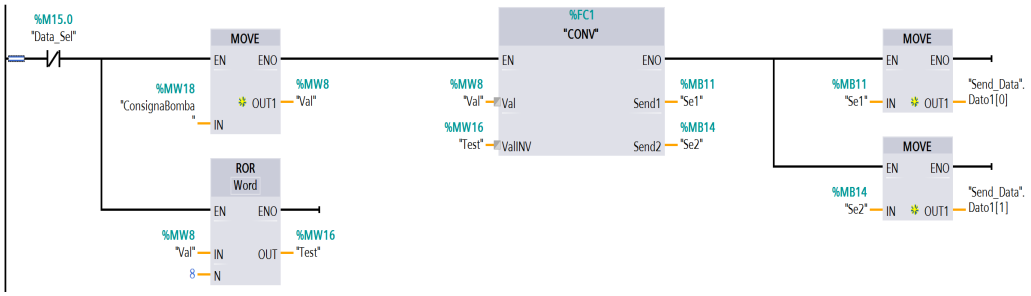
Comunicación Señal de Salida PLC → Target PC por UDP

El modelo de MATLAB/Simulink® que describe el comportamiento de las variables flujo y presión de la UEA mostrado en la Figura 30, recibe como entrada la consigna equivalente a la bomba con valores entre 0-27000, este rango de valores son medidas enteras unipolares empleadas en sensores y actuadores (SIEMENS, 2016); en este caso, este rango equivale a frecuencias entre 0 Hz y 60 Hz. Dicha consigna es enviada desde el PLC al Target PC.

- División de la consigna en Bytes

El envío del dato se realiza mediante el protocolo UDP, considerando una posible saturación de la red se divide el dato de la consigna de la bomba en *bytes*, se proporciona un valor a la consigna en tipo de dato "Palabra", seguido a esto se rotan los *bytes*, la nueva palabra se guarda en una variable diferente, se realiza una conversión de las dos palabras a *bytes*, ésta conversión se realiza en el bloque "CONV" visualizado en la Figura 48, mediante la función "DINT_TO_ BYTE" que guarda en una variable *byte* el valor de los primeros 4 *bits* de la palabra, así, el bloque "CONV" retorna dos *bytes*, uno con la parte superior de la palabra y otro con el restante de la misma.

Figura 48: Word Division



Fuente: Autores.

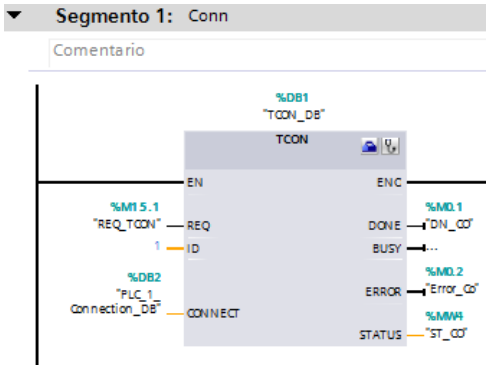
- Configuración de la conexión

Los requerimientos para enviar el dato de la consigna por UDP se dividen en: establecer la conexión, parametrizar el puerto y la dirección de envío y hacer una petición de envío.

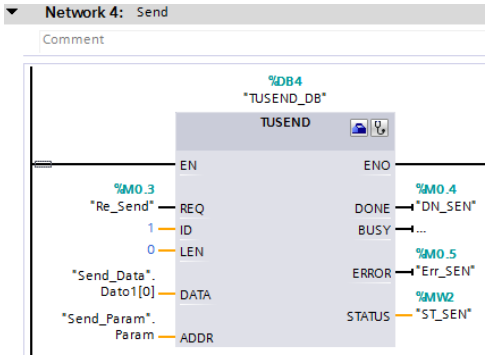
El bloque de la Figura 49a establece la conexión, la petición se realiza cuando el parámetro booleano "REQ" toma el valor "True".

El bloque de la Figura 49b envía el dato que se encuentre en el parámetro "DATA", en "ADDR" se ingresa la dirección del dispositivo a comunicar y el puerto que se usa para el envío, este bloque realiza el envío del *byte* cada vez que el parámetro booleano "REQ" tiene un flanco de subida. Los flancos de subida en el parámetro "REQ" definen la rata de transmisión, para el simulador se establece una rata de envío de 0.2 s.

Figura 49: UDP Configuration Blocks



(a) Connection Block



(b) Send Data Block

Fuente: Autores.

Para el envío de cada *byte* es necesaria la implementación de estos dos bloques, es decir, para el envío de la parte alta de la palabra requiere un "TCON" y un "TUSEND", y para la parte baja otro "TCON" y "TUSEND".

5.4.2. Implementación del control

La implementación de un controlador en un computador debe realizarse de forma discreta. Para la programación del control PI, se utiliza las siguientes formulas (Haugen, 2010):

