UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA QUÍMICA Y BIOTECNOLOGÍA

DISEÑO E IMPLEMENTACIÓN DE UNA INFRAESTRUCTURA PARA UN SISTEMA DE CONTROL DISTRIBUIDO (DCS).

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL QUÍMICO.

ÁLVARO STEFANO CERONI SALAZAR.

PROFESOR GUÍA:
J. CRISTIAN SALGADO HERRERA.

MIEMBROS DE LA COMISIÓN:
A. LEANDRO HERRERA ZEPPELIN.
JORGE CASTILLO GUZMÁN.

SANTIAGO DE CHILE
2015

RESUMEN DE LA MEMORIA PARA OPTAR AL
TÍTULO DE INGENIERO CIVIL QUÍMICO.
POR: ÁLVARO STEFANO CERONI SALAZAR.
FECHA: 2015.
PROF. GUÍA: J. CRISTIAN SALGADO HERRERA.

DISEÑO E IMPLEMENTACIÓN DE UNA INFRAESTRUCTURA PARA UN SISTEMA DE CONTROL DISTRIBUIDO (DCS).

En los últimos años, la masificación de sistemas embebidos y de computadores de placa reducida (SBC) han permitido el diseño, desarrollo y automatización de hardware de bajo costo para aplicaciones multidisciplinarias tales como impresoras 3D, sistemas de alarmas y climatización. Sin embargo, a nivel de laboratorios y pequeñas industrias su uso se limita a la creación de sensores, es por ello que esta memoria explora el diseño de un sistema de control enfocado en estos segmentos.

Este sistema de control está compuesto por un servidor de control que posee una base de datos, un servidor web que actúa como interfaz humano-máquina (HMI), y un elemento de control, conectados mediante una red inalámbrica. El elemento de control corresponde al dispositivo encargado de obtener desde los sensores, datos sobre las variables de entrada, manipular los actuadores a través de los puertos de salida y posee además, la capacidad de implementar una ley de control a través de un controlador en modo de supervisión o como controlador digital directo. Por otro lado, el servidor de control tiene la función de registrar las mediciones de los sensores del equipo conectados al sistema, además, también puede registrar modificaciones hechas al sistema desde el HMI.

El sistema de control diseñado se implementó en el equipo HL630, equipo que posee un circuito de calefacción de agua que incluye un radiador con ventilador, calefactor, una bomba de potencia regulable y un sensor de flujo. Para esto, se utilizó como servidor de control el SBC Raspberry Pi B+, y como elemento de control el sistema embebido Arduino Yún. Para conectar el equipo al elemento de control, se utilizó circuitos y dispositivos electrónicos con el fin de enlazar los actuadores y sensores. Asimismo, se observó la presencia de interferencias electromagnéticas (EMI), problema que se solucionó utilizando una jaula de Faraday. Además, se implementó un controlador digital directo de tipo PID con PWM debido a que no es posible utilizar el elemento de control en modo de supervisión.

Finalmente, para la sintonización del controlador, se utilizó un modelo empírico basado en la respuesta del sistema a un pulso de 15 seg., y con ello, mediante el ajuste de parámetros de simulaciones para un sistema con una función de transferencia de cierto orden, se determinó que la función de transferencia que mejor representa el sistema es de orden-(3,2). Así, se sintonizó el controlador utilizando algoritmos basados en la minimización de overshoot en función del tiempo de estabilización del sistema para problemas de regulación y servo-control, en base a una simulación del sistema con controlador PID continuo, para obtener parámetros utilizados como base de la ley de control implementada y mediante un análisis de sensibilidad se obtuvo un valor definitivo.

Posteriormente, se probó de forma experimental que al aumentar 10 veces el valor destinado a eliminar el windup reset aumenta la amplitud de oscilaciones y el sistema converge a estado estacionario oscilando alrededor del valor del setpoint. De igual manera, también se experimentó aumentar al doble el periodo de la señal de referencia del PWM y se observó que el controlador no cambia la amplitud, ni el tiempo de estabilización, sin embargo, se genera una oscilación de alta frecuencia de amplitud constante.

Así, fue posible diseñar una infraestructura de un sistema de control distribuido con las caracteristicas ya mencionadas, que se implementó en un equipo de laboratorio y de forma exitosa fue posible monitorear y controlar el equipo de forma remota a través del HMI. A su vez, este proyecto permitió ver la factibilidad técnica de implementar sistemas de control de bajo costo en algunos procesos de pequeñas escala, a modo de reemplazar la utilización de costosos sistemas industriales.

Agradecimientos

Comenzando por mi familia, a mis padres y hermanos, quienes con todo su esfuerzo y apoyo me permitieron sacar esto adelante, por dejarme ir lejos para alcanzar mis metas y obtener este gran logro, que no solo es mío sino que Uds., se lo merecen rotundamente. Gracias por su confianza, por todas las herramientas y oportunidades que me permitieron ser lo que soy, y todo lo que puedo lograr.

A mis amigos del colegio: Mario, Matías, Nico, Vane y a aquellos que por diversas razones hicieron que nuestros caminos se separaran. Agradecer a quienes seguimos reencontrándonos a pesar de la distancia y las obligaciones, pero que aún perduran lindos recuerdos, la amistad, el compañerismo y las mismas ganas de hace más de 6 años.

A aquellos que la vida me permitió conocer, a Cristian, Igor, Ismael, Lorenzo, Palta, Pancho, Piga y Simón, que se convirtieron en una segunda familia en Santiago, quienes con conversaciones, almuerzos, asados, torneos de FIFA/PES, entrenamientos de días sábados y más cosas, me permitieron no solo ir a Santiago a desarrollarme profesionalmente, sino que conocer buenas personas para luego convertirlas en buenos amigos.

A Eli, Igor y Fabián por su ayuda, compañía y largos momentos de estudios, trabajos e informes durante este largo camino, y claramente como no olvidar almuerzos, cumpleaños y tiempos de recreación.

Gracias Andrea, Carla, Dani, Daniel, Ximena y Willly, por sus comentarios y consejos, por darme la oportunidad de conocer un área un tanto alejada de mi interés, compartir momentos no relacionados con las reuniones y me permitieron conocerlos un poco más.

Al profesor Salgado quien me permitió llevar adelante esta alocada y desafiante idea, al profesor Castillo por toda la ayuda y buena disposición, y a todas aquellas personas que en algún momento nos encontramos y me brindaron de su ayuda o pasar un buen momento.

Finalmente, a ti Cony, un millón de gracias por tu esfuerzo, tu ayuda, tu compañía y tu compresión. Por esos viajes eternos, por nuestras aventuras, nuestras conversaciones interminables y todo este tiempo juntos. Gracias por cuidarme, por quererme, hacerme feliz y por todos esos sueños por cumplir.

Tabla de contenido

Índice de tablas

Índice de ilustraciones

Capítulo 1

Introducción

El fin de toda industria de procesos es generar un cierto producto a partir de determinadas materias primas, utilizando recursos humanos, equipos, instalaciones y energía. El proceso, durante su operación, debe tener mecanismos que permitan enfrentar cambios externo del sistema (perturbaciones), para cumplir con requerimientos de seguridad, sociales, técnicos y económicos, y es por ello que se hace indispensable el monitoreo y manipulación de los equipos utilizados en sus procesos.

Para ello, los sistemas control deben asegurar una producción segura para los operarios, la comunidad que los rodea y su entorno en general; además, que satisfaga las especificaciones de producción, que respete las restricciones medioambientales impuestas por las autoridades, que también considere las restricciones operacionales de ciertos equipos y procesos, y que el proceso sea económicamente rentable.

En la gran industria, los sistemas de control y automatización se encuentran muy desarrollados, donde grandes corporaciones brindan el servicio de instalación, mantención e incluso operación de estos sistemas. En cambio, en la pequeña industria o a nivel de laboratorios en universidades y centros de investigación no hay muchos o simplemente no existen debido al gran costo que conlleva instalar y operar un sistema de control, lo que deriva en un problema al momento de monitorear los diferentes equipos involucrados.

Por otro lado, en los últimos años el aumento de la popularidad del concepto del *Internet de las cosas*, permitió la masificación de sistemas embebidos y microcontroladores, y con ello el desarrollo de nueva tecnología a un menor costo. Entre estos sistemas se encuentran los Arduinos, los cuales son un sistema de desarrollo para tarjetas que incluye un microcontrolador, entradas y salidas análogas y digitales, y otros puertos de comunicación. Estas placas, son una de las más populares, debido principalmente a su fácil uso, creadas bajo el término de software y hardware libre lo que permite obtener este producto casi a precio costo.

Considerando el contexto actual de los sistemas embebidos y la necesidad de monitoreo y automatización en procesos de pequeña y mediana escala, se diseñó como alternativa para

esta problemática un sistema de control utilizando estos dispositivos, de manera de contar con prestaciones similares a un sistema industrial, pero a un menor costo.

1.1. Descripción del proyecto

Este proyecto tiene como principal objetivo diseñar e implementar los componentes de una infraestructura de un sistema de control distribuido (DCS) que permitan monitorear, configurar e implementar el sistema de forma inalámbrica, basándose en un diseño simple, genérico y extensible para la fácil identificación e integración de los equipos y elección del tipo de control y sus parámetros.

Además, los componentes que posee el sistema corresponden a sistemas embebidos, como es el caso de Arduino para los elementos dedicados a controlar y monitorear el proceso, o computadores de placa reducida para alojar servidores que registren y permitan la interacción con el usuario. Debido al tipo de componentes que integra el sistema permite reducir costos relacionados al hardware y software debido a su masividad. Además, esto permite que tengan un mayor desarrollo y soporte, debido a múltiples usuarios que ocupan estos componentes.

El sistema propuesto en este documento está enfocado en plantas de pequeña escala o para laboratorios docentes o de investigación, ya que el costo de instalación y operación de sistemas de control convencionales son muy elevados haciendo inviable este tipo de tecnología (Ver Anexo A). Además, también permitiría ser utilizado en aplicaciones orientadas hacia la Domótica, donde este sistema podría controlar la temperatura y humedad de ambientes, controlar la luminosidad, entre otras tareas.

1.2. Objetivos

A continuación se muestran los objetivos generales y específicos planteados para este proyecto. Dichos objetivos son los que fijan el rumbo para el desarrollo de este documento.

1.2.1. Objetivo general

Diseñar una infraestructura para un sistema de control distribuido que contenga un sistema de monitoreo y manipulación remota, utilizando sistemas embebidos y una red inalámbrica para la comunicar estos componentes.

1.2.2. Objetivos específicos

De manera de cumplir los objetivos generales, se determinó los siguientes objetivos secundarios:

- Construir una base de datos que permita almacenar información acerca de los equipos a ser controlados, es decir, modelo, actuadores, sensores, tipo de interfaz de comunicación, etc.
- Crear un sistema que permita reconocer e identificar de forma manual el equipo conectado al dispositivo de control para obtener información desde la base de datos.
- Modificar un equipo para permitir la implementación y operación del sistema de control propuesto.
- Crear una página web que permita al usuario manipular y observar de forma remota el proceso.
- Diseñar el sistema de tal manera que sea fácil la integración de otros equipos.
- Elaborar un software que permita exportar información almacenada desde la base de datos a hojas de cálculo (por ejemplo, Excel).

1.3. Motivación

En los últimos años la popularización de sistemas embebidos, como BASIC Stamp, Arduino, Raspberry Pi, OLinuXino, Intel Edison, entre otros, han permitido a personas tener este tipo de tecnologías a un bajo precio. Esto sumado a la característica de ser fácilmente programables, permite que estas tecnologías sean ampliamente desarrolladas en un gran espectro de usos; como decoración, automatización en herramientas o elementos de casas, robótica, entre otros. Es por ello, que la idea de este proyecto es el escalamiento de estos tipos de componentes de un uso más bien doméstico y entusiasta para orientarlo a un uso industrial de pequeña escala.

El incluir este tipo de tecnologías permitiría a la industria, en especial a la de pequeña escala y laboratorios de investigación, disminuir sus costos, debido al bajo precio de estos componentes y la masificación de software (generalmente libre para este tipo de tecnologías).

1.4. Alcances

Dado que el objetivo principal de esta memoria es el diseño del sistema de control y su implementación, y debido a las restricciones de tiempo de esta memoria y de manera de cumplir los objetivos mencionados, ésta incluye el diseño del sistema de control y su implementación en el equipo HL630. Para ello, se realizó modificaciones en el equipo para conectar el elemento de control, se desarrolló software para crear el interfaz humano-máquina,

y para comprobar el funcionamiento del sistema se sintonizó el controlador y 2 pruebas de sensibilidad con 2 parámetros del controlador. Es por esto que este documento no contiene un análisis profundo del equipo HL630, como por ejemplo, analizar la respuesta del sistema frente a pertubaciones utilizando distintas configuraciones del controlador.

Capítulo 2

Marco teórico

2.1. Sistemas de control

Puesto que una planta química está afecta a una serie de restricciones de tipo ambientales, económicas, operacionales y sociales, los procesos involucrados requieren ser monitoreados para satisfacer estos requerimientos y cumplir el objetivo de producción. Una serie componentes (como sensores, actuadores, controladores, entre otros) y la intervención humana conforman un sistema de control.

Estos sistemas tienen por objetivo satisfacer las siguientes necesidades: [22]

1. Suprimir la influencia de perturbaciones
2. Asegurar la estabilidad del proceso
3. Optimizar el desempeño del proceso

Los sistemas de control han evolucionado desde el punto de cómo se transmite la información y cómo se toman decisiones. La primera implementación de sistemas de control fue utilizando dispositivos mecánicos, y el medio de transmisión de información se realizó mediante sistemas neumáticos. Debido a esto, los cálculos que podían realizar estos dispositivos no eran muy sofisticados, sin embargo, permitían realizar sumas, multiplicaciones e incluso integración y diferenciación. Algunos problemas típicos de estos sistemas eran la lentitud en la transferencia de datos y que además era susceptible a interferencia, y por otro lado, los dispositivos eran simples y tendían a desgastarse fácilmente [15]. A pesar de esto, estos sistemas siguen siendo utilizados en plantas modernas, en especial en lugares donde el uso de dispositivos electrónicos es peligroso, como en presencia de atmósferas explosivas o la presencia de sustancia inflamables [10].

Posteriormente, aparecieron sistemas análogos que permitieron transmitir estos datos a través de señales análogas, donde la velocidad de transmisión son cercanas a la velocidad de la luz, lo que mejora la efectividad del sistema de control. Sin embargo, el mayor problema de

estos sistemas es que son susceptibles a ruido, donde la calidad de la señal se ve perjudicada con la distancia. Para enfrentar esto nacen los sistemas digitales, que envían señales digitales a través de un medio eléctrico, pero codificada mediante números binarios. Esto permite enviar información a gran distancia sin ser afectada a interferencias o ruido. Lo anterior, y sumado al desarrollo de esta tecnología, da al controlador una mayor versatilidad, lo que se traduce en realizar cálculos más complejos. Debido a la naturaleza de medición de algunos sensores, es imposible eliminar completamente los sistemas de comunicación análogos, y para conectar ambos sistemas se utilizan transductores que permiten convertir señales análogas a digitales o viceversa.

Por otro lado, un computador de control puede funcionar de 2 modos: actuar en modo de supervisor o como controlador digital directo (DDC) [15]. El modo de supervisor produce una modificación en el setpoint de uno o más controladores locales en el proceso, según el algoritmo que posee. Por otra parte, un DDC realiza un muestreo del proceso y a partir de la ley de control modifica un actuador local del proceso. Este tipo de controlador se basa en la utilización de un computador o procesador, en donde se convierten las señales análogas a digitales, si es necesario para aplicar dicha ley.

Estos controladores poseen una tasa de muestreo desde 0,2 muestreos/seg. hasta 10 muestreos/seg. [10]. Además, tiene una resolución típica entre 12-14 bits para entradas análogas y entre 10-12 bits para salidas análogas [10], y su operación puede ser programada a través de diagramas de bloques funcionales (FBD), diagrama de escaleras o lenguaje ladder (ladder diagram), o mediante lenguajes de mayor nivel como Pascal, C, entre otros.

Otro tipo de controlador usualmente utilizado es el PLC (Programmable Logic Controller), que es una sólida unidad basada en computadores que realiza control continuo y discreto en una gran variedades de procesos [13]. El PLC nace en 1968 a partir de la necesidad de reemplazar los poco fiables sistemas con relés electromecánicos y eliminar la costosa sustitución de los relés en las líneas de montaje cuando se realiza los cambios de modelos de automóviles para General Motors. Debido a que estos componentes fueron diseñados para la manufactura, y además para reemplazar líneas de relés, las primeras versiones de este controlador se basaron en sistemas de lógica combinacional. Sin embargo, con el tiempo estos sistemas se fueron adaptando a sistemas de control continuo, como por ejemplo, la implementación controladores de tipo PID, además de poseer sistemas de comunicación para la integración a sistemas de control en red, sistemas de monitoreo, entre otros.

Dependiendo de la aplicación del sistema de control, la tarea de supervisar y controlar el sistema puede ser dividida en una serie de controladores, los que en conjunto manejan el proceso. Uno de estos tipos corresponde al sistema de control distribuido (DCS), donde un grupo de controladores y programas permiten manipular ciento o miles de lazos de control. Un sistema comercial DCS (ver Figura 2.1) tiene cientos de controladores conectados bajo una red de alta velocidad (data highway), los que actúan de forma independiente pero supervisados bajo un controlador principal. Además, este sistema tiene acoplado una red de monitoreo y un sistema de registro utilizando una base de datos. La utilización de controladores locales permiten a este sistema realizar cálculos de rápida velocidad, sin embargo, son afectados por el controlador principal, el cual puede considerar relaciones entre lazos de control o tener algoritmos más complejos y de lenta dinámica que analicen, por ejemplo la eficiencia, la

productividad, cambios de especificaciones de los productos, entre otros factores.