$$e = ref - S \tag{5.27}$$

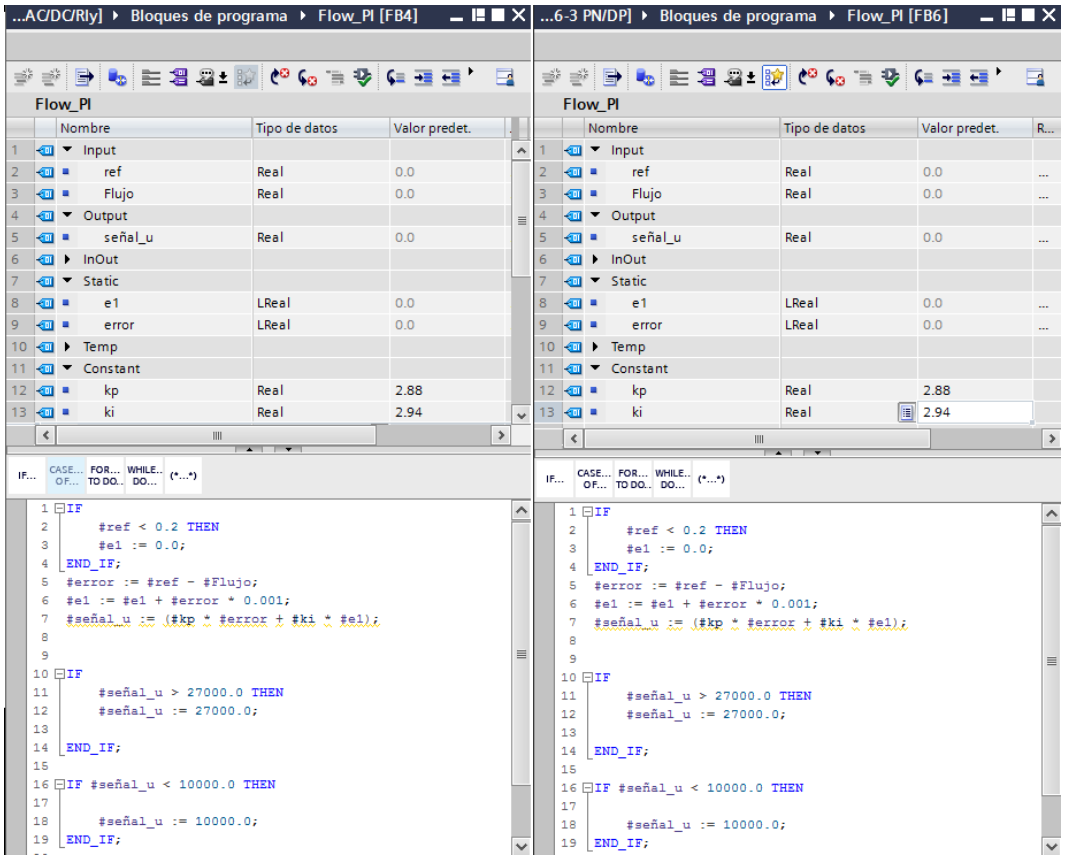
$$u = K_p * e + K_i * e_a \tag{5.28}$$

donde:

e=*Error actual*
e_a=*Error anterior*
K_i=*Constante integral*
u=*Señal de salida-Bomba*
K_p=*Constante proporcional*

Teniendo en cuenta que en las formulas existen variables recursivas, como se detalla en la Figura 50, se utiliza el lenguaje de programación ST, debido a que permite realizar este tipo de operaciones de forma más sencilla.

Figura 50: Control Implementation



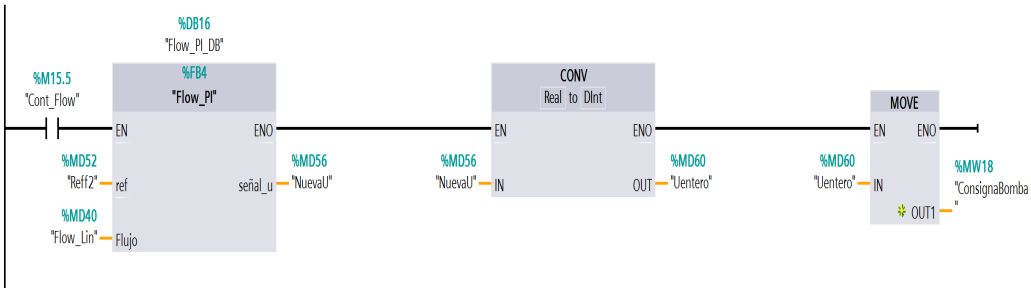
Fuente: Autores.

Las entradas al controlador PI son “Ref” y “Flujo”, donde “Ref” es la referencia del valor de flujo deseado asignada por el usuario y la variable “Flujo”, es el valor actual del caudal. La salida del control es la consigna de la bomba, la cual debe trabajar en un rango de operación de 10000 a 27000, por este motivo se implementan dos saturadores encargados de evitar daños en la bomba por sobrecarga.

Las variables Kp y Ki corresponden a los valores determinados en el diseño del controlador en la sección “DISEÑO DEL CONTROL”.

En la Figura 51, se expone la estructura en lenguaje de programación Ladder para la activación del control. La programación del control se realiza en el bloque “Flow-PI”, donde la señal “Nueva-U” corresponde a la consigna de la bomba, la cual pasa por un bloque de conversión “CONV” para cambiar el tipo de dato de Real a Entero, por último el valor es guardado en la variable “ConsignaBomba” que está conectada al puerto de salida asignado a cada variable del PLC.

Figura 51: Control Activation



Fuente: Autores.

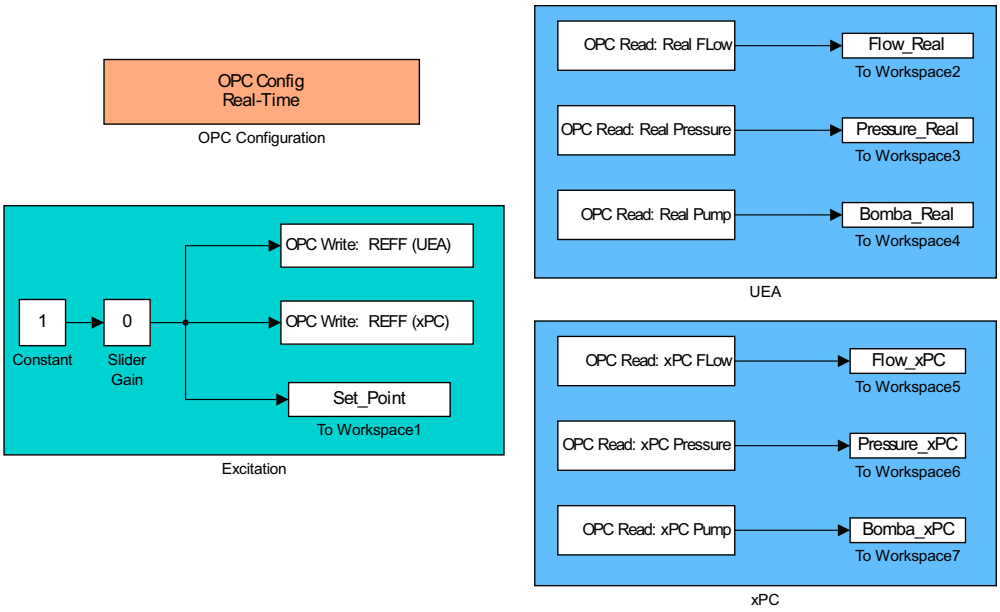
Para la activación del control de la variable presión, existe un bloque equivalente al mostrado en la figura anterior.

5.5. Adquisición de datos por OPC

Teniendo en cuenta que la UEA y el RTDS cuentan con un controlador PLC, es posible adquirir datos directamente desde un servidor OPC sin cargar la memoria y tampoco afectar la lógica ejecutada.

El estímulo a los dos controladores de manera simultánea se realiza configurando las variables de los dos dispositivos en un solo servidor, se propone como servidor el software KepServer Ex v5.0. El acceso a las variables del servidor se efectúa desde un cliente en MATLAB/Simulink® que permite su escritura y lectura, como se muestra en el modelo de la Figura 52.

Figura 52: OPC Configuration



Fuente: Autores.

Las variables que se configuran en el modelo para cada controlador son:

UEA

- Set Point del control: escritura.
- Flujo UEA: lectura.
- Presión UEA: lectura.
- Consigna Bomba UEA: lectura.