En general, un DCS incluye los siguientes componentes [10]:

1. **Red de control**: La red de control es el enlace entre los distintos componentes del sistema de control. Se suele utilizar cable coaxial o fibra óptica, con doble cableado para mejorar la redundancia y con ello reducir la posibilidad de fallas en la comunicación.
2. **Workstations**: Son poderosos computadores en el sistema capaz de realizar cálculos más complejos que otros componentes, como optimizaciones en tiempo real, control predictivo, entre otros.
3. **Unidades de control remoto (RTUs)**: Estos componentes tienen por función recibir información desde multiplexores (Mux), desde sensores y hacia actuadores. Además, tienen la función de implementar algoritmos básicos de control como PID, así también, actúan en modo supervisor manipulando el setpoint y parámetros de controladores, como PLCs.
4. **Estaciones de aplicaciones**: Son computadores donde se ejecutan aplicaciones como bases de datos, hojas de cálculos, software financieros, simulaciones del proceso, o interfaces humano-máquina (en caso de ser una estación de operario).

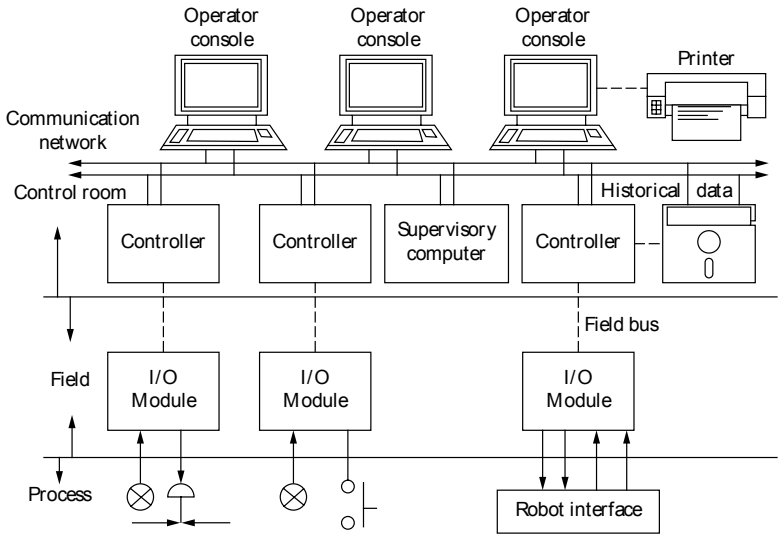


Figura 2.1: Esquema de un sistema DCS genérico [13].

Por otro lado, el sistema SCADA (Supervisory Control and Data Acquisition), corresponde a un sistema que combina la utilización de bases de datos y software para el interfaz con el usuario, HMI (Human Machine Interface), entre otros componentes que permiten monitorear y controlar de forma remota un sistema de control. Usualmente, estos sistemas se combinan con los sistemas de seguridad para mostrar y registrar alarmas, otorgando seguridad a la operación de la planta, por ejemplo, permite alejar los operarios a cargo del proceso a un radio seguro.

2.2. Dispositivos utilizados

2.2.1. Arduino

Arduino es una plataforma de desarrollo de hardware y software libre, enfocado en el fácil uso de éstos. Posee un entorno de desarrollo (IDE), que permite programar y compilar el Arduino a través de un lenguaje de programación Arduino (basado en C++). Esta plataforma posee una gran variedad de modelos, los cuales se diferencian por el tipo de micro-controlador, cantidad de puertos y el tamaño de la placa.

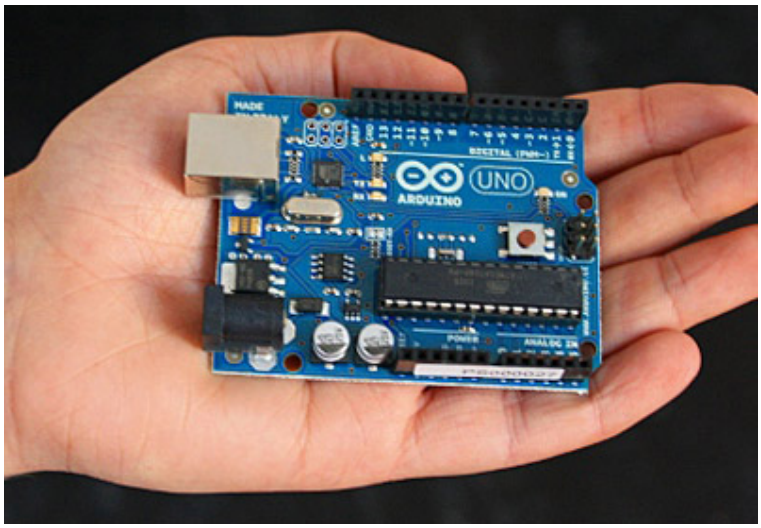


Figura 2.2: Arduino Uno R3.

Una placa Arduino está compuesta por lo menos por un micro-controlador, puertos de entrada y salida (análogos y digitales) y un puerto USB para ser programado. Las entradas análogas permiten medir un voltaje entre 0 y 5 V. con una resolución de 10 bits. Por otro lado, existen bibliotecas que permiten emular una salida discreta a través de PWM, medir frecuencia de una señal de entrada, entre otras funciones.

Además, algunos modelos de esta placa traen mejoras, como por ejemplo micro-controladores de mayor velocidad, mayor cantidad de puertos (hasta 54 puertos I/O), conexiones para Ethernet, WiFi, Bluetooth, tarjetas micro-SD, entre otras características. Debido a su fácil programación y bajo coste, este tipo de dispositivos posee una gran cantidad de usuarios que los ocupan en una gran diversidad de usos, como la creación de impresoras 3D, robots, sensores e instrumentos de bajo costo.

En esta memoria se utilizaron 2 versiones: Arduino Mega 2560 y Arduino Yún (Ver descripción en Anexos F). El primer modelo utiliza un micro-controlador ATmega2560 y posee 56 I/O digitales, y 14 entradas análogas. Por otro lado, el Arduino Yún posee un micro-controlador ATmega328, con rendimiento similar al ATmega2560, 20 I/O digitales y 12 entradas análogas. Sin embargo, la características principal del Arduino Yún es que incluye

un procesador, que permite ejecutar un sistema operativo y con ello, programar algoritmos en otros lenguajes y de mayor complejidad.

2.2.2. Raspberry Pi

El Raspberry Pi modelo B+ es un ordenador de placa reducida (SBC) que posee un procesador de tipo ARMv6, 512 MB de memoria RAM, 4 puertos USB, procesador gráfico-GPU (capaz de reproducir vídeos en 1080p), conexión vía Ethernet, adaptador de memoria micro-SD y un consumo medio de 3W. Este SBC permite ejecutar varios sistemas operativos como Debian, Fedora, ArchLinux, entre otros.

A pesar del pequeño tamaño (similar a una tarjeta de crédito), posee la capacidad levemente menor a un computador de escritorio, permitiendo la ejecución de software como base de datos, servidor web y aplicaciones livianas de escritorio, como por ejemplo, hojas de cálculo, edición de documentos, etc.

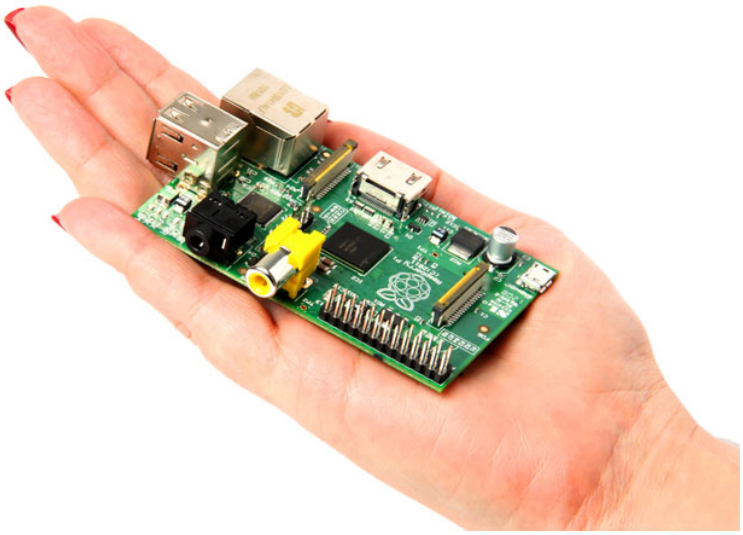


Figura 2.3: Raspberry Pi modelo B.

2.3. Descripción HL630

El banco de ensayos HL630 corresponde a sistema destinado a la enseñanza, el cual posee un circuito de calefacción compuesto por un calefactor eléctrico, estanque de expansión, bomba de circulación, válvula de gobierno termostático y un radiador con sistema de ventilación para la extracción de calor.

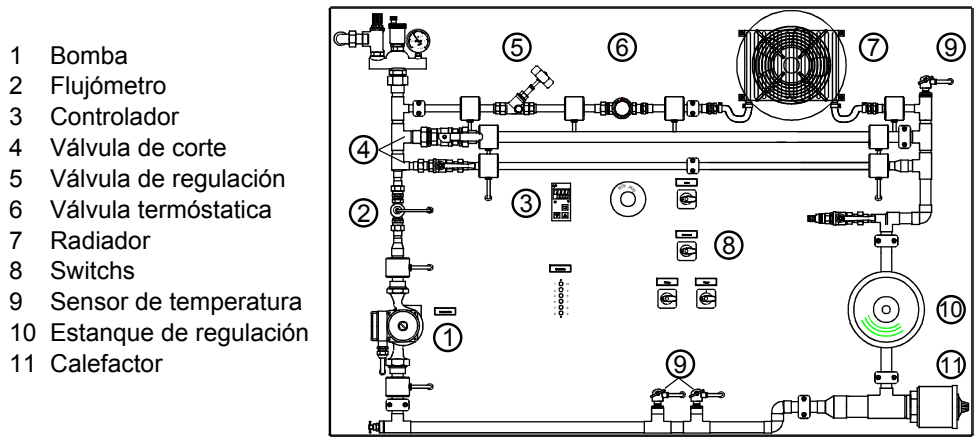


Figura 2.4: Esquema de componentes del equipo HL630.

Este equipo está diseñado para la medición de pérdidas de cargas de distintas singularidades en función de la temperatura del fluido de trabajo. Posee una tarjeta de adquisición de datos mediante la cual se puede medir variables desde los sensores, obtener las mediciones, controlar el funcionamiento del calefactor, ventilador y la bomba desde un PC. El control de la temperatura se realiza mediante un controlador de 2-puntos que posee el equipo.

El modo de funcionamiento del equipo se elige a través de un switch, el que está compuesto por 3 formas: modo automático controlado por PC (1), modo de regulador industrial (2) y modo manual (3). El modo automático controlado por PC permite que los actuadores del sistema sea manipulado a través de software del PC, el que muestra a través de una interfaz gráfica los valores medidos en tiempo real, así como también encender o apagar el calefactor y el ventilador, elegir la potencia de la bomba y controlar la temperatura del sistema mediante una emulación del controlador de 2-puntos en el software del PC. El modo de regulador industrial utiliza el controlador de 2-puntos KS40, mediante el cual el encendido y apagado del calefactor solo es manipulado por este controlador, este componente será descrito en mayor profundidad en la Sección 2.3.1. En cambio, el modo manual permite que la manipulación de los actuadores se realice de forma manual a través de distintos switchs que posee el tablero del equipo.

2.3.1. Descripción de componentes

Sensor de temperatura

El sensor de temperatura corresponde a un sensor del tipo PT100, que basa su medición en la variación de la resistencia de un conductor debido a la temperatura. El equipo HL630 posee 3 sensores, uno ubicado a la salida del calefactor el que está conectado a la tarjeta de adquisición de datos, otro también ubicado a la salida del calefactor pero que está conectado

con el controlador KS40 y el último a la salida del radiador, también conectado a la tarjeta de adquisición de datos.

Los tres sensores son del mismo tipo y tienen la configuración de 3 hilos, lo que permite eliminar el efecto de la temperatura entre la resistencia de platino y el dispositivo que mide la resistencia (por ejemplo, un transmisor).

Transmisor de temperatura

Un transmisor es un transductor que responde a una variable medible a través de un sensor y lo convierte a una señal estandarizada que solo es función de esta variable medible [13]. En este caso en particular, el transmisor de temperatura que se encuentra en el equipo corresponde a la marca Pförtner modelo 1311. Este dispositivo se conecta al sensor de temperatura y entrega un voltaje entre 0 y 10V, el cual tiene un comportamiento lineal con respecto a la temperatura a la cual se mide. Las características técnicas se observan en la Tabla 2.1.

Tabla 2.1: Características del transmisor de temperatura Pförtner 1311 [16].

	Descripción
Salida	0...10V
Carga máxima	900 Ω
Influencia de carga	0,02 %/100Ω
Límite de corriente	35 mA
Tiempo de respuesta	$< 1s$
Señal de error	$I_{out} > 23$ mA
Entrada	PT100 2 o 3 cables
Presión	0,2 %
Rango	0...100°C
Alimentación	230V AC

Controlador

En el modo de regulador industrial, la manipulación del calefactor es producida por el controlador KS40 [17], producto orientado a aplicaciones térmicas. El equipo posee instalada la versión 406, la cual posee 2 relés (una para calentar y otro para enfriar), sin embargo sólo se encuentra conectado el calefactor al equipo. Esta versión implementa un controlador de 2-puntos cuando solo realiza calentamiento, y cuando calienta o enfría posee un controlador de 3-puntos.

El controlador posee un sistema de alarma (una alarma relativa con respecto al setpoint y otra absoluta) y además permite almacenar un segundo setpoint en caso de falla del controlador.

La señal de entrada proviene de la medición directa de un sensor de tipo PT100 (3 hilos) o termocuplas tipo L,J. No posee un interfaz digital, por lo tanto no se puede conectar a un sistema de control debido a que su manipulación se realiza de forma directa a través de los botones que posee el controlador. Las propiedades técnicas se observan en la Tabla 2.2.

Tabla 2.2: Características técnicas del Controlador PM KS40 [17].

	Descripción
Rango de medición	0..99,9°C o 0...400°C
Conexión	PT100 3 hilos
Corriente del sensor	$\leq$ 1,5 mA
Controlador	2 relés
Tipo de relé	NO SPST
Carácteristica de línea	Máx. 500VA, 250V,3A; 48...62Hz
Alimentación	230V AC/115V AC; 48...62Hz
Tipos de controlador	Controlador 2-puntos y 3-puntos
Sintonización	Manual o automática
Indicador alarma	LED
Modo de funcionamiento	Calentamiento/Enfriamiento/Ambas

Flujómetro

El sensor de flujo que posee el equipo es del tipo miniturbina, de marca Kobold modelo DPL250K, el cual posee un sensor óptico que detecta la rotación del álabe del sensor.

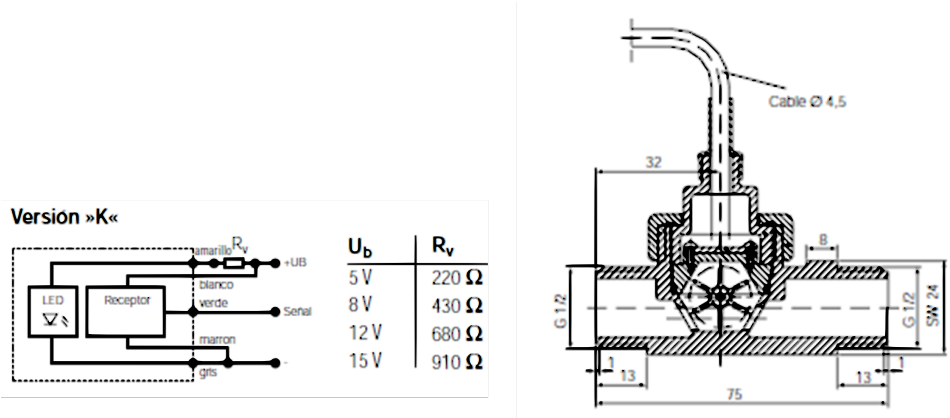


Figura 2.5: Izquierda: Circuito necesario para que la construcción de la señal de salida del sensor. Derecha: Vista frontal del sensor [12].

Las características técnicas del sensor de flujo se muestran en la Tabla 2.3.

Tabla 2.3: Características técnicas del sensor de flujo Kobold DPL250K [12].

	Descripción
Rango de medición	1 a 25 L/min de agua
Linealidad	±1 %
Presión máxima	10 bar
Temperatura máxima	70°C
Rango de viscosidad	Baja viscosidad
Salida	Pulsos
Sensor	Transmisitividad de luz infrarroja.

Bomba

El equipo cuenta con una bomba Grundfos UPS 25-60, la cual está destinada al uso domiciliario para sistemas de calefacción y climatización. Posee 3 niveles de potencia (50, 60 y 70 W), y además un selector de intensidad máxima.

Calefactor

La fuente de calor del sistema corresponde un calefactor eléctrico, que tiene una potencia máxima de 2000 W, y como sistema de seguridad posee un termostato que permite el funcionamiento de este componente hasta una temperatura máxima que puede variar entre 30 y 80°C.

Radiador

El sumidero de calor del sistema corresponde a un radiador de aluminio el cual posee un ventilador que permite mejorar la transferencia de calor y con ello aumentar la potencia de extracción de calor.