RTDS

- Set Point del control: escritura.
- Flujo RTDS: lectura.
- Presión RTDS: lectura.
- Consigna Entrada RTDS: lectura.

De esta manera, la escritura de la excitación al sistema y la lectura de las salidas se elaboran de manera simultánea en el mismo software.

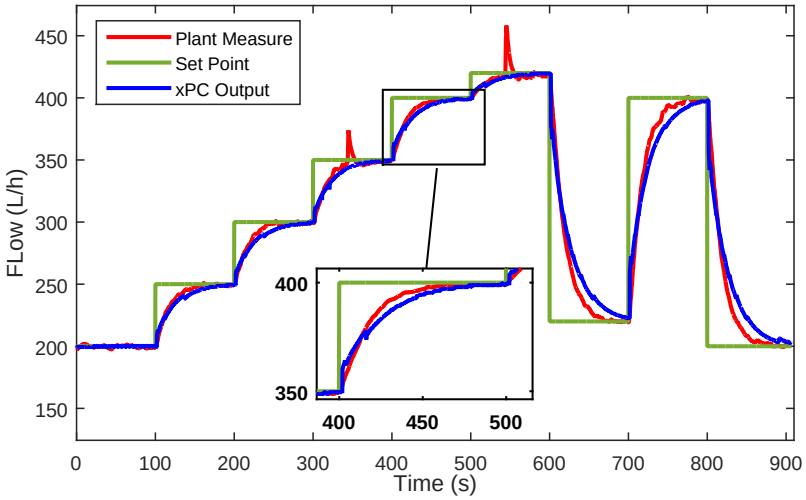
5.6. ANÁLISIS DEL CASO DE ESTUDIO IMPLEMENTADO EN EL RTDS CON UN CONTROL

El control diseñado en la sección "DISEÑO DEL CONTROL" se implementa en la UEA y en el RTDS, las condiciones de operación para la validación consisten en cambiar la referencia del control para los dos sistemas simultáneamente. Los resultados obtenidos se exponen a continuación.

5.6.1. Flujo con una válvula: variaciones en un rango de 200 L/h a 430 L/h.

En la Figura 53 se observa la respuesta de los dos sistemas con el control implementado. Los picos en la señal "Plant Measure" en los segundos 350 y 550 son un efecto al cambio de dirección en el flujo de la UEA, este cambio de flujo es necesario ya que con el sistema recirculando con una válvula de llenado abierta se satura el tanque de llenado, por otra parte, se observa que el simulador presenta un retardo con respecto a la UEA para cambios bruscos en el *Set Point*, en el segundo 680 la respuesta de la UEA con el control llega al *Set Point* mientras que el simulador presenta un retardo de 5 L/h.

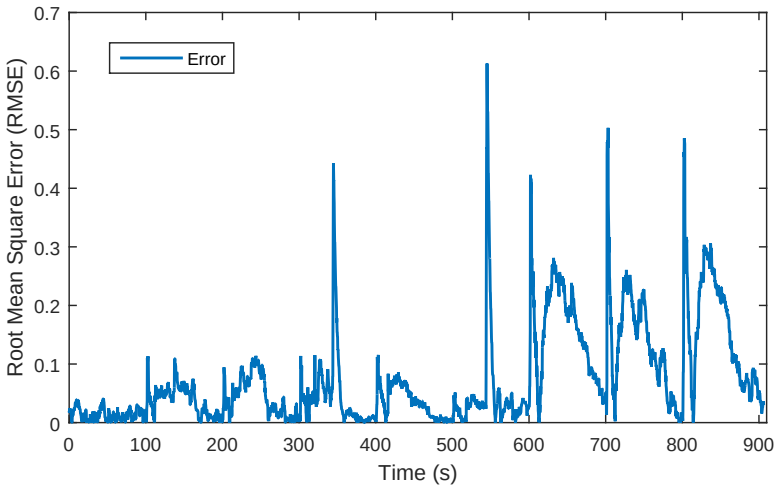
Figura 53: PI Flow Control UEA - xPC Comparison with One Valve



Fuente: Autores.

En la Figura 54, se visualiza que los aumentos del RMSE en los segundos 350 y 550 corresponden al cambio de flujo realizado. En los segundos 600, 700 y 800 se identifica un patrón compuesto por un pico que decrece con el tiempo, se evidencia el retardo del simulador para seguir la planta a cambios bruscos, cabe aclarar que el simulador se va acercando a la señal de la planta hasta alcanzarla y se comporta de la misma manera para perturbaciones positivas o negativas.

Figura 54: PI Flow Control Error UEA - xPC Comparison with One Valve



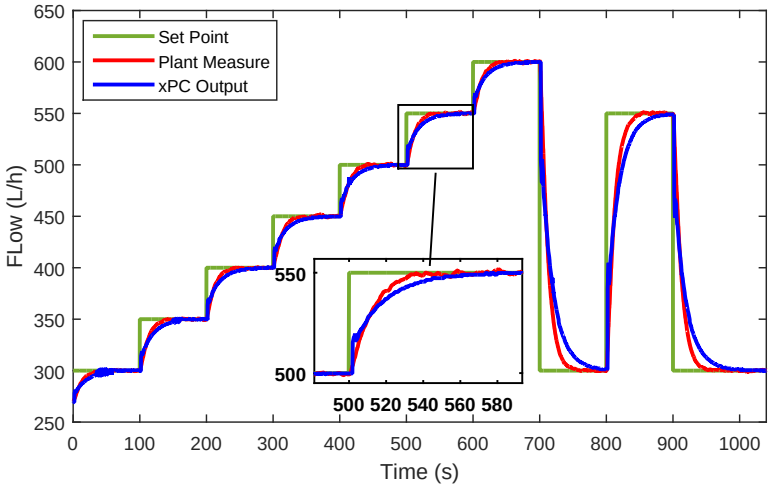
Fuente: Autores.

5.6.2. Flujo con dos válvula: variaciones en un rango de 300 L/h a 600 L/h.

En la Figura 55 se muestra la respuesta de los dos sistemas con el control implementado. En el sistema recirculando con las dos válvulas de llenado abiertas, no se presenta saturación en los tanques, por ello no se hace necesario cambiar la dirección del flujo.

El simulador representa de manera correcta la UEA con la implementación del control, sin embargo, en los segundos 750, 850 y 950, al igual que en la configuración con una válvula de llenado abierta, se presenta un retardo respecto a la planta ya que se genera un cambio brusco de *Set Point*.