Capítulo 3

Metodologías

La siguiente sección tiene por objetivo mostrar los distintos procedimientos experimentales realizados en el equipo HL630 utilizados para la identificación de un modelo, necesario para un acercamiento para la sintonización de los parámetros del controlador, y la sintonización propiamente tal. Además, en esta sección se muestra la utilización de 2 softwares pertenecientes a Matlab para las tareas recién nombradas.

3.1. Metodología experimental

3.1.1. Procedimientos de operación de equipo HL630

Esta sección entrega información básica para la operación del equipo HL630, como la puesta en marcha del equipo y el apagado de este.

Procedimiento de encendido de equipo:

1. Poner el sistema en modo manual.
2. Verificar que el calefactor, la bomba y el ventilador estén en modo apagado.
3. Encender el interruptor principal.
4. Verificar que la presión del sistema este dentro del rango de operación (1,5 a 2,5 bar).
5. Cerrar las válvulas de cierre de tramo deseadas.
6. Encender bomba en la posición deseada.

Procedimiento de disminución de presión del sistema:

1. Verificar que el interruptor principal se encuentra desconectado.
2. Abrir válvulas de cierre de tramo.
3. Abrir válvula de alivio hasta alcanzar el rango de presión de operación.

Procedimiento de aumento de presión del sistema:

1. Verificar que el interruptor principal se encuentra desconectado.
2. Abrir válvulas de cierre de tramo.
3. Conectar el equipo a la red de abastecimiento de agua.
4. Abrir válvula de alimentación hasta alcanzar 2,5 bar (presión máxima de operación).
5. Cerrar válvula de alimentación y abrir válvula de alivio con el fin de eliminar la presencia de burbujas en el sistema.
6. Repetir pasos 4 y 5 hasta que las burbujas sean eliminadas.
7. Abrir válvula de alimentación hasta alcanzar el rango de presión de operación.
8. Desconectar el equipo a la red de abastecimiento.

Procedimiento de apagado del sistema:

1. Seleccionar la operación del equipo en modo manual.
2. Apagar el calefactor y encender el ventilador.
3. Seleccionar la potencia de la bomba en nivel 3.
4. Abrir la válvula de regulación y cerrar las válvulas de cierre de tramo medio e inferior.
5. Esperar que el sistema alcance un estado cercano al estacionario.
6. Apagar el ventilador y la bomba.
7. Seleccionar el interruptor principal en posición de apagado.

3.1.2. Pruebas de reconocimiento

Esta prueba tiene como objetivo conocer cualitativamente la respuesta del sistema en lazo abierto a partir de pulso en el calefactor y en el ventilador. La siguiente lista muestra el procedimiento utilizado para esta prueba.

1. Encender el equipo utilizando el procedimiento de encendido.
2. Cerrar las válvulas de cierre tramo medio e inferior, para permitir el paso del agua solo por el radiador.
3. Seleccionar potencia de la bomba en nivel 3.
4. Encender el calefactor durante 30 segundos.
5. Esperar que el sistema alcance un estado cercano al estado estacionario.
6. Repetir pasos 3 al 5 utilizando potencia de bomba en nivel 2 y 1.
7. Seleccionar potencia de la bomba en nivel 3 y encender el ventilador hasta que el sistema alcance un estado cercano al estado estacionario.
8. Abrir válvula de tramo medio y cerrar válvula de regulación (válvula del tramo superior) y válvula de tramo inferior.
9. Repetir los pasos 4, 5 y 6.
10. Apagar el equipo utilizando el procedimiento adecuado.

3.1.3. Pruebas de sintonización y sensibilidad

Las pruebas de sintonización y de sensibilidad tienen por objetivo analizar el efecto en el sistema en lazo cerrado debido a los parámetros del controlador. Las pruebas de sintonización corresponden a las pruebas donde se modifican parámetros (K_C, τ_I y τ_D) desde el interfaz humano-máquina (HMI), en cambio las pruebas de sensibilidad corresponden a las pruebas donde es necesario modificar los parámetros (P_{PWM} y I_{max}) desde el código de fuente del controlador. Los siguientes pasos muestran el procedimiento hecho para estas pruebas:

Pruebas de sintonización:

1. Encender el equipo utilizando el procedimiento de encendido.
2. Cerrar las válvulas de cierre tramo medio e inferior, para permitir el paso del agua solo por el radiador.
3. Seleccionar potencia de la bomba en nivel 3.
4. Seleccionar el equipo en modo controlador externo (ex modo automático controlado por PC).
5. En el interfaz humano-máquina (HMI) seleccionar el controlador en modo *auto* e ingresar los parámetros.
6. Seleccionar el sensor de temperatura inferior y fijar el setpoint en 35°C, y esperar que el sistema alcance un estado cercano al estado estacionario.
7. Repetir el paso anterior, fijando como setpoint 55°C, y esperar un estado similar al estado estacionario.
8. Apagar el equipo utilizando el procedimiento adecuado.

Pruebas de sensibilidad:

1. Utilizar un cliente SSH para conectar al controlador mediante su dirección IP, usuario y contraseña.
2. Mediante un editor de texto, abrir el documento *pid_controller.py* y modificar la variable:
   - *self._max_e*: para fijar el valor de I_{max}.
   - *self._dutytime*: para fijar el valor de P_{PWM}.
3. Guardar la modificación y reiniciar el controlador.
4. Realizar el procedimiento de pruebas de sintonización.

3.2. Herramientas utilizadas

3.2.1. System Identification Toolbox

Durante el desarrollo de esta memoria, fue necesario realizar procedimientos para determinar un modelo empírico del sistema. Para ello, se requirió realizar un ajuste de parámetros de sistemas lineales de distinto orden, lo que fue realizado mediante la herramienta System Identification Toolbox de Matlab.

System Identification Toolbox es una aplicación de Matlab que permite construir modelos matemáticos de un sistema dinámico a través de mediciones de variables de entrada y salida del sistema. Esta herramienta permite elaborar modelos a partir de funciones de transferencia, modelo de estado de sistemas, modelos polinomiales, así como también modelos no-lineales. Esta aplicación posee una interfaz gráfica que facilita su utilización (ver Figura 3.1).

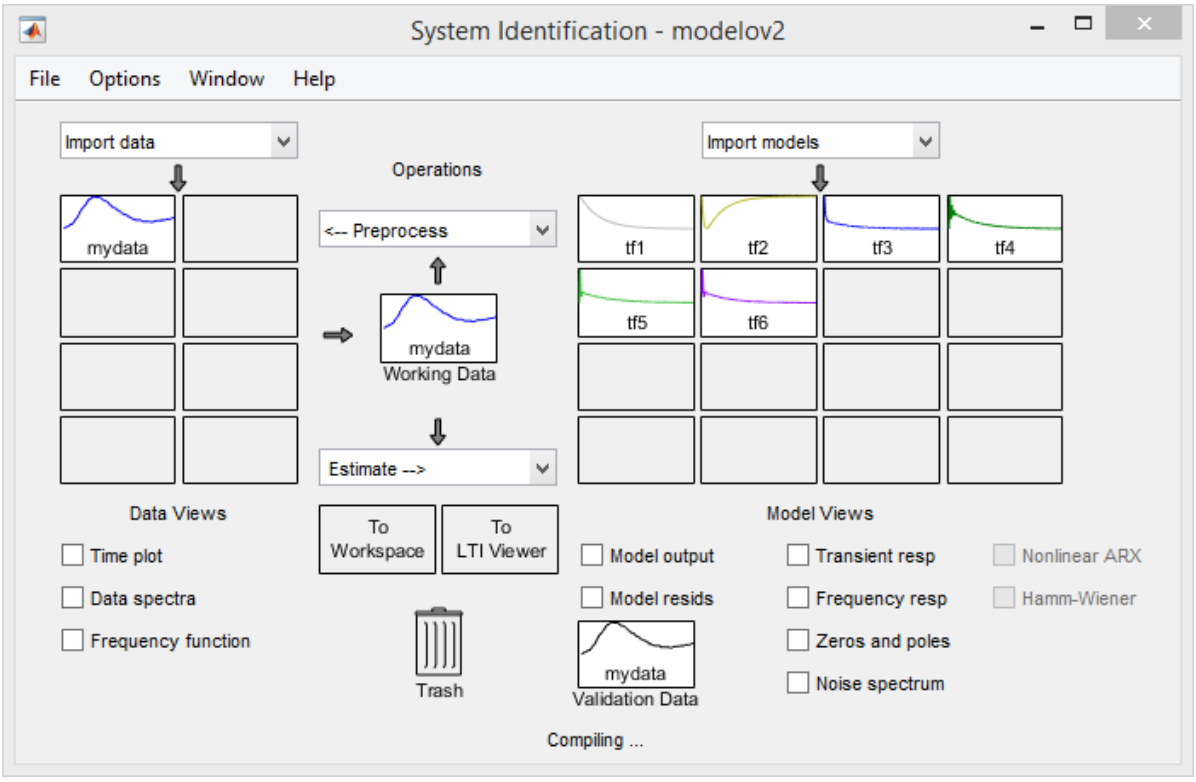


Figura 3.1: Interfaz gráfica de System Identification Toolbox.

En este caso particular, se utilizó la utilización de identificación del modelo del proceso a partir de funciones de transferencias, tomando como datos experimentales:

- **Entrada**: Pulso de 15 seg. de duración y amplitud 2000 W.
- **Salida**: Medición de temperatura, en respuesta al pulso, en el sensor inferior obtenido desde la base de datos vía HMI.

Luego, sin pre-procesar los datos, se seleccionó estimador modelo basado en función de transferencia y se utilizó las siguientes ordenes:

- Orden 1
- Orden 2
- Orden-(2,1)
- Orden 3
- Orden-(3,1)
- Orden-(3,2)

Con esto, al estimar un modelo se entrega los valores de los parámetros obtenidos con una significancia del 95 % al minimizar la diferencia entre la simulación del modelo estimado frente a los datos experimentales. Con esto, se muestran el interfaz del System Identification Toolbox, que permite observar los parámetros estimados, su intervalo de confianza, diagrama de bode del modelo estimado, gráficas entre una simulación del modelo y los datos experimentales, el valor de estadísticos como FPE, MSE y NRMSE (Ver Figura 3.2).

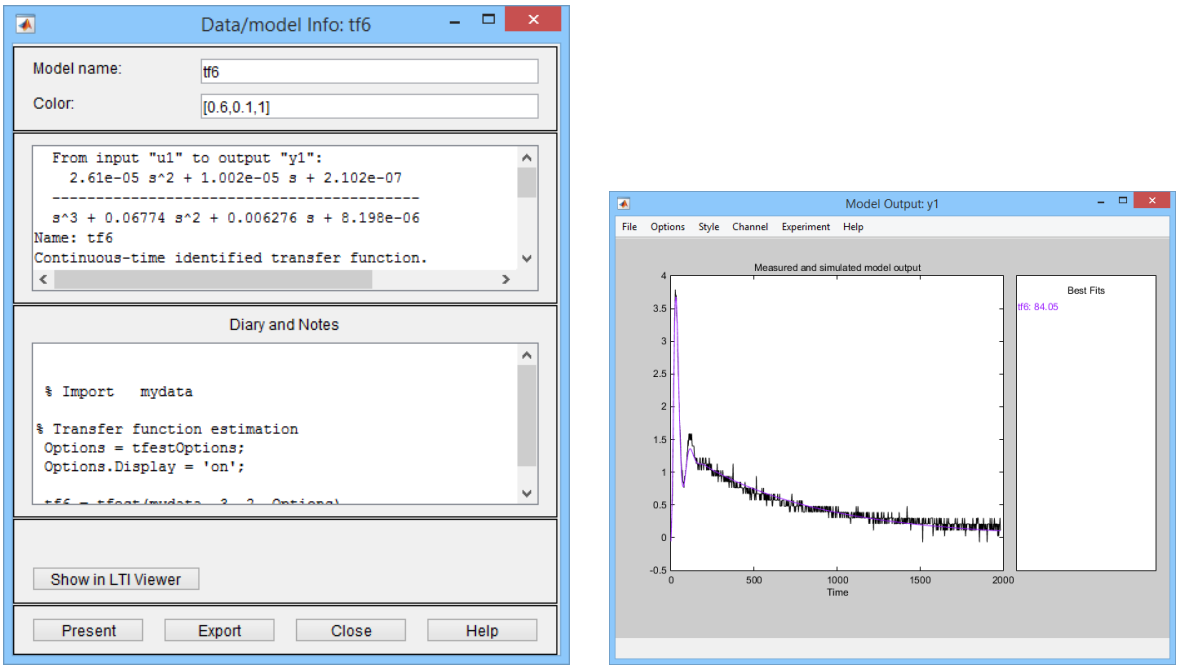


Figura 3.2: Interfaz gráfica entregada por System Identification Toolbox mostrando el modelo estimado y una gráfica comparativa.

3.2.2. PID Tuner

Para obtener los primeros parámetros del controlador, se realizó en Simulink (de Matlab), un modelo continuo del sistema en lazo cerrado. Esta herramienta gráfica (Ver Figura 3.3) basa su funcionamiento en simulaciones de la respuesta del sistema en lazo cerrado

minimizando el overshoot para cambios escalones en la referencia y en la perturbación (en caso de existir), entregando los parámetros del controlador, y 2 coeficientes: robustez del sistema (0 a 0,9) y tiempo de estabilización.

Para el sistema, se fijó un tiempo de estabilización de 1845 seg., ya que se consideró un buen tiempo para que el proceso alcance un estado cercano al estacionario. Además, se especificó una robustez de 0,6, ya que el overshoot no presenta una gran amplitud, valores mostrados en la sección de resultados.

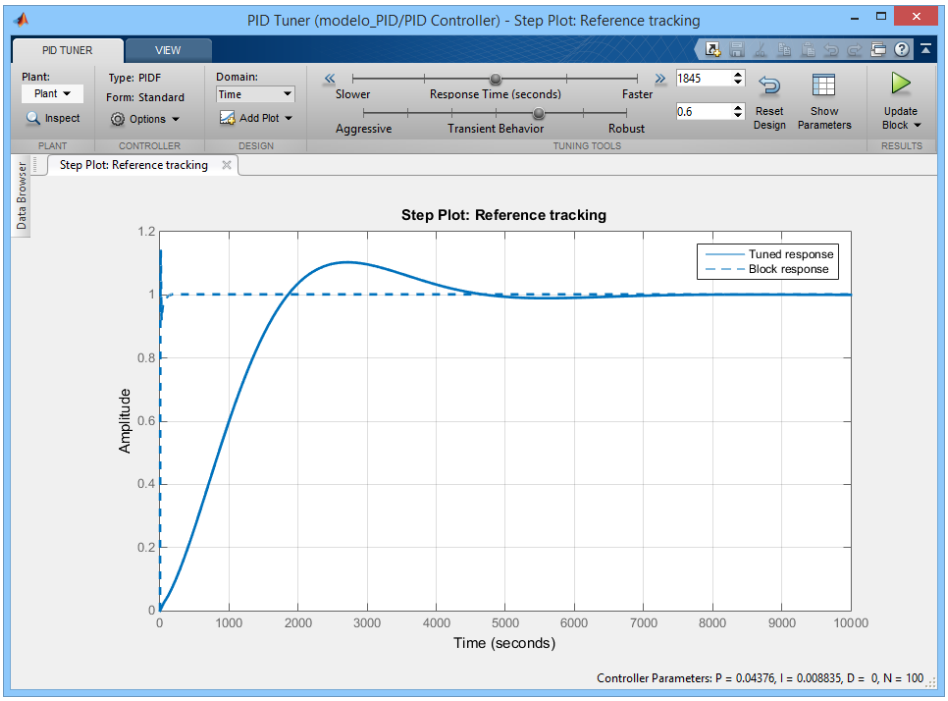


Figura 3.3: Interfaz gráfica de PID Tuner utilizada para la estimación de parámetros de un controlador PID Continuo.

Capítulo 4

Resultados

4.1. Diseño del sistema de control

4.1.1. Descripción del sistema

El sistema de control que se describe en este documento está inspirado en un sistema de control distribuido, debido a que posee componentes que interactúan de forma directa y descentralizada en el proceso, ya sea midiendo o controlando. Estos componentes poseen un radio de acción acotado correspondiente a una sección del proceso, cubriéndolo con una red de dichos componentes lo que permite mejorar el desempeño de los controladores, así como también, aislar el área afectada en caso de falla del sistema. Para este sistema, estos componentes son denominados como elementos de control.

Cada Elemento de Control funciona de forma autónoma con respecto a otro de su mismo tipo, sin embargo, también interactúa con otro componente de mayor nivel, denominado Servidor de Control (Ver Figura 4.1). Este servidor tiene por función transmitir los datos que fueron medidos desde el Elemento de Control y modificarlos mediante el cambio de parámetros en el controlador o modificar el estado de los actuadores de forma directa. Esta acción es recogida desde el o los interfaces usuario/máquina (HMI) que estén conectados al sistema de control. Estos HMI poseen un interfaz gráfica que muestran mediciones e interactúan con el usuario para modificar el funcionamiento del equipo. Para esto, el HMI debe estar conectado al sistema de control a través del Servidor de Control.