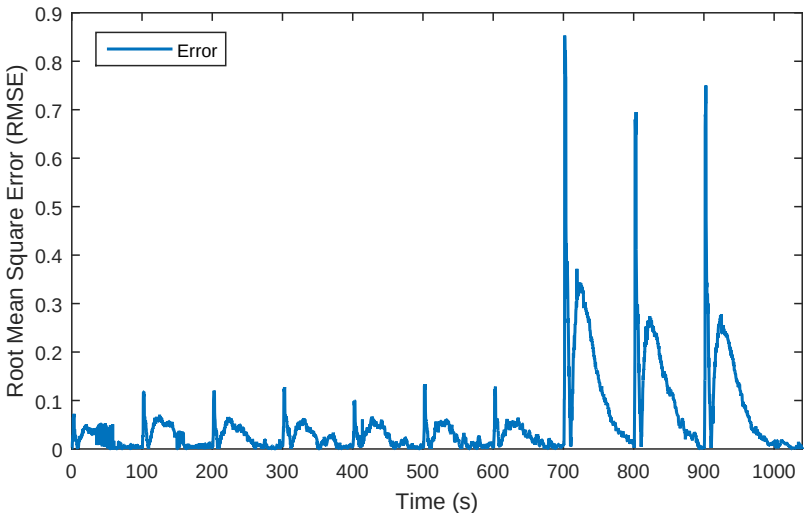
Figura 55: PI Flow Control UEA - xPC Comparison with Two Valves



Fuente: Autores.

En la Figura 56 se evidencia que el patrón en la amplitud del error para la variable de flujo es igual para las dos configuraciones de operación, además, en todos sus puntos de operación responde de la misma manera.

Figura 56: PI Flow Control Error UEA - xPC Comparison with Two Valves



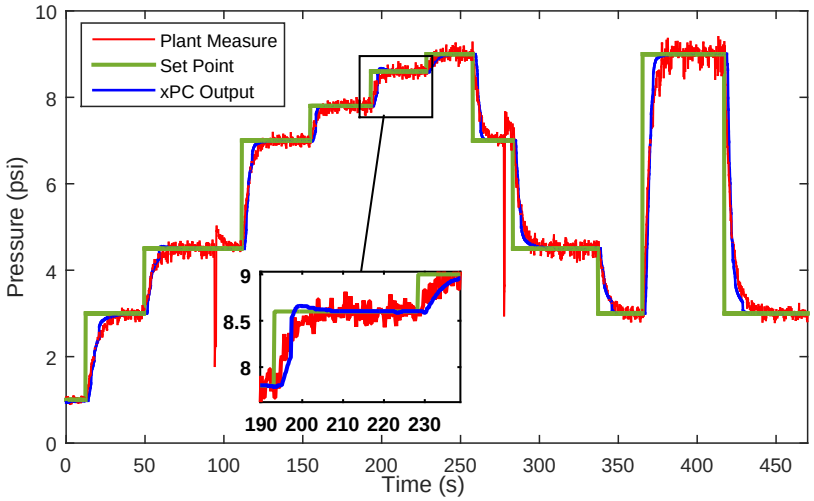
Fuente: Autores.

5.6.3. Presión con una válvula: variaciones en un rango de 1 psi a 9 psi.

En la Figura 57 se observa la respuesta de los sistemas con el control implementado para la variable de presión. Los picos en la señal "Plant Measure" presentados en los segundos 95 y 290 son un efecto al cambio de dirección en el flujo de la UEA, como se menciono anteriormente, este cambio es necesario debido a que el sistema recirculando con una válvula de llenado abierta satura el tanque de llenado.

Se identifica que en el segundo 200 hay un *overshoot* en la respuesta del simulador debido a que los rangos de operación en los que se identifica la UEA son muy cercanos, es decir, ofrecen una representación acertada de la UEA, aunque se detallan pequeños saltos en la señal en el cambio de un rango a otro. A diferencia de la variable de flujo, en presión el retardo de la respuesta del RTDS frente a la UEA ante cambios bruscos en el *Set Point* es mucho menor.

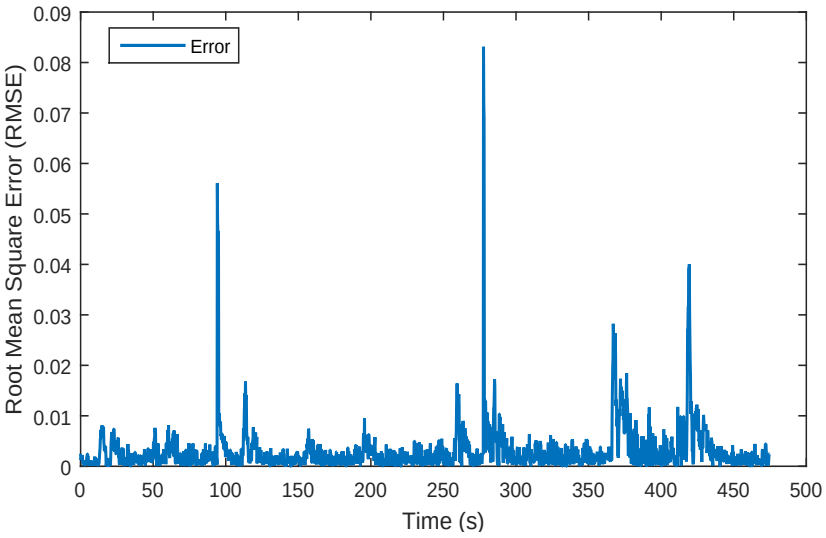
Figura 57: PI Pressure Control UEA - xPC Comparison with One Valve



Fuente: Autores.

En la Figura 58 los picos en el RMSE de los segundos 95 y 290 corresponden a el cambio en la dirección de flujo mencionada anteriormente. El patrón que se detalla desde el segundo 370 evidencia el retardo de la respuesta del simulador respecto a la planta a cambios bruscos y la recuperación del seguimiento, la amplitud máxima del RMSE es de 0.04 *psi* y el tiempo que toma el simulador para recuperar el comportamiento exacto de la planta es de 50 s.

Figura 58: PI Pressure Control Error UEA - xPC Comparison with One Valve

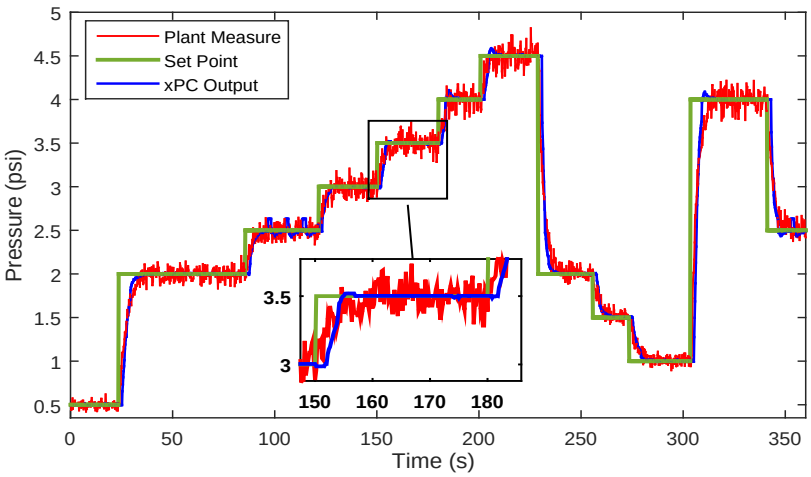


Fuente: Autores.

5.6.4. Presión con dos válvula: variaciones en un rango de 0.5 psi a 4.5 psi.

En la Figura 59 se muestra la respuesta del simulador y la UEA con el control implementado. Como se nombra anteriormente, los rangos de operación para la variable de presión están muy cercanos y producen saltos en la señal, esto se puede evidenciar al asignar un *Set Point* de 2.5 *psi* ya que el simulador opera justo en el límite de dos rangos de operación lo que genera 3 pequeños picos en la señal. Para la configuración con las dos válvulas de llenado abiertas la presión máxima es de 5 *psi* lo que genera que las oscilaciones en la medida de la UEA sean notorias.

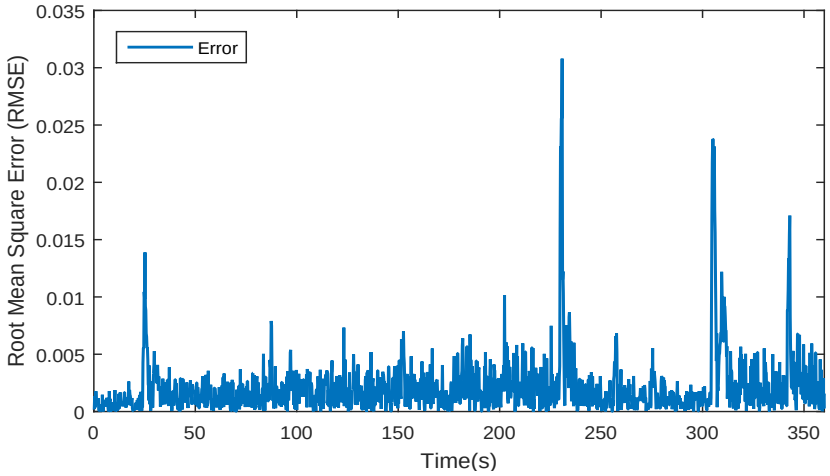
Figura 59: PI Pressure Control UEA - xPC Comparison with Two Valves



Fuente: Autores.

En la Figura 60, se visualiza que el valor máximo del error es de 0.03 *psi*, este error se presenta para un cambio brusco del *Set Point* en el segundo 240, al igual que en los segundos 300 y 350. Se evidencia que el simulador para la variable de presión operando con dos válvulas de llenado abiertas presenta una respuesta más rápida y la corrección del error se genera en 10 s.

Figura 60: PI Pressure Control Error UEA - xPC Comparison with Two Valves



Fuente: Autores.

Se analiza que el RTDS funciona de forma correcta para la variable de flujo con un RMSE máximo de 0.9 L/h y para la variable de presión con un RMSE máximo 0.009 psi.

6. CONCLUSIONES

Para el desarrollo del RTDS, se realiza el análisis de 3 casos de estudio y se concluye que el sistema que cumple con todos los requerimientos para la implementación es la UEA. Por medio del método de la curva de reacción, se define una representación matemática de la planta para todos los puntos de operación, esta identificación tiene un RMSE máximo de 0.23 L/h para la variable de flujo y 0.015 *psi* para la variable de presión.

Para diseñar la arquitectura de software, se realiza un análisis de las comunicaciones correspondientes entre los componentes del simulador (Host PC, Target PC, DAQ y PLC), definiendo los protocolos de comunicación a utilizar y el software necesario para la implementación, logrando un sistema HIL. Por otro lado, para diseñar la arquitectura de hardware se consideran los componentes con los que cuenta el Centro de Desarrollo Tecnológico (CDT) de la Universidad de La Salle, ya que son los disponibles para el desarrollo del RTDS.

Con el fin de evaluar la exactitud entre la UEA y el RTDS desarrollado, se diseña un control PI para la variable de flujo y uno para la variable de presión, estos controles son programados en los dos sistemas y se aplica como consigna una escalera de valores deseados para cada variable, por último, se realiza la adquisición de datos de la respuesta del simulador y la planta real simultáneamente, obteniendo un RMSE máximo de 0.9 L/h para la variable de flujo y 0.09 *psi* para la variable de presión.

El RTDS desarrollado, permite validar la efectividad de controles diseñados para las variables de flujo y presión antes de ser aplicados al sistema real, además de ofrecer una predicción de la respuesta de la planta en puntos críticos de operación, este procedimiento permite evitar el deterioro de la unidad en caso de que el control exija a los elementos físicos acciones que estos no puedan realizar y por lo tanto causar el daño de los mismos. Teniendo en cuenta que el simulador opera en cuatro configuraciones diferentes, la validación de los controles debe ser evaluada independientemente, es decir, el simulador permite cargar un modelo a la vez.

El prototipo del simulador de la UEA construido tiene un costo de 18'000.000 de pesos, mientras que el precio de adquirir una UEA es de 70'000.000 de pesos, esta comparación y los resultados del simulador pueden determinar que el RTDS es una herramienta útil y rentable para el ensayo de técnicas de control; también puede ser empleado para realizar simulaciones en tiempo real de otros sistemas siempre y cuando se cuenta con el modelo de los mismos.

Para un trabajo futuro, se considera evaluar el comportamiento de las variables de temperatura, peso y nivel con el fin de poder incluir sus modelos en el RTDS, por otra parte, construir un *rack* que contenga todos los componentes del simulador con el objetivo de lograr un hardware más estructurado y asequible.

BIBLIOGRAFÍA

Abdullah Mohamed, D. A. (2013). Creating real-time operation system based on xpc target kernel. *IJRTE*.

Albert, J. (2003). *Teaching statistics using baseball*. Mathematical Association of America.

Alfaro, V. M. (2007). Método de identificación de modelos de orden reducido de tres puntos 123c. *Escuela de Ingeniería Eléctrica, Universidad de Costa Rica*.

Ang, C. G. . L. Y., K. H. (2005). Pid control system analysis, design, and technology. *IEEE Transactions on Control Systems Technology*.

Barreto, H., y Howland, F. (2006). *Introductory econometrics: Using monte carlo simulation with microsoft excel*. Cambridge University Press.