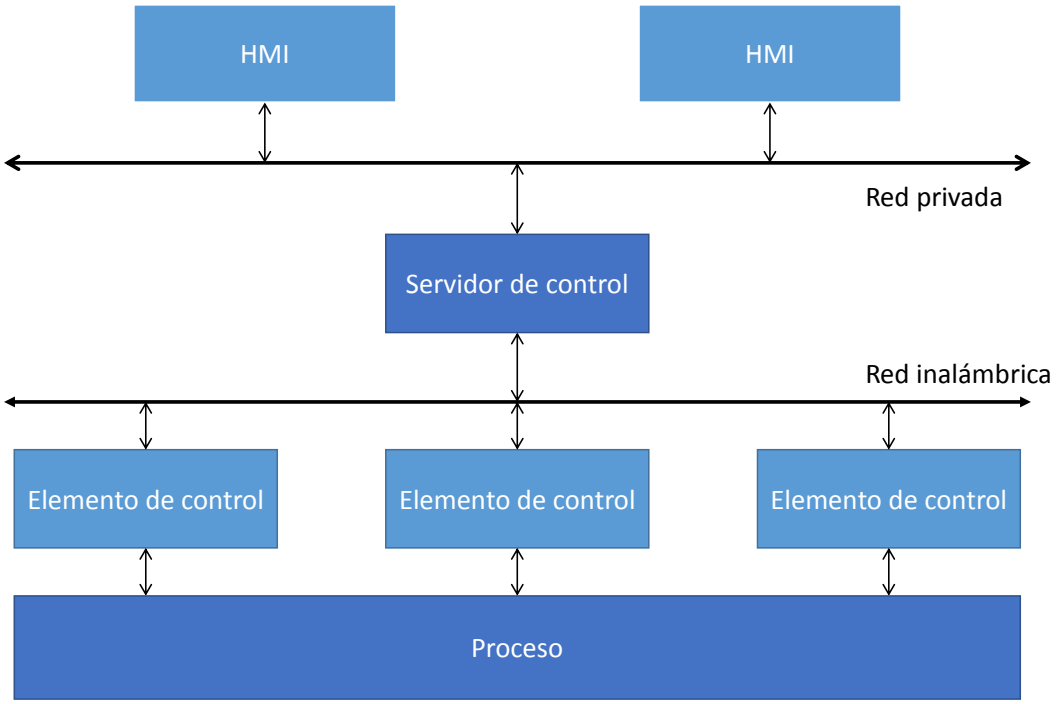


Figura 4.1: Representación del sistema de control.

La comunicación entre el Elemento de Control con el Servidor de Control, que incluye un servidor web y la base de datos, se desarrolló en dos etapas distintas, de forma evolutiva, en donde la versión actual corrige los errores de la versión anterior.

La primera versión (ver Figura 4.2), posee como característica que la comunicación desde el Elemento de Control hacia el servidor se realizó a través de un interfaz que recibía datos desde el Elemento de Control y se enviaban hacia la base de datos. Este interfaz se creó debido a que no se conocía una biblioteca que permitiera comunicar directamente el Arduino con la base de datos. Además, para otorgar más seguridad a la comunicación se utilizó una encriptación en ARC4 en el Arduino, y en el Interfaz ocurría la desencriptación.

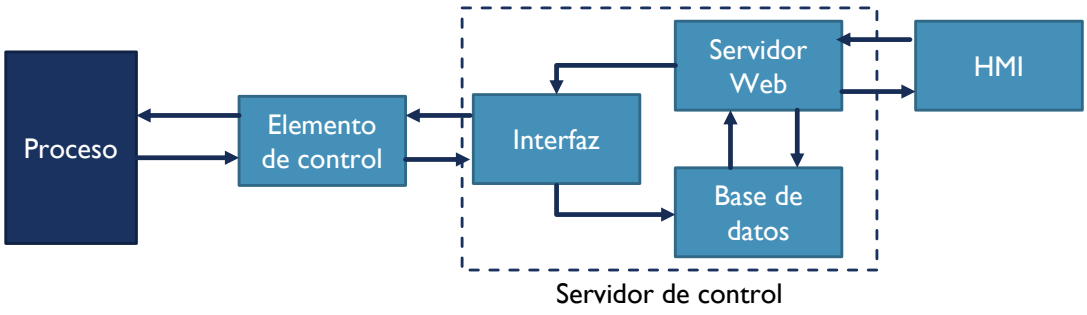


Figura 4.2: Primera versión del sistema de comunicación entre el usuario/operador y el proceso para un sistema de control con un único Elemento de Control.

Para que un valor apareciera en el HMI, el servidor web se encargaba de mostrar e interactuar con el usuario, consultando a la base de datos sobre el último valor medido que fue registrado. Por otro lado, el Elemento de Control cada cierto intervalo medía desde las entradas conectadas al proceso, encriptaba un mensaje que contiene el componente y su valor medido, y lo enviaba al interfaz ubicado en el Servidor de Control. Este interfaz desencriptaba el mensaje, y utilizando una biblioteca que lo conectaba con la base de datos (Connector/J), se creaba el mensaje en SQL Syntax para insertar el valor medido.

Para realizar una modificación en el proceso desde el HMI, por ejemplo cambiar la potencia de una bomba, el servidor web enviaba hacia el Interfaz un comando. El Interfaz encriptaba un mensaje y luego enviaba este comando al Elemento de Control. En el Elemento de Control, se descriptaba el mensaje y se realizaba la acción mediante el uso de los puertos de salida para afectar el proceso.

Además, para comunicar el Servidor de Control con el Elemento de Control se utilizó el mismo interfaz, donde el proceso de encriptación ocurría en el Interfaz y la desencriptación sucedía en el Elemento de Control. Este interfaz se desarrolló utilizando JAVA, y poseía la característica de poder establecer una comunicación múltiple utilizando multitasking, además, la comunicación con el Elemento de Control fue a través de la configuración cliente-servidor. Durante la implementación de esta versión del sistema, se utilizó como Elemento de Control el Arduino Mega 2560 y en la Figura 4.3 se observa el hilo de ejecución que para realizar en el ciclo de monitoreo y control.

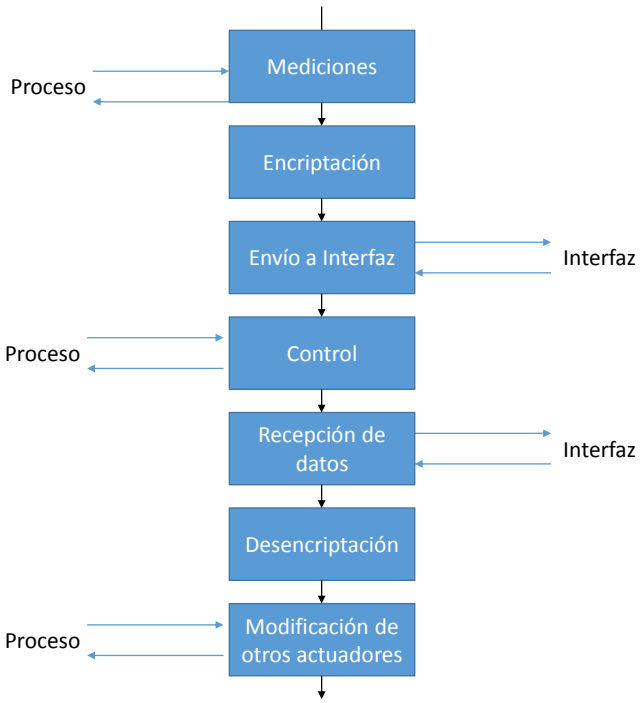


Figura 4.3: Hilo de ejecución del Elemento de Control utilizando Arduino Mega 2560 y el Interfaz en el Servidor de Control.

En esta implementación, la duración del ciclo de control para este sistema fue alrededor

de 15 seg, y se observó que el cuello de botella en la ejecución de este ciclo correspondía a la comunicación entre el Elemento de Control y el interfaz. Cabe destacar que la implementación de este interfaz demostró poca eficacia, ya que exhibió problemas en el algoritmo de encriptación para ciertos datos, y además, la comunicación que realizaba era inestable.

Para mejorar la estabilidad y disminuir el tiempo en cada ciclo de control, se modificó el sistema de comunicación entre los componentes. Se encontró una biblioteca, en estado de desarrollo, que permitió la comunicación directa en el Arduino y la base de control. Con ello, ya no fue necesario la utilización de Interfaz debido a que el Elemento de Control pudo insertar directamente registros a la base de datos, y todas las modificaciones hechas desde el usuario se enviaban a otra tabla donde el Elemento de Control consultaba si hay cambios y con ellos se producía la modificación. El diagrama que describe esta nueva versión se aprecia en la Figura 4.4.

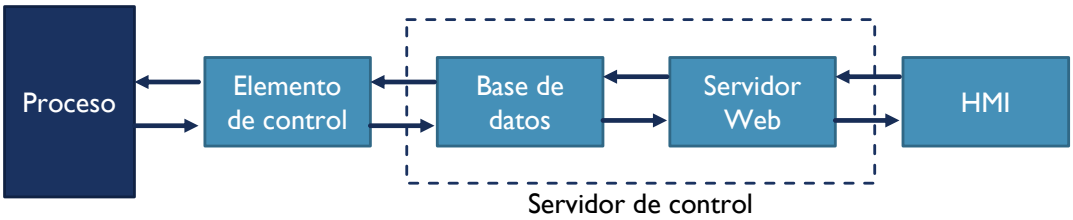


Figura 4.4: Segunda versión del sistema de comunicación entre el usuario/operador y el proceso para sistema de control con un único Elemento de Control.

Como se mencionó anteriormente, esta versión mejoró el tiempo de ciclo de control, sin embargo, fue un lapso de magnitud similar (8 a 10 segundos), siendo lento para el sistema en el que fue implementado. Sin embargo, hay que mencionar que la conexión entre el elemento y el Servidor de Control fue más segura, ya que la comunicación entre el Arduino y la base de datos fue encriptada con un algoritmo más robusto (SHA256) y los datos se fueron transferidos mediante una capa de seguridad (TLS/SSL).

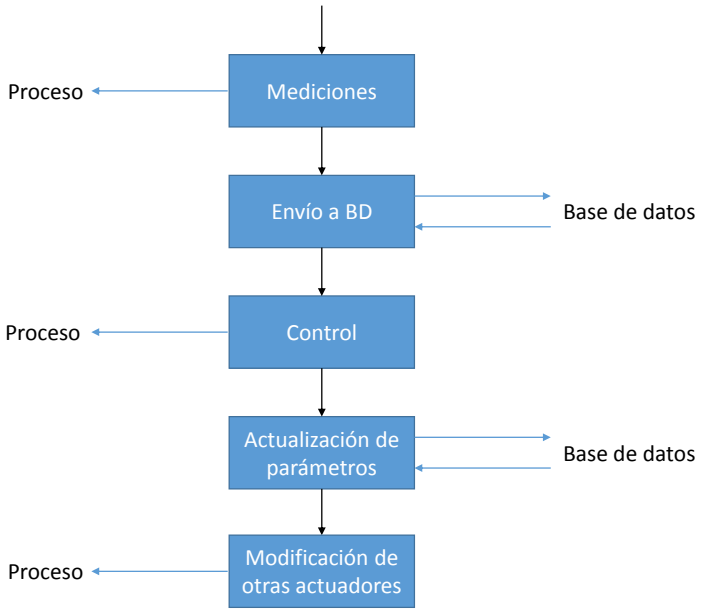


Figura 4.5: Hilo de ejecución del Elemento de Control utilizando conexión directa con la base de datos.

Otro aspecto importante es que la ejecución de comandos en el Arduino Mega 2560 se realizó en un solo hilo de ejecución, por lo tanto, alguna demora en un comando del script producía un retardo en el ciclo de control y muestreo, lo que se ve reflejado en el tiempo recién mostrado.

Debido a lo anterior, se cambió el modelo del Arduino Mega 2560 al Arduino Yún, el cual posee un microcontrolador con capacidad de procesamiento similar al Arduino Mega 2560, pero además este nuevo dispositivo trae un procesador que permite ejecutar un sistema operativo, lo que le otorga mayor versatilidad, utilización de multithead, entre otras características mencionadas en la Sección 2.2.1. A pesar del cambio, la estructura del sistema de control no cambió, y solo se modificó el funcionamiento del Elemento de Control y como éste interactuó con los demás componentes del sistema de control.

4.1.2. Elemento de Control

El Elemento de Control tiene por función registrar las mediciones de los sensores y modificar el proceso a través del envío de señales hacia los actuadores del equipo que se encuentran conectados con éste. Para realizar esto, el Elemento requiere estar conectado directamente con el controlador de un lazo de control (modo supervisor) o actuar como tal al estar conectado directamente con los actuadores.

Para que el Elemento de Control actúe como un controlador, en primer lugar, debe estar

conectado al sensor y al actuador que forma el lazo de control, y en el Elemento de Control debe estar programada una ley de control que permita realizar los cálculos y controlar el proceso a través de ese lazo. En el caso contrario, donde el Elemento de Control funciona en modo supervisor, este manipula un controlador ya instalado en un lazo de control, y para ello, este componente debe poseer una comunicación con dicho controlador de manera que pueda modificar el valor de referencia (setpoint) y los parámetros de éste posee.

Este componente puede ser un microcontrolador o procesador que contenga un procesador digital de señales (DSP). Las características que debe poseer este equipo son las siguientes:

- Capacidad de comunicarse con el proceso, intrínseca en el componente, ya que el Elemento de Control requiere saber los valores medidos de los sensores para enviarlos al Servidor de Control y registrarlos en la base de datos.
- Capacidad para conectarse con redes inalámbricas, esta función se eligió porque facilita la instalación del Elemento de Control en el equipo, ya que elimina un cableado en el proceso.
- Ser programable, también es una característica propia del componente, y es necesaria para que se desarrolle el algoritmo de control, donde ciertas funciones a desarrollar dependen del proceso que controla el sistema, ya que existe una gran diversidad de actuadores y sensores en la industria, lo que hace que cada elemento esté adaptado para determinado proceso.

Como se dijo anteriormente, la principal característica de este componente es la comunicación directa con el proceso, ya sea mediante señales análogas, puertos Serial, RS-455, Ethernet, entre otros. Al recibir mensajes o leer señales desde el proceso se obtendrán datos que provienen de los sensores conectados a dicho Elemento de Control. Por otro lado, los datos enviados al proceso van a modificar el actuar de éste, por ejemplo, modificando una variable de entrada o modificando el actuar del controlador.

Dada la gran variedad de formas de comunicación utilizadas en distintos protocolos, la utilidad de utilizar puertos digitales y/o análogos para conectar interfaces que se adapten a los distintos componentes ya sean válvulas, bombas, calefactores, controladores de diversos tipo, entre otros, permiten al Elemento de Control una gran versatilidad y adaptabilidad.

Otra exigencia del Elemento de Control es que requiere conectarse a redes inalámbricas, esta conexión es necesaria para la comunicación con el Servidor de Control, y con ello poder enviar los datos medidos (datos que serán recibidos desde el mismo componente), y además, poder modificar la operación.

Finalmente, el sistema debe ser programable por diversas razones: este componente debe saber que dada la diversidad de formas de comunicación se necesita distinguir las señales recibidas, y con ello poder transformarlas en una medición. Por ejemplo, el flujometro DPL250-K, el cual fue utilizado en la implementación de este sistema, entregaba una señal digital, la que mediante una transformación se obtuvo el flujo volumétrico.

Otra razón es qué hacer con los datos recibidos desde el proceso, por ejemplo, recibir la

señal proveniente de un sensor, identificarla, manipularla y luego enviarla de manera correcta hacia el Servidor de Control para que entienda el mensaje y pueda registrarlo en una base de datos. Lo mismo ocurre en la dirección contraria, por ejemplo, recibir un mensaje para modificar un flujo y luego adaptarlo para enviar realizar la acción, enviando señales análogas o digitales al proceso para que ocurra este cambio.

Inicialmente, se implementó como Elemento de Control el Arduino Mega 2560 más la utilización de la tarjeta de expansión denominada Arduino WiFi Shield (ver Anexos F.1) la cual permitió comunicar el Elemento de Control a través de una red inalámbrica. Como se mencionó anteriormente, la ejecución del ciclo de control presentó problemas debido a que la comunicación con el Interfaz fue lenta e inestable, y para solucionar esto, se utilizó una librería que permitió la comunicación con una base de datos, pero la inserción y consulta a la base de datos produjo un retardo en el ciclo de control, donde se observó que este ciclo tarda alrededor de 10 segundos.

Para solucionar lo anterior, se implementó el Arduino Yún (ver características en Anexos F.2), donde la comunicación con el sistema de control (base de datos y servidor web en el Servidor de Control) se utilizó mediante un módulo que trae incorporado de WiFi. El diagrama del hilo de ejecución de ciclo de monitoreo y control se observa en la Figura 4.6, donde se muestra la relación entre el microcontrolador y el procesador del sistema.

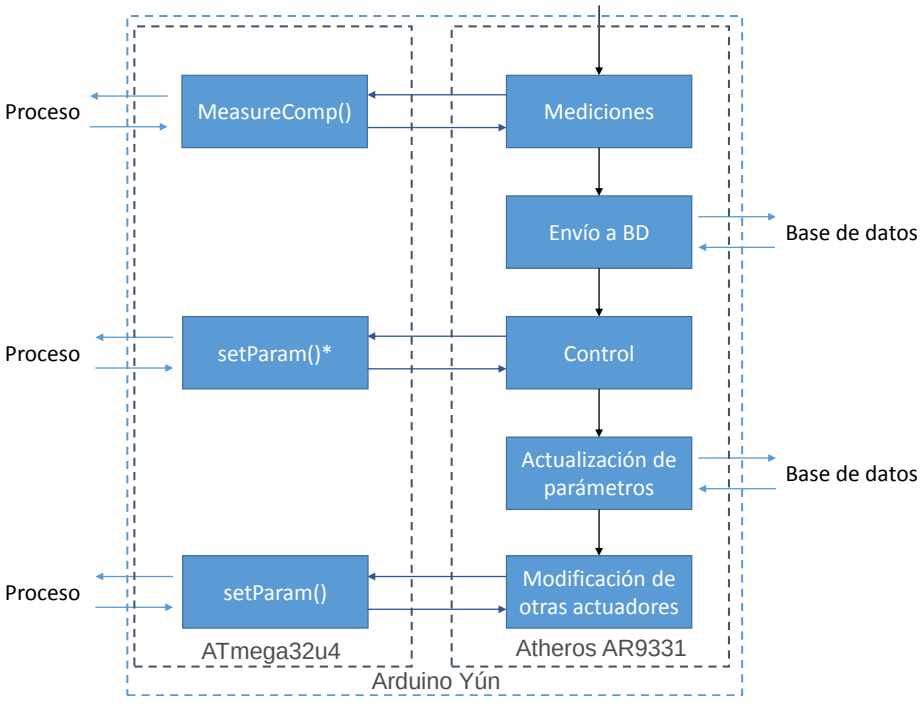


Figura 4.6: Hilo de ejecución del Elemento de Control utilizando Arduino Yún, con conexión directa con la base de datos.