Bemporad, P. A. (2011). Automatic control 2. *University of Trento*.

Cannon, S. J. R. H. (1967). *on the control of unestable systems, in automat. remote contr.* IFAC.

Carne, E. B. (2004). *A professional's guide to data communication in a tcp/ip world*. Artech House.

Corriou, J.-P. (2013). *Process control: Theory and applications*. Springer Science.

Dennison, P. E., y Roberts, D. A. (2003). Endmember selection for multiple endmember spectral mixture analysis using endmember average rmse. *Remote Sensing of Environment*, *87*(2–3), 123 - 135.

Deon Reynders, E. W. (2003). *Practical tcp/ip and ethernet networking*. Newnes.

de sistemas y automática, I. (1996). diseño de sistemas de control universidad de oviedo.

Dunlop, J. (2004). *Photovoltaics systems engineering*. CRCPress.

G. Randolf, R. M. M. (2005). Test system design for hardware-in-the-loop evaluation of pem fuel cells and auxiliaries. *Power Sources*, 392-396.

Haugen, F. (2010). Discretization of simulator, filter, and pid controller. *TechTeach*.

Held, G. (2002). *The abcs of tcp/ip*. CRC Press.

Hughes, T. A. (2002). *Measurement and control basics*. ISA Press.

Islam, G., Muyeen, S., Al-Durra, A., y Hasanien, H. (2016). Rtds implementation of an improved sliding mode based inverter controller for pv system. *ISA Transactions*, *62*, 50-59.

John, T. M., Karl-Heinz. (2001). *Iec 61131-3: Programming industrial automation systems: Concepts and programming languages, requirements for programming systems*. Springer.

J. Schoukens, R. P. (1991). Identification of linear systems. *Pergamon Press*.

Kezunovic, M., Domaszewicz, J., Skendzic, V., Aganagic, M., Bladow, J. K., McKenna, S. M., y Hamai, D. M. (1996). Design, implementation and validation of a real-time digital simulator for protection relay testing. *IEEE Transactions on Power Delivery*, *11*(1), 158-164.

K.J.Astrom, y K.Furuta. (1996). Swinging up a pendulum by energy control. *Tokio Institute of Technology*.

Kuffel, R., Giesbrecht, J., Maguire, T., Wierckx, R. P., y McLaren, P. (1995). Rtds - a fully digital power system simulator operating in real time. En (p. 19).

Kuo, B. C. (1996). *Sistemas de control automático*. Pearson Education.

Lewis, R. W. (1998). *Programming industrial control systems using iec 1131-3*. IEEE.
Liptak, B. G. (2005). *Instrument engineers' handbook, volume two: Process controland optimization*. CRC Press.
Lipták, B. G. (2013). *Process control: Instrument engineersh́andbook*. Butterworth-Heinemann.
Liu, Y., Goorts, K., Ashasi-Sorkhabi, A., Mercan, O., y Narasimhan, S. (2016). A state space-based explicit integration method for real-time hybrid simulation. *Structural Control and Health Monitoring*, *23*(4), 641-658.
Lu, B., Wu, X., Figueroa, H., y Monti, A. (2007). A low-cost real-time hardware-in-the-loop testing approach of power electronics controls. *IEEE Transactions on Industrial Electronics*, *54*(2), 919-931.
Luo, A. C. J. (2015). *discretization and implicit mapping dynamics*. Springer.
María Elena Anaya Pérez, J. H. P. R. F. M. V., Víctor Hugo Benítez Baltazar. (2014). Dieño de controladores p, pi y pid para el comportamiento dinámico de un servo sistema hidráulico, basado en desarrollo experimental. *EPISTEMUS*.
MathWorks. (2014a). *Opc toolbox*. The MathWorks, Inc.
MathWorks. (2014b). *Simulink real-time; getting started guide*. The MathWorks, Inc.
McLaren, P. G., Kuffel, R., Wierckx, R., Giesbrecht, J., y Arendt, L. (1992). A real time digital simulator for testing relays. *IEEE Transactions on Delivery*, *7*(1), 207-213.
Miller, P. (2009). *Tcp-ip, the ultimate protocol guide*. Universal Publishers.
Ogata, K. (2003). *Ingeniería de control moderna*. Pearson Education.
Ogata, K., y Yang, Y. (2010). *Modern control engineering*. Pearson Education.
Opc foundation. (2017).
Paquin, J. B. (2010). The what, where and why or real time simulation. *IEEE*.
Ponsa Asensio, R. V. A., Pere Vilanova. (2006). *Automatización de procesos mediante la guía gemma*. Universitat Politecnica de Catalunya.
Seborg, M. D., Edgar. (2004). *Process dynamics and control*. Wiley.
Sevilla, A. G. P., y Fernández, F. O. A. (2013). Evolución de las redes eléctricas hacia smart grid en países de la región andina. *Revista Educación en Ingeniería*, *8*(15), 48-61.
Sharma, P. (2013). Modeling fluid system.
Shi, P., Miao, L., Zou, G., y Jiao, X. (2009). Development of uniform hardware driver for real-time windows and xpc target. , *1*, 377-380.
Siano, P. (2014). Demand response and smart grids a survey. *System Dynamics*.
SIEMENS. (2016). Escalización de valores analógicos.
Spartacus Gomáriz, J. M. M. R., Domingo Biel. (2004). *Teoría de control*. Universitat Politecnica de Catalunya.
Uriarte, F. M., y Butler-purry, K. L. (2006). Real-time simulation using pc-based kernels. En (p. 1991-1995).
White, D. J. R. (1997). Mathematical modeling of engineering systems. *Spring*.
Wolfgang Mahnke, M. D., Stefan-Helmut Leitner. (2009). *Opc unified architecture*. Springer.
Zhang, J., y Cheng, M. (2015). A real time testing system for wind turbine controller with

xpc target machine. *International Journal of Electrical Power and Energy Systems*, *73*, 132-140.