Además, se determinó utilizar el Elemento de Control como controlador digital directo debido a que los parámetros del controlador que posee el equipo HL630 sólo son modificables al pulsar botones físicos que trae en su cubierta, por lo tanto, para permitir la manipulación

remota del lazo de control, el Elemento de Control cerró el lazo y aplicó la ley de control la que corresponde a un controlador PID con un modulador de ancho de pulsos (PMW).

En este caso, el controlador actuó sobre el calefactor (elemento final de control para este lazo) en función de la variable de entrada correspondiente a la medición del sensor de temperatura, ubicado después del calefactor. Además de esta función, el Elemento de Control realizó las funciones propias de éste, las cuales fueron recopilar mediciones de los sensores y enviarlas a la base de datos para su registro, y además obtener, desde la base de datos, el registro de las modificaciones hechas desde el HMI y luego aplicarlas sobre el equipo, donde estos registros fueron variables de entrada para el proceso o también, ser parámetros del controlador, por ejemplo, el encendido del ventilador, o modificar el valor del setpoint, K_c, entre otros.

4.1.3. Servidor de control

Otro dispositivo que compone el sistema, es el Servidor de Control cuya función es poder comunicarse con los distintos elementos de control conectados al sistema, interactuar con ellos para recibir los datos provenientes de las mediciones del proceso y enviar mensajes para la modificación de parámetros y/o variables de entrada.

Este componente está dividido por 3 unidades: un interfaz, un servidor web que aloja el HMI, cuya función es crear la interacción entre el sistema de control y el usuario, el que es mostrado en otras pantallas; y una base de datos, que almacena las mediciones desde el proceso y registra los cambios realizados desde el HMI.

El Interfaz corresponde a un software que corre dentro del Servidor de Control y tiene por función recibir las mediciones del Servidor de Control y almacenarlas a la base de datos, y además recibir información del servidor web y enviarlos al Elemento de Control. Sin embargo, como se mencionó anteriormente este componente del Servidor de Control quedó obsoleto luego de implementar el Elemento de Control al Arduino Yún.

El HMI es una herramienta que permite a los operarios observar y modificar el proceso a distancia, así como también puede generar reportes, analizar y observar registros históricos. Para el sistema de control propuesto, la inclusión de este interfaz gráfico permite observa un diagrama del proceso, el valor de las variables medidas en tiempo real, observar gráficas históricas de los datos almacenados, así como también cambiar el funcionamiento de los equipos del proceso, y modificar parámetros estos.

Para mejorar su adaptabilidad a los distintos sistemas operativos, tamaños de pantalla, entre otros factores, se generó este interfaz basado en una aplicación web, mediante herramientas como JavaScript, JQuery, PHP, entre otros, para que permitieran una fácil interacción con el usuario y un buen manejo con la base de datos.

Además, para almacenar información medida desde el proceso así como también registrar los cambios producidos por el sistema de control, se creó una base de datos. Esta base de

datos se hizo utilizando lenguaje SQL, mediante el uso del sistema de gestión de base de datos MySQL debido a la experiencia previa adquirida con este software.

El Servidor de Control fue implementado sobre un Raspberry Pi B+, el cual es un computador de placa reducida (Ver características en F.3). En él está implementada la base de datos, así como también el servidor web para el HMI (Apache2). Para que este dispositivo se encuentre en la misma red que los Elemento de Control, se conectó al sistema a través de un cable de red (Ethernet) hacia el router, el que entregaba la red inalámbrica para el sistema de control.

Dentro de los distintos componentes del sistema de control, fue necesario realizar un script para el Elemento de Control. También se diseñó e implementó la base de datos y lo mismo se realizó con la página web que muestra el HMI. Cabe destacar, que cada sistema está programado en distintos lenguajes, debido a las limitaciones de software que posee cada parte del sistema, por ejemplo, la página web que aloja en HMI se programa en lenguajes orientados para ello, como lo son HTML, PHP, entre otras, en cambio el microcontrolador del Arduino se programa en un lenguaje de nivel medio como C++, tema que es mencionado con mayor profundidad en la Sección 4.2.

4.1.4. Comunicación entre componentes

Como se mencionó anteriormente, se utilizó una conexión inalámbrica con la finalidad operar y monitorear el sistema de forma remota. El Elemento de Control, que es el componente que va conectado directamente con el proceso puede ser conectado al sensor, transmisor, actuador o controlador, o puede requerir algún tipo de interfaz dependiendo del componente a conectar. Además, cabe mencionar que la conexión inalámbrica a este nivel de conexión no es posible, debido a que la comunicación entre el proceso y el Elemento de Control debe tener la menor latencia posible para no generar problemas en la eficacia del controlador.

Por otro lado, para conectar el Servidor de Control con los distintos elementos de control ubicados en el proceso, se optó por realizar ahí la conexión inalámbrica ya que de esta manera se puede tener un servidor remoto alejado del proceso. Sin embargo, fue necesario utilizar un software en el servidor (denominado Interfaz) que permitió escuchar a los distintos elementos de control y comunicarse con ellos a través de la arquitectura cliente/servidor mediante el uso de Pipes. Además, este software se encargaba de almacenar la información proveniente de los elementos de control y además, enviar mensajes proveniente de acciones realizas por el usuario. El esquema de funcionamiento de este sistema de comunicación se observa en la Figura 4.5.

Este Interfaz al estar incluido dentro del Servidor de Control no posee restricciones como las existentes en el Elemento de Control, ya que funcionan dentro de un computador y no sobre un microcontrolador. Además, se optó por crear este interfaz utilizando Java debido a que se posee mayores conocimientos en este lenguaje, pero esto no implica que no pueda realizarse en otro lenguaje de programación.

Debido a lo anterior, la conexión entre el Interfaz con la base de datos se realizó a través de Connector/J, biblioteca escrita en Java con funciones para conectar, insertar y consultar a bases de datos de MySQL. Por otro lado, las acciones realizadas por el usuario a través del HMI se registraban en la base de datos (utilizando PHP) y además se enviaba el mensaje al interfaz mediante Pipes.

Luego, al reemplazar el Arduino Mega 2560 al Arduino Yún la utilización del Interfaz quedó obsoleta, por lo que la nueva versión del Servidor de Control solo contenía el servidor web y la base de datos.

4.2. Software

4.2.1. Controlador

Para que el Elemento de Control cumpla con las funciones que posee se ideó un software, dentro del Elemento de Control, que realizaba estas 3 tareas dentro de un ciclo que corre indefinidamente, sin embargo, no fue posible paralelizarlas debido a las dificultades que posee Arduino para esto.

Con el objetivo de realizar el monitoreo, se creó bibliotecas que realizaban la conversión de la magnitud medida (voltaje o frecuencia) y la transformaba a un valor medido mediante regresiones utilizando valores experimentales. Con esto, en el ciclo de control se estableció a cual puerto se encontraba conectado el sensor y que cálculo se debe realizar en cada medición solicitada durante la ejecución del ciclo de monitoreo y control.

Para el control del sistema, se utilizó la medición del sensor conectado al lazo de control y se aplicó la ley de control para establecer si el elemento final de control se apaga o se enciende. Esta operación solo se realizaba si el controlador se encontraba en modo automático, ya que en caso contrario este paso se omitía. En la implementación, la ley de control correspondió a un controlador PID con modulador de ancho de banda (PWM).

Debido a la necesidad de comunicar el Elemento de control con el Servidor de Control, se creó una sección en el algoritmo de manera de recibir modificaciones de parámetros del controlador, así como también variables de entradas las que fueron realizadas por el usuario a través del HMI. La Figura 4.1 esquematiza estas funciones y como se relacionan con el proceso y el Servidor de Control. Cabe considerar que este hilo de ejecución corresponde a un ciclo que corre indefinidamente.

El bloque de mediciones toma los valores de las variables de salida desde las entradas correspondientes a los sensores. Posteriormente, estas mediciones se ingresan a la base de datos, para ello se construye en SQL Syntax la inserción y luego se envía a la base de datos. En la primera versión (la que utiliza el Arduino Mega 2560), esta información se encriptó bajo el cifrado ARC4 y luego se enviaba hacia el Interfaz, y en una segunda versión se utilizó

una biblioteca que comunicaba el Arduino directamente con la base de datos.

Por otra parte, en el bloque de control, y a partir de la temperatura medida del sensor ubicado en la parte inferior del equipo (en la salida del calefactor), se utilizaba este valor en la ley de control, el cual arroja como resultado si el elemento final de control (calefactor) debía estar encendido o apagado. El controlador de PID implementado corresponde al que se muestra en la ecuación, representada en la Ecuación 4.1 [15], que considera en el cálculo entre el setpoint y el valor al momento de hacer el cómputo (denominado error); el mismo valor en el computo anterior, y la suma de los errores acumulados .

$$u(k) = K_c \left(1 + \frac{\tau_D}{\triangle t} + \frac{\triangle t}{\tau_I}\right) \varepsilon(k) + \left(\frac{K_c \triangle t}{\tau_I} - \frac{K_c \tau_D}{\triangle t}\right) \varepsilon(k-1) + \frac{K_c \triangle t}{\tau_I} \sum_{\mathrm{i}=1}^{k-2} \varepsilon(\mathrm{i}) \tag{4.1}$$

Además, el controlador PID entrega una salida continua y no acotada, en cambio el elemento final de control solo puede entregar una potencia nula (apagado) o 2000W (cuando está encendido), por lo tanto, el controlador requirió de una conversión continua a control ON/OFF. El algoritmo implementado para esta situación corresponde a la modulación de ancho de pulsos, lo cual es explicado a más profundidad en Anexos G.

Debido a que el sistema consideraba el error en cálculos anteriores, frente a grandes cambios en el setpoint o saturación del actuador ocurrió que el término integral ($I = \sum_{\mathrm{i}=1}^{k-2} \varepsilon(\mathrm{i})$) se acumuló de forma considerable, produciendo oscilaciones en la salida. Este fenómeno se denomina Windup Reset, y para solucionar este problema, se implementó en el controlador que el valor de I esté acotado entre un valor mínimo y máximo (I_{max}).

Luego, el bloque de actualización de parámetros tiene por función modificar los parámetros del controlador que fueron cambiados en el HMI y que son almacenados en la base de datos para, posteriormente, ser actualizados en el controlador. Sin embargo, en este bloque se adquieren cambios de variables de entrada del proceso, como por ejemplo, cambios de potencia de la bomba o encendido del ventilador, los cuales son modificados en el proceso en el siguiente bloque.

El software en el que el Elemento de Control del sistema es programado fue en Arduino IDE bajo el lenguaje Arduino, el cual es un derivado de C++. En la primera implementación de este software se observó que el tiempo entre cada ejecución de la ley de control fue muy largo, debido a que software del Arduino Mega 2560 sólo poseía un hilo de ejecución, y además no existen bibliotecas que permitieran o facilitaran una ejecución en paralelo.

Al probar el sistema de control en el equipo HL630, se observó que al estar encendido el calefactor durante 15 segundos produjo cambios cercanos a 4°C, registrados en el sensor ubicado en la parte inferior del equipo. Como la frecuencia de muestreo y de control fue relativamente lenta en relación al proceso, en la primera versión del software del Elemento de

Control (con el interfaz en el Servidor de Control), el tiempo de entre cada cálculo y control al sistema eran alrededor de 15 segundos, por lo que se pudo ver cambios de temperatura del mismo orden. En cambio, en la segunda versión, al conectar el Arduino directamente con la base de datos se disminuyó este tiempo a 10 segundos, sin embargo, aún fue insuficiente para la implementación, debido a la rápida respuesta del proceso en relación a la respuesta del sistema de control. Es por ello que se probó una nueva versión de Arduino, la tiene por ventaja poseer un procesador con Linux y un microcontrolador con características similares al utilizado anteriormente, lo que permite desarrollar un ciclo de control más rápido.

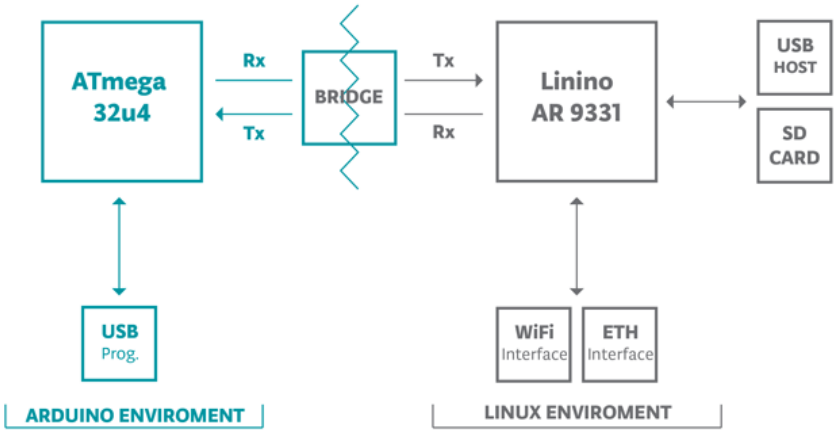


Figura 4.7: Esquema de comunicación entre los componentes del Arduino Yún.

Dado que el procesador tiene un entorno Linux, le otorgó varias características, por ejemplo acceder a bibliotecas más optimizadas que permitirían la ejecución de comandos del ciclo de control en paralelo, y por lo tanto, controlar con mayor frecuencia el proceso. Sin embargo, el manejo de puertos no lo realizó el procesador sino que el microcontrolador, lo que se tradujo en una desventaja ya que el script que se ejecutaba en el procesador requirió establecer una conexión con el microcontrolador. Cabe mencionar que el microcontrolador se programó en lenguaje Arduino y se ejecuta de forma autónoma con respecto al funcionamiento del procesador, programado en Python, y la comunicación entre ambos se realiza a través de una interfaz de red.

Para el Arduino Yún, en el procesador se programó el algoritmo de control y la recolección de datos, el envío a la base de datos, la actualización de parámetros y cambios en el proceso modificado desde el HMI en un mismo hilo de ejecución; en cambio, en el microcontrolador se recibían las conexiones desde el procesador y utilizando el software REST (Representational State Transfer, que actúa a nivel de comunicación cliente/servidor), se ejecutaban comandos a partir de URLs, que permitió al procesador actuar sobre los puertos del Arduino, que fueron manipulados y medidos a través del microcontrolador como se observa en la Figura 4.6.

Por el lado del microcontrolador, se crearon bibliotecas que realizan las operaciones para cada actuador o sensor del sistema. Para modificar un actuador se envió un mensaje a *localhost/arduino/setParam/x/y*, donde x es el número del componente según tabla almacenada en la base de datos (Ver Tabla 4.1), y el valor y correspondía al valor que

se le asignó al componente, por ejemplo, para seleccionar la bomba en la potencia 3 se debe visitar la página *localhost/arduino/setParam/5/3*.

Para realizar mediciones de los sensores del equipo y el estado de los actuadores, se debe utilizó la página *localhost/arduino/measureAll/0*, retornando como mensaje "(1,m_1),(2,m_2)...."donde m_{i} correspondía a la medida del componente i. En el caso de medir un componente, se utilizaba la dirección *localhost/arduino/measureComp/x* donde x es el número del componente. Cabe considerar que el sistema retornaba el estado de todos los componentes manipulados por el microcontrolador (pero no considera los parámetros del sistema de control, por ejemplo parámetros del controlador, lo que son componentes según la base de datos).

En el caso de sensores se retornaba el valor medido, y en cambio, para los actuadores se retornaba el estado en que el Elemento de Control envió al proceso, sin embargo no necesariamente correspondió al estado actual del componente, como podría ocurrir con problemas en la conexión entre el Elemento de Control y el actuador, también ocurriría en el caso que el equipo se encontrará en modo manual o el modo controlador industrial para el calefactor para el equipo HL630 (modo de control externo al sistema de control descrito).

Cabe considerar que este componente no necesariamente es un equipo único, ya que el servidor web y la base de datos pueden estar separadas en equipos distintos. De forma física, este equipo puede estar implementado, por ejemplo a un computador, donde en él se encuentre alojado un servidor web que permita mostrar el HMI y en otro computador distinto un servidor que contenga la base de datos.

4.2.2. Interfaz humano-máquina

Cómo se mencionó anteriormente, el interfaz humano-máquina (HMI) se diseñó como una aplicación web debido a que elimina la necesidad de realizar instalaciones y problemas de compatibilidad en los distintos sistemas operativos. El HMI diseñado fue dividido en 3 secciones: un encabezado, un cuerpo y un panel lateral (Ver figura 4.8).

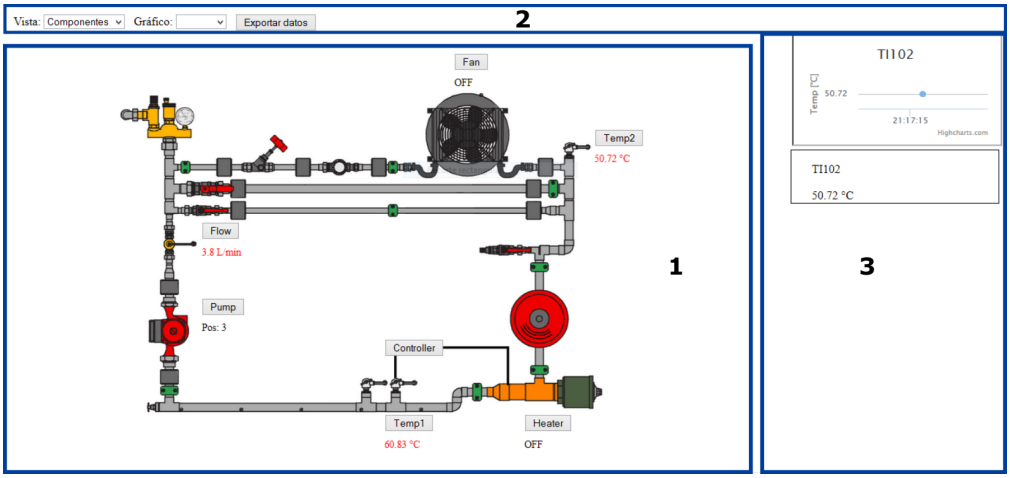


Figura 4.8: División del HMI en 3 secciones: (1) Cuerpo principal, (2) Encabezado y (3) Barra lateral.

El encabezado del HMI incluye un selector que indica si el usuario desea observar el diagrama del proceso como un diagrama de flujo o un diagrama más cercano al aspecto físico del equipo. Además, este panel incluye un enlace que abre una nueva ventana con el registro histórico del proceso, y un botón que abre otra ventana para observar el formulario utilizado para descargar los registros desde la base de datos.

El cuerpo principal muestra el diagrama del proceso, diagrama de flujo o un diagrama más visual que incluye los componentes del sistema, además en cada componente se encuentra un indicador de valor actual del proceso y un botón que permite observa el menú de configuración de dicho componente.

Por otro lado, el panel lateral muestra un gráfico que muestra los valores medidos en tiempo real y además el menú de configuración del cada componente, el cual sólo se puede observar al seleccionar dicho componente en el diagrama. Este menú contiene botones que permitirán cambiar el funcionamiento de los actuadores (encender, apagar o cambiar el estado), cambiar parámetros del controlador, así como también abrir o cerrar el lazo de control mediante la selección del funcionamiento del controlador en modo automático o manual.

El interfaz humano-máquina se implementó en un servidor HTTP, donde la comunicación con la base de datos se realizó utilizando PHP, para la interacción con el usuario se utilizó Javascript y AJAX (Asynchronous JavaScript And XML) y la creación de gráficos usando HighChart y HighStock. La página diseñada tiene las mismas características mencionadas en el diseño del HMI.

Se escogió utilizar el sistema bajo un servidor HTTP debido a que favorece la adaptabilidad a distintos tamaños de pantalla, tanto de computadores, así como también para dispositivos móviles como smartphones y tablets. Además, no requiere instalar algún tipo de programa en el computador o dispositivo en el que se observará el HMI. Sin embargo, la vista del HMI está optimizado para pantallas con relación de aspecto 16:9.

En la implementación del sistema de control desarrollado en el equipo HL630, como ya fue mencionado, el HMI se implementó con 2 vistas: un diagrama de flujos y una vista más realista de los componentes del sistema. En cada componente del equipo aparece un botón, el cual al ser presionado despliega en la columna izquierda de la pantalla un gráfico que actualiza en tiempo real el valor del componente, que en caso de ser un sensor es el valor medido y en caso de ser un actuador muestra el estado aparente en el equipo (último registro de comando enviado desde el Elemento de Control al proceso). Bajo este gráfico se muestra el nombre del componente, que es el número de identificación según la norma ISA 5.1, el valor del componente ya mencionado y las opciones de acciones a realizar. En el caso de sensores que no estén en el lazo de control, por ejemplo en el sensor de temperatura superior del equipo y el sensor de flujo, para ellos este bloque solo se da el nombre y el valor medido; en el caso contrario, aparece un cuadro que muestra el valor del setpoint y un campo de texto para modificar este valor. Para los actuadores también se muestran acciones, por ejemplo: encender, apagar o cambiar la potencia. Para el controlador, se muestra el modo de funcionamiento (manual o automático), botones para cambiar el modo de funcionamiento y el valor de los parámetros del controlador (K_C, τ_I, τ_D). Cabe considerar que si está seleccionado el controlador, el gráfico que se muestra corresponde al valor del sensor de temperatura que pertenece al lazo de control.

Tabla 4.1: TAG Number según norma ISA para los componentes del equipo.

TAG Number	Descripción
TI101	Sensor de temperatura ubicado a continuación del calefactor
TI102	Sensor de temperatura ubicado a continuación del radiador
TIC101	Calefactor
TIC102	Radiador
FI103	Sensor de flujo
FIC103	Bomba

El modo manual del controlador corresponde a aquel donde el funcionamiento de todos los actuadores del sistema se manipulan a través del HMI, incluyendo el calefactor. Todas las modificaciones de los parámetros del controlador se registran en la base de datos, pero el actuar del calefactor depende del botón de encendido o apagado (ubicado en el panel de dicho componente). En el modo automático, cada vez que se oprime el botón se resetea el valor del setpoint (a 0°C), y los valores de los parámetros del controlador a K_C=0, τ_I=10000 y τ_D=0, sin embargo, el HMI no envía estos valores al Servidor de Control, sino que este reseteo lo realiza el Elemento de Control, y al momento de iniciar el modo automático fija los valores ya mencionados.

Al oprimir un botón que selecciona un componente o realiza una modificación, la acción es efectuada por AJAX; de manera que al actualizar el despliegue del panel o actualizar el valor del componente o parámetro no se requiera refrescar la página. Lo mismo ocurre con los valores mostrados en el diagrama principal del HMI que se actualizan cada 2 segundos. En el caso que se presenten problemas en el muestreo, estos se representarán por colores: el color de fuente negro indica que el sistema de monitoreo trabaja con normalidad, en cambio el color amarillo y rojo indican problemas entre la actualización del valor mostrado y el momento

en que fue medido. Este problema puede ocurrir debido a que el ciclo de muestreo, y por lo tanto de control, es mayor al diseñado; o por la existencia de problemas de comunicación entre el Elemento de Control y el proceso, así como también, problemas en la base de datos. Si el intervalo de tiempo entre que el valor mostrado es actualizado y el tiempo de muestreo es mayor a 10 segundos se cambiará a color amarillo. En cambio, si este lapso es mayor a 20 segundos, será de color rojo.

Además, para visualizar los valores medidos, se muestra en otra página una línea de tiempo donde se observa el valor de la variable en el tiempo. Además, este gráfico se actualiza en tiempo real. En esta página se puede elegir el rango de datos a mostrar a través de una barra deslizable, eligiendo el inicio y el fin de ésta.

Para obtener los datos registrados en la base de datos, se agregó al HMI un vínculo para acceder a un formulario que indica el tipo de dato variable/parámetro, el nombre del componente y el intervalo de tiempo. Una vez ingresado correctamente estos datos, y utilizando la biblioteca PHPExcel, se genera automáticamente una hoja de cálculo que contiene los datos solicitados.

En la sección de Anexos J se observan imágenes del HMI implementado en el equipo.

4.2.3. Base de datos

Debido a la gran cantidad de datos a almacenar, ya que en cada ciclo de monitoreo se almacena el estado de los 3 sensores y 3 actuadores, lo cual se aproxima a alrededor de 360 datos almacenados por minuto, sumado al registro de modificaciones realizadas por el usuario hace que se requiera de la utilización de una base de datos para almacenar de forma segura y eficiente estos valores.

Para almacenar estos valores, se diferenció en 2 grupos los registros guardados: las mediciones realizadas en el equipo y las modificaciones realizadas por el usuario. A pesar que ambas tablas comparten columnas como componente, fecha y hora, y magnitud de la variable/parámetro; la diferencia radica en el objetivo de cada tabla: la de mediciones es utilizada para monitorear el sistema, en cambio la tabla de modificaciones está hecha con el objetivo de registrar cambios como parámetros y estado de actuadores, los cuales son enviados hacia el Elemento de control en la versión final del sistema de comunicación.

Como se mencionó en el capítulo anterior, la base de datos tiene como función almacenar las mediciones registradas por el o los elementos de control. En la implementación se le agrega otra función, una tabla que registra las modificaciones de los parámetros o actuadores del sistema, para que el Elemento de Control las modifique a partir de cambios enviados desde el HMI. El sistema de gestión de la base de datos es MariaDB, derivado de MySQL, y se utiliza como mecanismo de almacenamiento InnoDB. Adicionalmente, la implementación se realizó en el Servidor de Control que corresponde al Raspberry Pi B+. El diagrama que muestra las tablas y sus relaciones se encuentra en la sección de Anexos H.

4.3. Implementación

4.3.1. Conexión al equipo

Para comunicar el Elemento de Control con el proceso, desde el PCB hay 3 entradas: 2 entradas análogas para los sensores de temperatura y 1 entrada digital para el flujómetro. Por otro lado, desde el Arduino, hay 5 salidas digitales: 1 hacia el relé en estado sólido que controla el ventilador, y las restantes hacia los relés electromecánicos, de los cuales 1 controla el calefactor y las otras 3 permiten elegir la potencia de la bomba.

Al implementar lo anterior en el equipo HL630 se obtuvieron datos por medio del Elemento de Control, datos que fueron llevados al servidor, donde se registraron con el fin de ejecutar una acción en el HMI donde se muestra un diagrama del proceso y permite, por medio de botones, la interacción con el usuario. Dichos datos fueron obtenidos por medio de la conexión de los distintos sensores del equipo a través de puertos físicos que posee el Elemento de Control, los cuales miden directamente un voltaje por medio de entradas análogas o el estado de una señal digital, a través de una entrada digital.

La mayoría de las conexiones entre el equipo y el sistema se realizó utilizando la tarjeta de adquisición de datos que posee el equipo, la que se encontraba en mal estado. Se reconocieron 4 líneas a 220V, 3 de las cuales eran para el funcionamiento y elección de potencia de la bomba, y la restante para el encendido del calefactor. En el caso de la bomba, se detectó la configuración que se observa en la Tabla 4.2, esta se hizo a partir de combinar el encendido de estas líneas en el equipo y observando el resultado las mediciones de flujo y el indicador de potencia en el equipo. Para manipular estos actuadores desde el Elemento de Control, se utilizó relés electromecánicos que poseen un voltaje de control de 5V y fueron conectados a las líneas de 220V, mencionadas anteriormente.

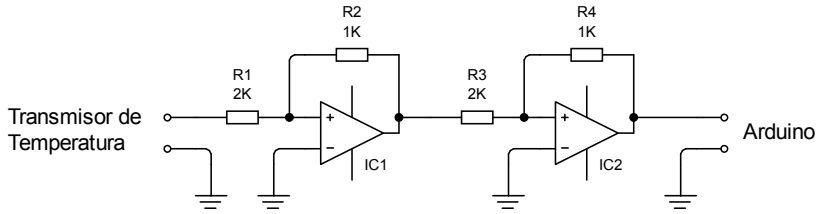
Tabla 4.2: Configuración de líneas para elegir la potencia de funcionamiento de la bomba.

	Línea 1	Línea 2	Línea 3
Apagada			
Potencia 1	ON		
Potencia 2	ON	ON	
Potencia 3	ON		ON

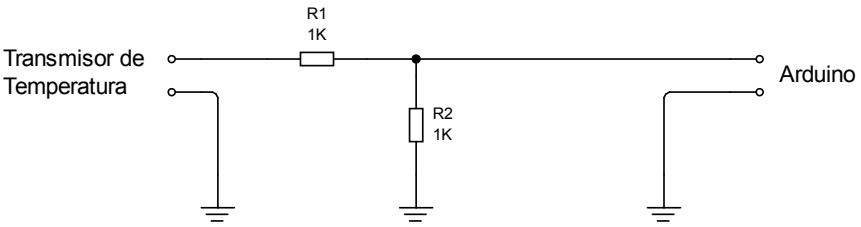
En el caso del ventilador, se detectó que estaba conectado a un relé en estado sólido, donde el voltaje de control que entregaba la tarjeta de adquisición de datos para encenderlo o apagarlo corresponde a 10V, de todas formas se conectó directamente el Elemento de Control a este relé, ya que también permite un voltaje de control de 5V (voltaje máximo que entrega Arduino en los puertos de salida). Cabe destacar que el switch que permite encender o apagar el ventilador de forma manual no utilizaba el relé mencionado, en contraste de la bomba que utilizaba contactores para dicha función (los que pueden ser utilizados tanto por la conexión proveniente de la tarjeta de adquisición de datos y el panel). Para el calefactor

se tuvo la misma conexión que la bomba, y además, se adiciona la manipulación en modo controlador industrial.

Para la medición de temperatura se decidió medir directamente desde la salida del transmisor de temperatura, y a partir de ese voltaje medirlo en una entrada análoga del Arduino. Sin embargo, como se ve en Anexos 2.3.1, la salida del transmisor es entre 0 y 10V y por otro lado, la entrada análoga mide entre 0 y 5V [2], por lo que fue necesario realizar algún circuito que disminuya el voltaje manteniendo la linealidad entre la temperatura que mide el sensor y el voltaje de salida del circuito recién nombrado. En una primera instancia se utilizó el circuito que se observa en la Figura 4.9a, el que fue elaborado con amplificadores operacionales y resistencias de manera que, la salida sea exactamente igual a la mitad de la entrada. Sin embargo, una vez montado en el PCB, fue reemplazado por un circuito más simple (ver Figura 4.9b) debido a que uno o varios amplificadores operacionales se quemaron al momento de ser soldados en la placa, y la sospecha de problemas en la señal, que finalmente se encontró que se debía a la presencia de ruido proveniente del transmisor de temperatura, por lo que este nuevo circuito fue investigado e implementado.



(a) Versión inicial, con *Op Amp*



(b) Versión final, utilizando un divisor de voltaje.

Figura 4.9: Circuito que adapta el voltaje proveniente del transmisor de temperatura.

Como se mencionó anteriormente, el transmisor de temperatura presentó problemas de ruido en la señal de salida, es por ello que se revisaron los sensores, los cables que conectaban el sensor con el transmisor de temperatura (ver conexión en Figura 4.10) y así, se llegó a la conclusión que el problema no estaba en la entrada de este componente sino que estaba en el

equipo mismo. Para eliminar el ruido en el cable que comunica el transmisor de temperatura con el circuito reductor de voltaje, se reemplazó dicho cable por uno apantallado, lo que logró eliminar el ruido en la señal de salida, sin embargo, cuando un contactor cambiaba de estado (por ejemplo, al encender el calefactor) aparecía un cambio drástico de señal (en forma de peak), lo que incluso produjo que el transmisor entregara, a través de la señal de salida, el voltaje mínimo o máximo, es decir, 0 ó 10V. Y esta señal de salida se mantenía hasta que se reiniciaba el equipo.

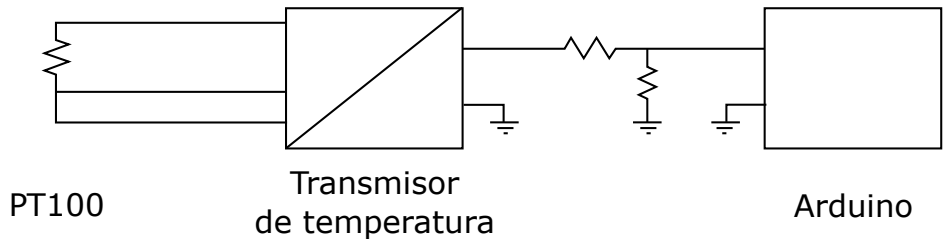


Figura 4.10: Conexión del sensor de temperatura utilizando un sensor de tipo PT100 (RTD), transmisor de temperatura, divisor de voltaje y Arduino para su medición.

Se analizó dos posibles fuentes de ruido, la primera,fue el hecho que la red de alimentación de los distintos componentes eléctricos y la red de control era la misma, por lo tanto, el cambio de estado de los contactores pudo producir ruido a través de la alimentación y con ello, afectar el transmisor. Por otro lado, se vio la posibilidad que el transmisor de temperatura fuera afectado por el ruido a través de la emisión de ondas electromagnéticas, ya que la separación entre los contactores (5 en total) y los transmisores de temperatura es alrededor de 5 cms. La solución para el primer problema fue probar su existencia al conectar a una red aparte y en caso de desaparecer el ruido se solucionaba aplicando un filtro de línea (Anexos C.1). En cambio para la segunda hipótesis, se decidió implementar una jaula de Fadaray (Anexos C.2). Se eligió la segunda solución, debido a que era más fácil y rápida para implementar, y luego de varias pruebas no se observó ruido, con lo se dio por solucionado el problema. El historial de mediciones para la calibración se muestra en la sección de Anexos D.1.