GLOSARIO

BIOS: Pieza fundamental de toda computadora, su función principal es permitir el arranque del PC.
Control: Permite actuar sobre los periféricos de salida del controlador.
Controlador: Dispositivo que puede ser programado para controlar los periféricos del mismo.
DAQ: Sistema que consta de hardware de medidas de señales y una PC con software programable.
Drivers: Es el software que comunica los periféricos con el sistema operativo.
Encoder: Dispositivo electromecánico que convierte la posición angular de un eje en un código análogo o digital.
HIL: Técnica para el desarrollo y ensayo de sistemas embebidos en tiempo real.
MATLAB: Herramienta de software matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M).
Offset: Desplazamiento vertical de un señal.
OS: Sistema Operativo, Conjunto de órdenes y programas que controlan los procesos básicos de una computadora y permiten el funcionamiento de otros programas.
Planta: Sistema físico del caso de estudio.
PLC: Controlador Lógico Programable.
Real-Time: Se considera en Real-Time cuando la respuesta del simulador es tan rápida como el sistema lo requiera, con el tiempo de muestreo dependiendo de la frecuencia del sistema.
RTDS: Real Time Digital Simulator por sus siglas. Implementación de software y hardware para simular un sistema con respuesta rápida (Tiempo real).
SIL: Ambiente de programación para los simuladores en tiempo real.
Simulink: Entorno de programación visual, que funciona sobre MATLAB.
Sistema embebido: Es un controlador que incluye interfaces de entrada/salida en el mismo equipo. Poseen un interfaz externo para efectuar un monitoreo del estado del sistema.
Solver: Este parámetro define la parada inicial y el tiempo de muestreo para su aplicación de destino.
ST: Texto Estructurado, lenguaje de programación.
Target: Sistema físico donde se descarga el simulador (Modelo, drivers).
TCP-IP: "Protocolo de control de transmisión/Protocolo de Internet", representa todas las reglas de comunicación para Internet y se basa en la noción de dirección IP.
TET: Task Ejecution Time.
UEA: Unidad de Entrenamiento en Automatización.
Voltaje RMS: Root Mean Square, el valor RMS es el valor del voltaje o corriente en ac.
xPC Target: Es un ambiente para el desarrollo de sistemas en tiempo real. Requiere de un computador maestro que parametriza el modelo y un computador esclavo que ejecuta las aplicaciones.
xPC: Herramienta de Simulink que permite el control y HIL de prototipos, adicionalmente el desarrollo de simulaciones en tiempo real.

Anexos

Funciones de transferencia para los diferentes modelos.

Flujo:

Tabla 4: Flow One Valve Transfer Functions.

Nombre	Función de Transferencia
IFD12	$\frac{1{,}17e^{-4}Z^2+5{,}06e^{4-4}Z+3{,}88e^{-4}}{Z^2-0{,}66Z+0{,}29}$
IFD13	$\frac{6{,}64e^{-5}Z^2+4{,}56e^{-4}Z+3{,}89e^{-4}}{Z^2-0{,}79Z-0{,}16}$
IFD14	$\frac{-5{,}78e^{-5}Z^2+3{,}02e^{-4}Z+3{,}6e^{-4}}{Z^2-1{,}12Z-0{,}15}$
IFD15	$\frac{-1{,}7e^{-4}Z^2+2{,}3e^{-4}Z+4e^{-4}}{Z^2-1{,}38Z+0{,}41}$
IFD16	$\frac{-6-2e^{-5}Z^2+2{,}98e^{-4}Z+3{,}6e^{-4}}{Z^2-1{,}13Z+0{,}17}$
IFD17	$\frac{-2{,}27e^{-4}Z^2+1{,}65e^{-4}Z+3{,}91e^{-4}}{Z^2-1{,}64Z+0{,}56}$
IFD18	$\frac{-2{,}23e^{-4}Z^2+1{,}43e^{-4}Z+3{,}65e^{-4}}{Z^2-1{,}57Z+0{,}59}$
IFD19	$\frac{-2{,}3e^{-4}Z^2+1{,}41e^{-4}Z+3{,}71e^{-4}}{Z^2-1{,}58Z+0{,}6}$
IFD20	$\frac{-2{,}52e^{-4}Z^2+1{,}16e^{-4}Z+3{,}68e^{-4}}{Z^2--1{,}64Z+0{,}66}$
IFD21	$\frac{-2{,}36e^{-4}Z^2+1{,}16e^{-4}Z+3{,}52e^{-4}}{Z^2-1{,}63Z+0{,}65}$
IFD22	$\frac{-2{,}34e^{-4}Z^2+1{,}12e^{-4}Z+3{,}46e^{-4}}{Z^2-1{,}64Z+0{,}65}$
IFD23	$\frac{-2{,}56e^{-4}Z^2+1{,}02e^{-4}Z+3{,}58e^{-4}}{Z^2-1{,}68Z+0{,}69}$
IFD24	$\frac{-2{,}52e^{-4}Z^2+8{,}75e^{-5}Z+3{,}4e^{-4}}{Z^2-1{,}7Z+0{,}71}$
IFD25	$\frac{-2{,}61e^{-4}Z^2+8{,}11e^{-5}Z+3{,}42e^{-4}}{Z^2-1{,}72Z+0{,}73}$
IFD26	$\frac{-2{,}61e^{-4}Z^2+8{,}11e^{-5}Z+3{,}42e^{-4}}{Z^2-1{,}72Z+0{,}73}$

Tabla 5: Flow Two Valves Transfer Functions.

Nombre	Función de Transferencia
IFD12	$\frac{-2{,}03e^{-4}Z^2+2{,}38e^{-4}Z+4{,}41e^{-4}}{Z^2-1{,}42Z+0{,}44}$
IFD13	$\frac{-2{,}25e^{-4}Z^2+2{,}28e^{-4}Z+4{,}53e^{-4}}{Z^2-1{,}46Z+0{,}48}$
IFD14	$\frac{-2{,}57e^{-4}Z^2+2{,}01e^{-4}Z+4{,}59e^{-4}}{Z^2-1{,}52Z+0{,}54}$
IFD15	$\frac{-2{,}65e^{-4}Z^2+1{,}97e^{-4}Z+4{,}62e^{-4}}{Z^2-1{,}54Z+0{,}55}$
IFD16	$\frac{-2{,}96e^{-4}Z^2+1{,}77e^{-4}Z+4{,}73e^{-4}}{Z^2-1{,}57Z+0{,}60}$
IFD17	$\frac{-2{,}75e^{-4}Z^2+1{,}81e^{-4}Z+4{,}56e^{-4}}{Z^2-1{,}57Z+0{,}58}$
IFD18	$\frac{-3{,}21e^{-4}Z^2+1{,}59e^{-4}Z+4{,}8e^{-4}}{Z^2-1{,}63Z+0{,}64}$
IFD19	$\frac{-3{,}44e^{-4}Z^2+1{,}39e^{-4}Z+4{,}83e^{-4}}{Z^2-1{,}67Z+0{,}68}$
IFD20	$\frac{-3{,}8e^{-4}Z^2+1{,}99e^{-4}Z+4{,}99e^{-4}}{Z^2-1{,}72Z+0{,}73}$
IFD21	$\frac{-3{,}72e^{-4}Z^2+1{,}18e^{-4}Z+4{,}9e^{-4}}{Z^2-1{,}72Z+0{,}73}$
IFD22	$\frac{-3{,}68e^{-4}Z^2+1{,}17e^{-4}Z+4{,}85e^{-4}}{Z^2-1{,}72Z+0{,}73}$
IFD23	$\frac{-3{,}62e^{-4}Z^2+1{,}13e^{-4}Z+4{,}75e^{-4}}{Z^2-1{,}72Z+0{,}73}$
IFD24	$\frac{-3{,}61e^{-4}Z^2+1{,}08e^{-4}Z+4{,}7e^{-4}}{Z^2-1{,}73Z+0{,}74}$
IFD25	$\frac{-3{,}97e^{-4}Z^2+1{,}05e^{-4}Z+5{,}02e^{-4}}{Z^2-1{,}75Z+0{,}76}$
IFD26	$\frac{-4{,}03e^{-4}Z^2+9{,}45e^{-5}Z+4{,}97e^{-4}}{Z^2-1{,}77Z+0{,}78}$

Presión:

Tabla 6: Pressure One Valve Transfer Functions.