En el caso del sensor de flujo, y en las condiciones que el equipo de prueba se encontró (desconectado de la tarjeta de adquisición de datos), se configuró según el circuito mostrado en la Figura 2.5, y se conectó directamente al Arduino, al cual se programó un contador de frecuencia (valor que se relacionaba con el flujo medido). Su funcionamiento no presentó problemas hasta que, al realizar el cambio del Arduino Mega 2560 al Arduino Yún, produjo que el mismo script que mide la frecuencia ya no funcionara, debido a que su funcionamiento estaba basado en que durante un lapso de tiempo se interrumpía el microcontrolador al existir un cambio de LOW a HIGH en la señal digital proveniente del sensor de flujo, y esta interrupción provoca que en el Arduino Yún el sistema de comunicación entre el microcontrolador y el procesador se caiga. La solución momentánea para esto correspondió a una función de Arduino que contaba el tiempo hasta el siguiente HIGH en la señal de entrada y que no requiere interrupciones. Sin embargo, el tiempo que mide no necesariamente corresponde a un período de la onda medida y además, la relación entre período y flujo no

es lineal (Ver Anexos D.2).

4.3.2. PCB

Para fijar los circuitos al equipo de forma definitiva se construyó un circuito impreso (PCB), que contiene los relés que controlan los contactores conectados a la bomba y calefactor (también el PCB contiene los circuitos para los sensores de temperatura y el flujómetro). En la sección de anexos se encuentran, de forma detallada, los planos utilizados para la construcción de esta placa, la que fue hecha utilizando el software PCB Wizard, además el ancho de pistas mínimo se determinó a partir de la norma IPC-2221 [14]. Además, el proceso de elaboración del circuito impreso se observa en la sección de Anexos E.

4.3.3. Sintonización

Una vez conectado el sistema de control con el equipo, es necesario sintonizar el controlador para que la respuesta del proceso frente a una perturbación o cambio en la referencia tenga ciertas características deseadas para su operación, cómo por ejemplo que el sistema sea estable, no presente offset y el sistema alcance el estado estacionario entre 20 y 30 minutos para la realización de pruebas experimentales con el equipo. A continuación se observan pruebas y metodologías realizadas para sintonizar el controlador.

Al aplicar encender el calefactor por 30 segundos se observó la respuesta del sistema variando las condiciones de operación (Ver Tabla 4.3), lo que se observa en la Figura 4.11. Al realizar esto se llegó a la conclusión que es imposible utilizar el método de la curva de reacción propuesto por Cohen y Coon [15], ya que no es posible realizar un cambio escalón en la variable de entrada (calefactor) debido a que la temperatura que alcanzaría el sistema en el estado estacionario sería superior a la temperatura máxima de operación en el equipo, lo que desencadenaría el encendido del sistema de seguridad (apagado del calefactor y encendido del ventilador), y con ello no se lograría el estado estacionario para este experimento.

La Figura 4.11 muestra la respuesta del sistema frente a pulsos de la misma duración cuando el flujo fluye a través del radiador (1-3) estando apagado el ventilador, luego estando encendido el ventilador y apagado el calefactor (4), y cuando la válvula que controla el flujo hacia el radiador está cerrada y se abre el paso a través del tubo de mayor diámetro (5-6) (Ver configuración en Tabla 4.3 y procedimiento experimental en la Sección 3.1.2). En él se puede observar que cuando el flujo se mueve a través del radiador se producen las oscilaciones (1-3), sin importar la potencia de la bomba, hecho que no ocurre cuando el sistema desconecta el paso de agua a través del radiador (5-6).

Tabla 4.3: Configuración de válvulas, ventilador y calefactor para analizar respuesta del sistema frente a pulsos de igual forma.

Prueba	Vál. radiador	Vál. tubo grueso	Ventilador	Pot. bomba
1	Abierta	Cerrada	Apagado	3
2	Abierta	Cerrada	Apagado	2
3	Abierta	Cerrada	Apagado	1
4 (Sin pulso)	Abierta	Cerrada	Encendido	3
5	Cerrado	Abierto	Encendido	3
6	Cerrado	Abierto	Encendido	2

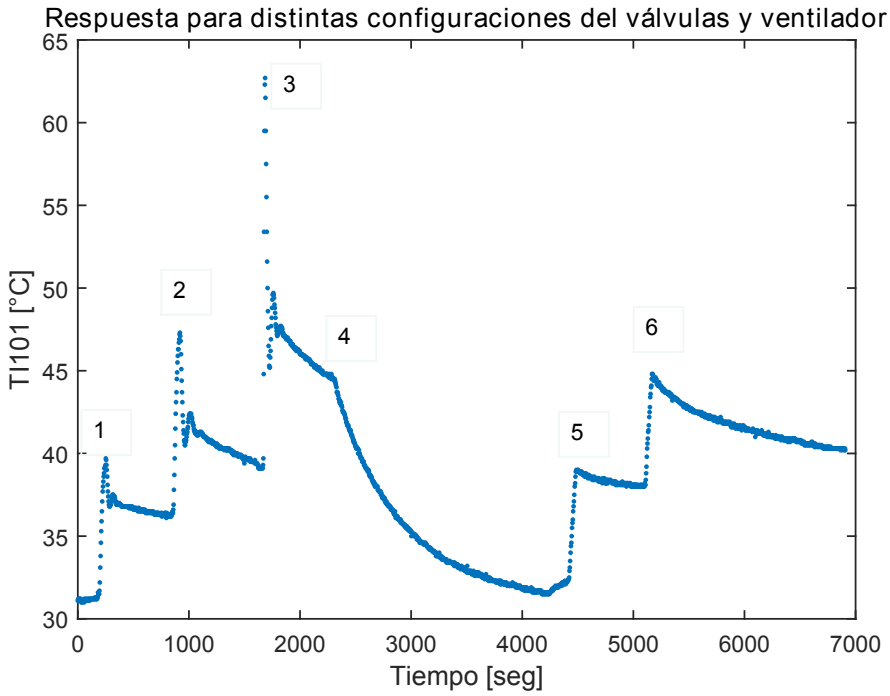


Figura 4.11: Prueba experimental para conocer la respuesta al sistema variando la configuración de válvulas y radiador utilizando pulsos de 30 seg.

Modelación

Debido a lo anterior, se decidió elaborar un modelo matemático de manera de sintonizar el controlador con métodos basados en modelos. Para ello, se toma el supuesto que el diámetro de la tubería y que el campo de velocidades del fluido dentro de la tubería son constantes. Además, como el circuito de agua corresponde a un loop, se considera que la temperatura a la salida del sistema es igual a la de entrada, con ello se obtuvo la ecuación en derivadas parciales mostrada en la Ecuación 4.2. Para realizar este modelo se utilizó como base un

balance de energía para un reactor PFR [11].

$$\frac{\partial T}{\partial t} = -v\frac{\partial T}{\partial z} + \begin{Bmatrix} \frac{q}{A_c \rho C_p} & z \leq L_0 \\ -\frac{4U}{\mathrm{d}\rho C_p}(T - T_{amb}) & z > L_0 \end{Bmatrix} \quad T(t,0) = T(t,L)\,\forall t, \quad T(0,z) = T_{amb}\,\forall z \tag{4.2}$$

Para resolver el modelo de forma numérica, se utilizó en primera instancia un método explícito, pero se observó que la solución era incondicionalmente inestable (Ver Anexos I). Para enfrentar esto, se decidió resolver la EDP con un método implícito, pero la resolución de este problema escapa de las restricciones de tiempo de la memoria.

Para analizar de forma empírica el modelo del sistema, se utilizó la herramienta System Identification Toolbox de Matlab a partir de los datos experimentales. Se decidió estudiar el sistema a partir de la respuesta de un pulso, ya que como se mencionó anteriormente, es imposible realizar este estudio a través de la respuesta de un cambio escalón, y además, realizar un impulso era imposible ya que el corto tiempo de aplicación dificulta la observación del fenómeno producido, esto debido a la resolución de la medición de temperatura ($\triangle T_{min}$ =0,9°C) y a la contaminación dada por otros errores experimentales. Se escogió de manera arbitraria 15 segundos para el ancho del pulso, así se obtuvo el gráfico que se observa en la Figura 4.12. Se observó que el sistema era relativamente lento, ya que tarda varias horas en llegar a un estado cercano al estacionario, por esta razón se utilizó la potencia de la bomba en nivel 3 y el ventilador del radiador encendido, así el sistema es más rápido debido a que la velocidad de transferencia de calor con el ambiente es mayor. Las válvulas de corte del equipo estaban cerradas al momento de realizar el experimento de forma que todo el fluido pase a través del radiador.

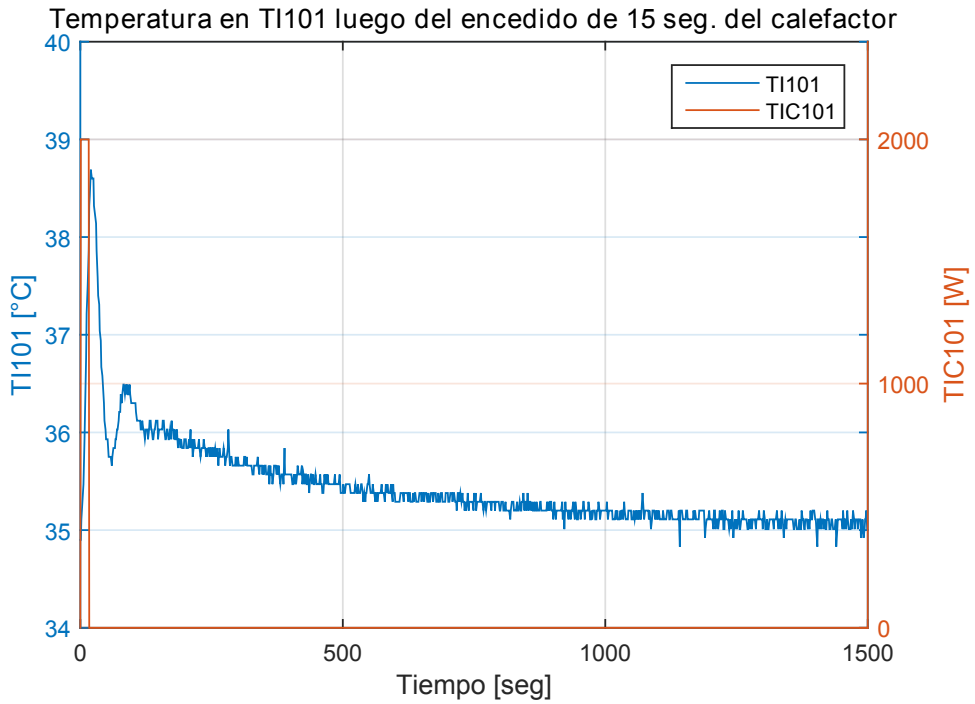


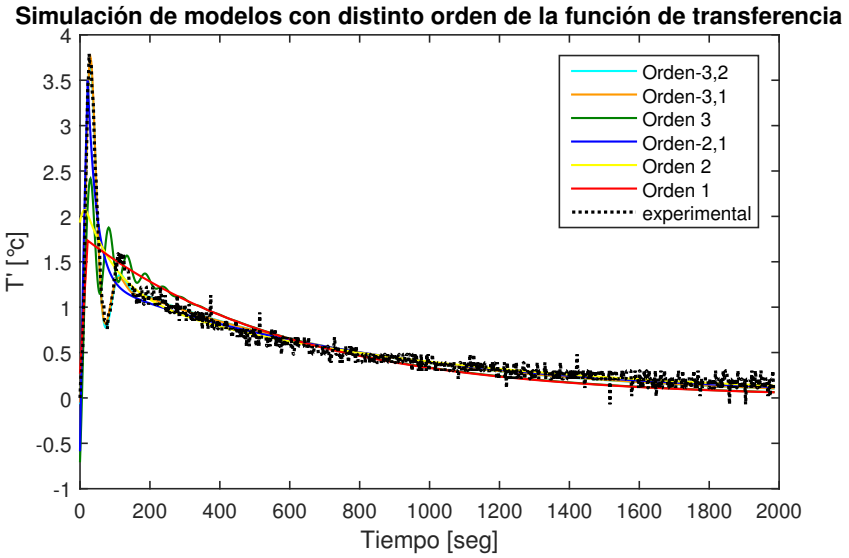
Figura 4.12: Prueba experimental utilizada para obtener el modelo del sistema.

Se observó que sin importar el ancho del pulso, ni la potencia de la bomba, la respuesta del sistema bajo las mismas condiciones de operación posee la misma forma, como se ve en la Figura 4.11; esto implica que dicha oscilación es propia de la respuesta del sistema, y por medio de ésta se pudo definir un criterio para la selección de una función de transferencia. Además, la pendiente de la respuesta luego del pulso es mayor que cero, por lo tanto, el sistema de orden-(p,q) debería ser tal que $p - q = 1$. Se probó los siguiente órdenes para la identificación del sistema en la System Identification Toolbox:

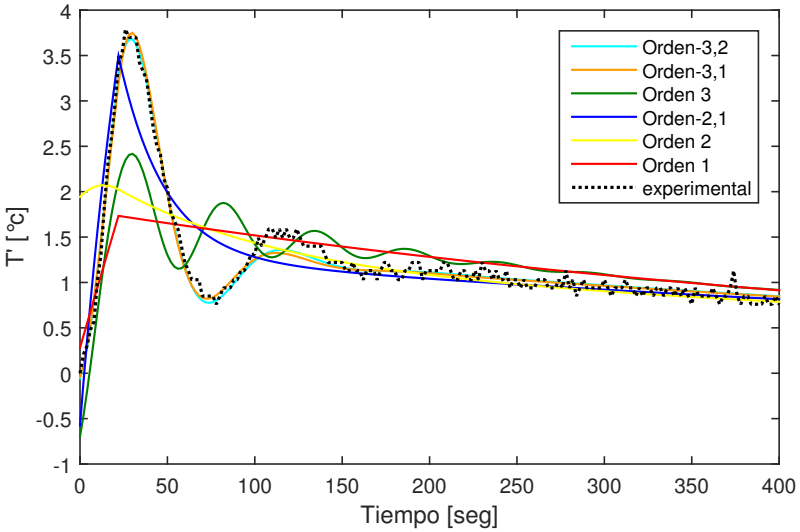
- orden 1
- orden 2
- orden-(2,1)
- orden 3
- orden-(3,1)
- orden-(3,2)

Esta herramienta obtiene los valores de los parámetros de la función de transferencia para los órdenes ya mencionados, minimizando la diferencia entre la salida obtenida experimentalmente y la salida simulada para la función de transferencia de un orden ya establecido. Con ello se obtienen como resultados la Tabla 4.4 que muestra el valor de la función de transferencia, número de ceros y polos, y valores estadísticos que muestran el

grado de ajuste, y la Figura 4.13 que exhibe los datos experimentales y las simulaciones de las funciones de transferencia.



(a) Simulación del sistema con una significancia del 95 %.



(b) Acercamiento entre el intervalo 0-400 seg.

Figura 4.13: Simulaciones para las distintas funciones de transferencia.

$$G_p(s) = \frac{a_0 s^2 + a_1 s + a_2}{b_0 s^3 + b_1 s^2 + b_2 s + b_3} \tag{4.3}$$

Tabla 4.4: Resultados del ajuste de parámetros para distintos tipos de órdenes de la función de transferencia con una significancia del 95 %.

p	q	a_0	a_1	a_2	b_0	b_1	b_2	b_3
1	0			3.39E-5 ± 3.42E-6[1]			1	1.69E-3 ± 8.99E-5
2	0			−6.50E-7 ± 4.02E-7		1	1.22E-2 ± 2.93E-3	1.22E-5 ± 3.92E-6
2	1		1.12E-4 ± 1.86E-5	1.24E-6 ± 1.31E-6		1	3.99E-2 ± 5.51E-3	4.69E-5 ± 7.57E-6
3	0			5.12E-7 ± 1.73E-7	1	3.25E-2 ± 7.69E-3	1.45E-2 ± 1.20E-3	2.43E-5 ± 2.39E-6
3	1		1.03E-5 ± 1.89E-6	2.14E-7 ± 1.62E-7	1	7.71E-2 ± 4.78E-3	7.04E-3 ± 2.19E-4	9.09E-6 ± 4.57E-7
3	2	2.61E-5 ± 1.41E-5	1.00E-5 ± 2.12E-6	2.10E-7 ± 2.00E-7	1	6.77E-2 ± 4.53E-3	6.28E-3 ± 2.51E-4	8.19E-6 ± 4.74E-7

Tabla 4.5: Análisis estadísticos de los coeficientes obtenidos para cada orden de la función de transferencia.

p	q	NRMSE [2]	R	R^2
1	0	49.7 %	0.8361	0.6968
2	0	6.97 %	0.3718	0.1346
2	1	71.6 %	0.7640	0.5707
3	0	55.3 %	0.9241	0.8485
3	1	83.7 %	0.9405	0.8832
3	2	84.1 %	0.9157	0.8379

[1] el valor mostrado con ± corresponde al intervalo de confianza.