Nombre	Función de Transferencia
IDP12	$\frac{-3{,}13e^{-5}Z^2+2{,}80e^{-5}Z+5{,}93e^{-5}}{Z^2-1{,}05Z+0{,}28}$
IDP13	$\frac{-4{,}16e^{-5}Z^2+3{,}19e^{-5}Z+7{,}35e^{-5}}{Z^2-1{,}02Z+0{,}26}$
IDP14	$\frac{-2{,}19e^{-5}Z^2+2{,}65e^{-5}Z+4{,}84e^{-5}}{Z^2-1{,}12Z+0{,}30}$
IDP15	$\frac{-2{,}44e^{-5}Z^2+2{,}23e^{-5}Z+4{,}68e^{-5}}{Z^2-1{,}22Z+0{,}36}$
IDP16	$\frac{-3{,}32e^{-5}Z^2+2{,}49e^{-5}Z+5{,}82e^{-5}}{Z^2-1{,}22Z+0{,}37}$
IDP17	$\frac{-3{,}20e^{-5}Z^2+2{,}21e^{-5}Z+5{,}41e^{-5}}{Z^2-1{,}28Z+0{,}40}$
IDP18	$\frac{-2{,}23e^{-5}Z^2+1{,}92e^{-5}Z+4{,}15e^{-5}}{Z^2-1{,}31Z+0{,}41}$
IDP19	$\frac{-2{,}21e^{-5}Z^2+1{,}72e^{-5}Z+3{,}94e^{-5}}{Z^2-1{,}36Z+0{,}44}$
IDP20	$\frac{-2{,}20e^{-5}Z^2+1{,}59e^{-5}Z+3{,}80e^{-5}}{Z^2-1{,}39Z+0{,}47}$
IDP21	$\frac{-2{,}35e^{-5}Z^2+1{,}45e^{-5}Z+3{,}81e^{-5}}{Z^2-1{,}43Z+0{,}50}$
IDP22	$\frac{-2{,}37e^{-5}Z^2+1{,}33e^{-5}Z+3{,}70e^{-5}}{Z^2-1{,}47Z+0{,}53}$
IDP23	$\frac{-2{,}22e^{-5}Z^2+1{,}48e^{-5}Z+3{,}70e^{-5}}{Z^2-1{,}43Z+0{,}50}$
IDP24	$\frac{-2{,}04e^{-5}Z^2+1{,}08e^{-5}Z+3{,}13e^{-5}}{SZ^2-1{,}51Z+0{,}56}$
IDP25	$\frac{-2{,}55e^{-5}Z^2+1{,}14e^{-5}Z+3{,}69e^{-5}}{Z^2-1{,}54Z+0{,}58}$
IDP26	$\frac{-2{,}55e^{-5}Z^2+1{,}14e^{-5}Z+3{,}69e^{-5}}{Z^2-1{,}59Z+0{,}63}$

Tabla 7: Pressure Two Valves Transfer Functions.

Nombre	Función de Transferencia
IDP12	$\frac{-1{,}20e^{-6}Z^2+1{,}26e^{-5}Z+1{,}38e^{-5}}{Z^2-0{,}88Z+0{,}06}$
IDP13	$\frac{-5{,}49e^{-6}Z^2+1{,}77e^{-5}Z+2{,}32e^{-5}}{Z^2-0{,}96Z+0{,}17}$
IDP14	$\frac{-7{,}66e^{-6}Z^2+2{,}82e^{-5}Z+3{,}59e^{-5}}{Z^2-0{,}87Z+0{,}14}$
IDP15	$\frac{-2{,}06e^{-5}Z^2+2{,}77e^{-5}Z+4{,}83e^{-5}}{Z^2-0{,}98Z+0{,}23}$
IDP16	$\frac{-2{,}20e^{-5}Z^2+2{,}84e^{-5}Z+5{,}04e^{-5}}{Z^2-0{,}98Z+0{,}24}$
IDP17	$\frac{-2{,}44e^{-5}Z^2+3{,}02e^{-5}Z+5{,}46e^{-5}}{Z^2-0{,}98Z+0{,}24}$
IDP18	$\frac{-2{,}45e^{-5}Z^2+2{,}28e^{-5}Z+4{,}74e^{-5}}{Z^2-1{,}13Z+0{,}31}$
IDP19	$\frac{-3{,}31e^{-5}Z^2+2{,}72e^{-5}Z+6{,}03e^{-5}}{Z^2-1{,}07Z+0{,}28}$
IDP20	$\frac{-1{,}42e^{-5}Z^2+1{,}60e^{-5}Z+3{,}03e^{-5}}{Z^2-1{,}24Z+0{,}36}$
IDP21	$\frac{-1{,}42e^{-5}Z^2+1{,}60e^{-5}Z+3{,}03e^{-5}}{Z^2-1{,}24Z+0{,}36}$
IDP22	$\frac{-1{,}41e^{-5}Z^2+1{,}29e^{-5}Z+2{,}70e^{-5}}{Z^2-1{,}33Z+0{,}42}$
IDP23	$\frac{-1{,}41e^{-5}Z^2+1{,}28e^{-5}Z+2{,}70e^{-5}}{Z^2-1{,}34Z+0{,}42}$
IDP24	$\frac{-1{,}42e^{-5}Z^2+1{,}12e^{-5}Z+2{,}55e^{-5}}{Z^2-1{,}39Z+0{,}46}$
IDP25	$\frac{-1{,}32e^{-5}Z^2+1{,}20e^{-5}Z+2{,}53e^{-5}}{Z^2-1{,}36Z+0{,}44}$
IDP26	$\frac{-1{,}33e^{-5}Z^2+1{,}12e^{-5}Z+2{,}45e^{-5}}{Z^2-1{,}38Z+0{,}45}$