[2] Normalized root mean square error.

$$\text{NRMSE} = 100 \cdot \left(1 - \frac{\|y_{exp} - y_{sim}\|}{\|y_{exp} - \bar{y}_{sim}\|}\right)$$

Como se mencionó anteriormente, el orden del sistema debe ser tal que la diferencia entre el número de ceros y polos sea igual a 1. Por lo que se escogió aquellas órdenes que cumplen la restricción impuesta y la elegida fue aquella que tuvo, cualitativamente, la misma forma de la respuesta experimental, teniendo el menor orden posible, de manera que el ajuste no esté afectado por la cantidad de parámetros. Así, se obtuvo que la función de transferencia que mejor representa el sistema es aquella que posee orden-(3,2), función de transferencia del sistema se observa en la Ecuación 4.4 y en la Figura 4.14 se muestra una simulación de esta función de transferencia y una banda en la cual se encuentra la trayectoria del proceso utilizando una significancia del 95 %.

$$G_p(s) = \frac{2.61 \cdot 10^{-5}s^2 + 1.00 \cdot 10^{-5}s + 2.10 \cdot 10^{-7}}{s^3 + 6.77 \cdot 10^{-2}s^2 + 6.28 \cdot 10^{-3}s + 8.19 \cdot 10^{-6}} \tag{4.4}$$

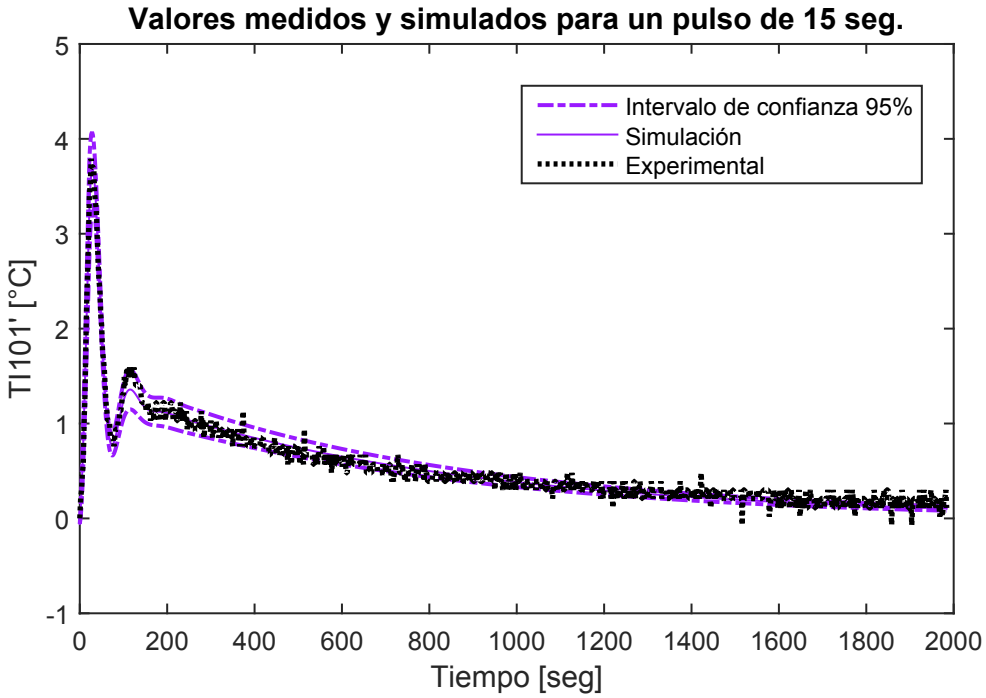


Figura 4.14: Simulación de la función de transferencia del sistema con una significancia del 95 %.

Sintonización del controlador

Para la función de transferencia mencionada, se pensó utilizar el método de Ziegler-Nichols para sintonizar el controlador, pero se observó que el sistema no presenta frecuencia de crossover (ω_{co}) como se ve en la Figura 4.15.

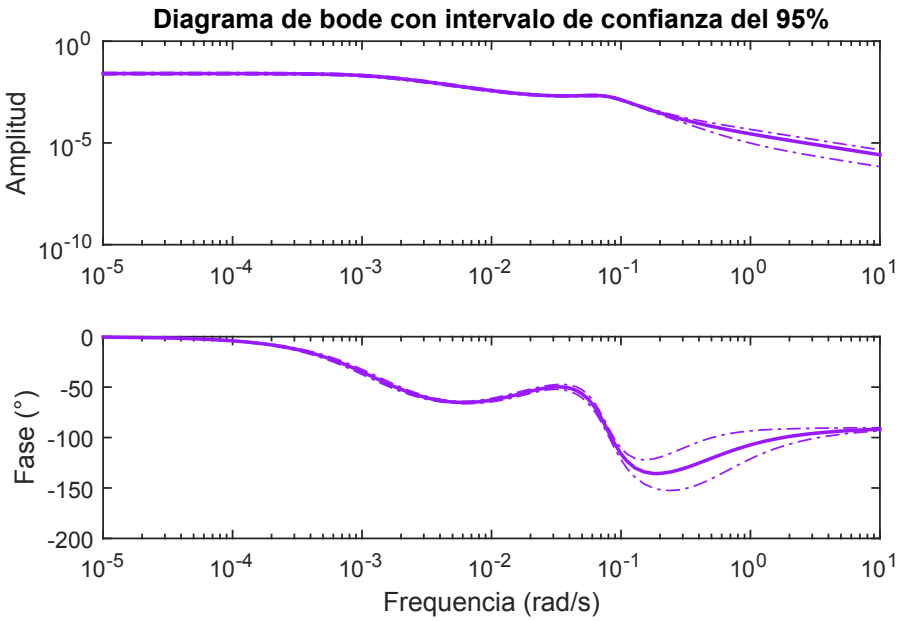


Figura 4.15: Diagrama de Bode para la función de transferencia de orden-(3,2) con una significancia del 95 %.

Para sintonizar el controlador se utilizó la herramienta PID Tuner de Matlab que se basa en realizar simulaciones para la respuesta del sistema frente a un cambio escalón, utilizando un margen de estabilidad de 60° con respecto a ω_{co}, luego el programa realiza 2 optimizaciones para minimizar el overshoot para el problema servo control y para el problema de regulación. Además de los parámetros obtenidos de esta forma, el programa permite modificar la velocidad de respuesta y el comportamiento dinámico simulando la respuesta en cada modificación. Con ello, se simuló el sistema como un sistema continuo y se asume que la función de transferencia del sensor y actuador son instantáneas, y con ganancia unitaria. Además, se consideró que el actuador entrega una salida continua. Como resultado de esto, los parámetros obtenidos con PID Tuner para el sistema descrito se observan en la Tabla 4.6. El diagrama utilizado para sintonizar el controlador se observa en la Figura 4.16.

Tabla 4.6: Sintonización de un controlador tipo PID utilizando PID Tuner para una respuesta cerca a 100 segundos.

Parámetros	Valor
K_c	40,9665
τ_I	$2{,}665{\cdot}10^{-3}$
τ_D	1,57458
N	0,30596

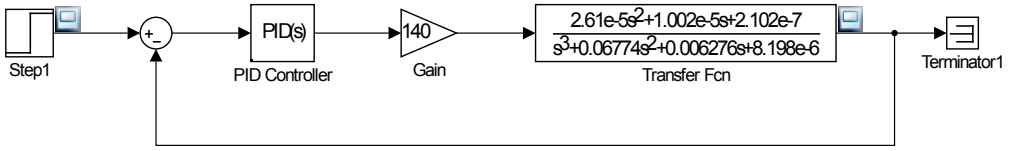


Figura 4.16: Diagrama de bloques para sintonización del controlador usando Tuner PID usando modelo continuo del sistema.

Sin embargo, la utilización de los parámetros de este controlador no tendrán la misma respuesta, ya que el controlador no es el mismo (controlador continuo vs discreto), sumado al efecto que produce el modulador de ancho de pulso en el sistema. Se intentó modelar a través de simulink una versión más real, que considere el modulador de ancho de pulso en el controlador, sin embargo, PID Tuner no puede utilizar este modelo ya que no puede linealizar el sistema. El diagrama de esta versión del modelo se observa en la Figura 4.17.

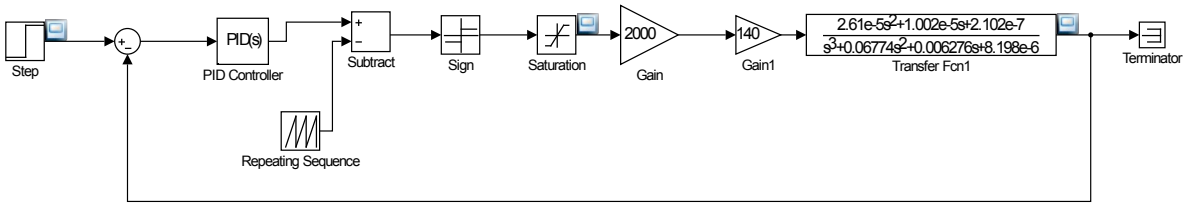


Figura 4.17: Diagrama de bloques para modelo continuo del sistema usando controlador con PWM.

En primera instancia, se utilizaron los parámetros similares obtenidos mediante el método descrito y aplicados en el controlador implementado en el equipo, pero como era de esperar éstos no tuvieron el mismo comportamiento observado en el sistema simulado. La utilización de un valor muy bajo de τ_I produce que el sistema produzca un overshoot de gran magnitud y un largo tiempo de estabilización, como se observa en la Figura X, ya que en la ley de control este término se ubica en el denominador de varios términos produciendo este fenómeno. Debido a esto, se desarrolló experimentalmente a través de pruebas de sensibilidad con este parámetro, siendo modificadas desde el HMI, lo que permitió observar una tendencia y obtener valores del parámetro τ_I que produce la disminución del overshoot y oscilaciones (Ver Tabla 4.7), lo que se aprecia en la Figura 4.18.

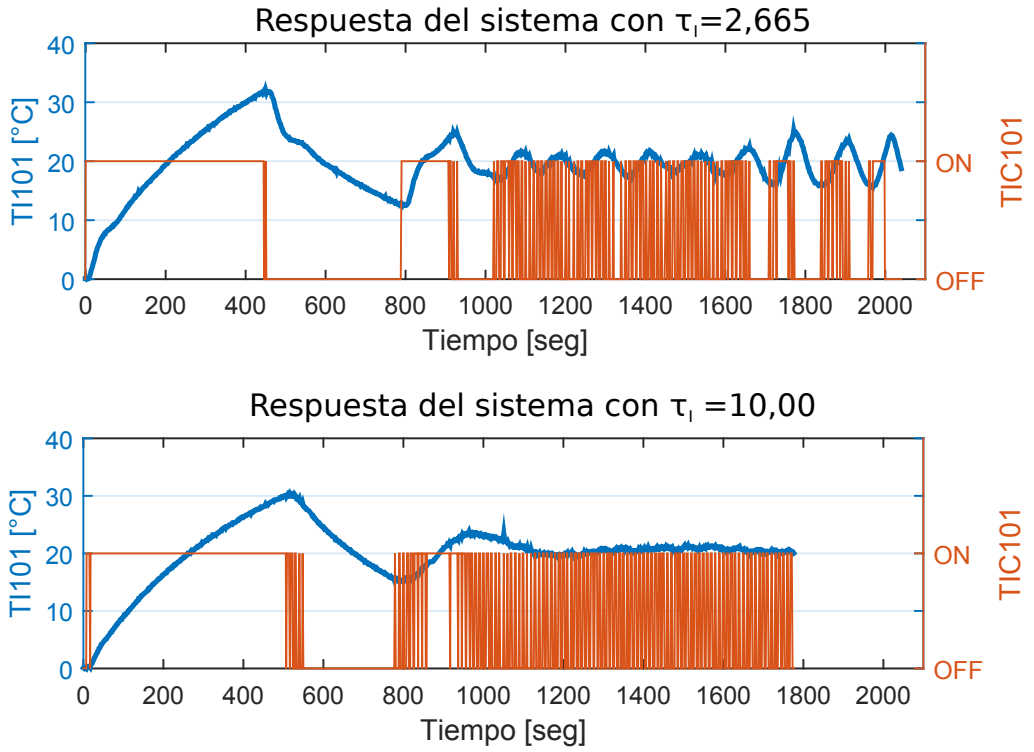


Figura 4.18: Respuesta del sistema frente a la acción del controlador al aumentar el setpoint en 20°C para distintos valores de parámetros.

Tabla 4.7: Valores de los parámetros del controlador para distintas pruebas.

Parámetros	Prueba 1	Prueba 2
K_c	40,96	40,96
τ_I	2,665	10,00
τ_D	1,575	1,575
I_{max}	1000	1000

4.3.4. Pruebas de sensibilidad

Con el objetivo de probar el correcto funcionamiento del sistema de control, se realizaron 2 pruebas de sensibilidad con el objetivo de estudiar 2 parámetros del controlador, que corresponden al periodo de oscilación de la señal de referencia del modulador PWM y además, el valor de I_{max} valor que limita al error acumulado en el controlador discreto. Cabe destacar que a diferencia de otros parámetros del controlador (K_c, τ_I y τ_D) estos no son modificables desde el HMI del sistema de control, sino que deben ser modificados directamente en el algoritmo de control.

En la Figura 4.19 se observa como cambia la respuesta del sistema al variar el periodo de la señal de referencia de 10 a 20 segundos.

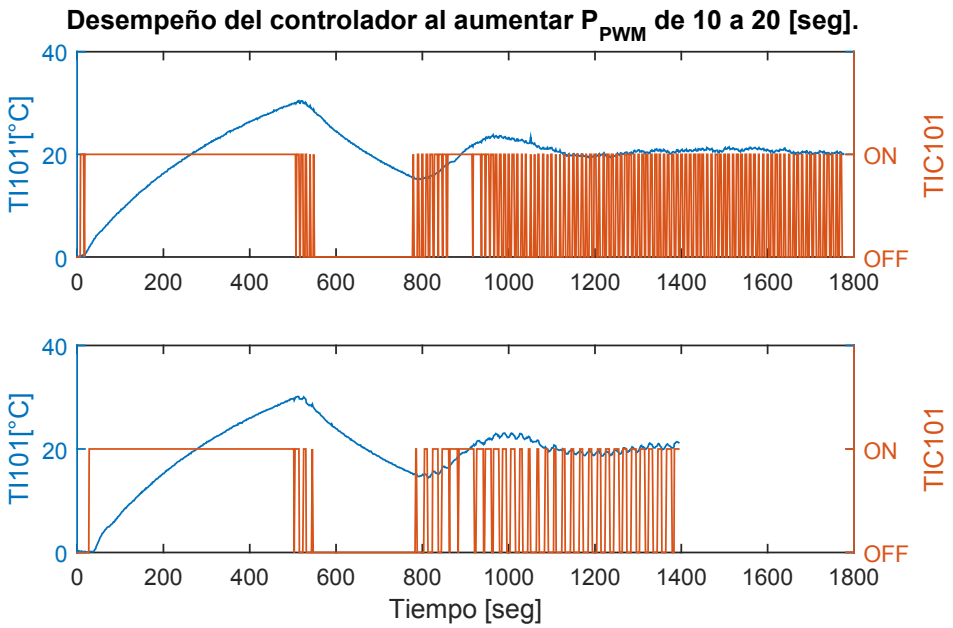


Figura 4.19: Prueba experimental para conocer el efecto del período de la señal de referencia del PWM.

En la Figura 4.20 se observa como cambia la respuesta del sistema al modificar el valor de I_{max} de 1000 a 100[W].

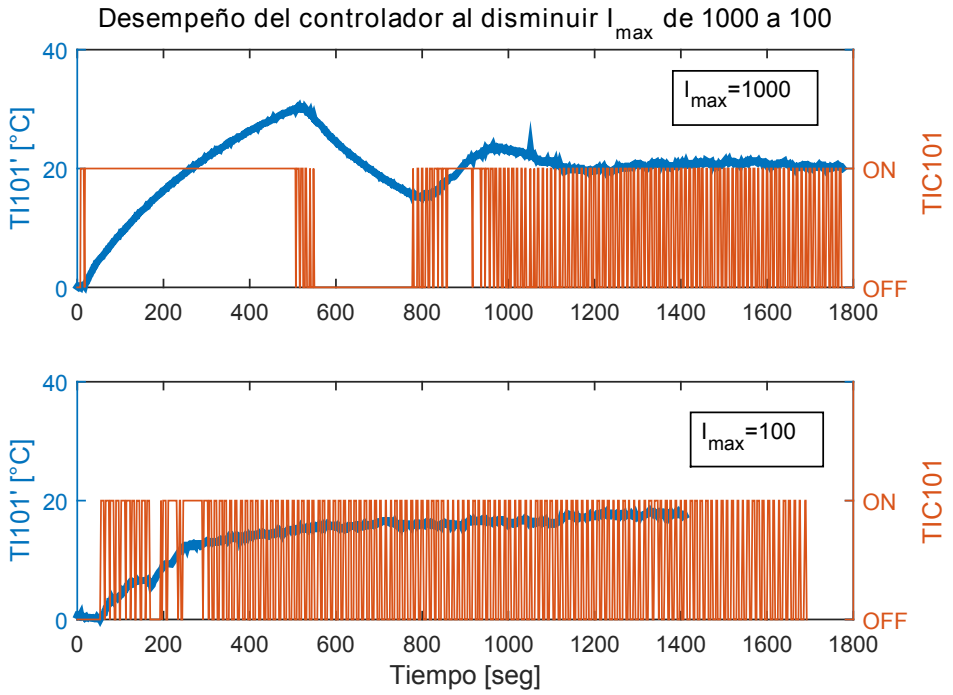


Figura 4.20: Prueba experimental para conocer el efecto del valor máximo de error acumulado en el desempeño del controlador.

Capítulo 5

Discusiones

Durante el trabajo realizado hubieron varias decisiones que tomar, ya sea enfrentando distintas alternativas, considerando las restricciones presentes, además de las ventajas y desventajas que traían para el diseño y la implementación del sistema, su funcionamiento y sus componentes.

5.1. Sistema de Control

Un sistema de control distribuido tiene varias ventajas frente a un sistema centralizado, por ejemplo, el nivel de robustez frente a la falla en un controlador o sistema de comunicación, así como también, en la capacidad de cálculo y el rendimiento del controlador, ya que un sistema distribuido permite manipular cientos de controladores trabajando en paralelo, en cambio, un sistema centralizado controla todo el proceso utilizando sólo un computador. Sin embargo, un sistema distruibuido tiene algunas desventajas, como lo son el alto costo debido a la gran cantidad de componentes como controladores, sistemas de comunicación, base de datos, RTU (Remote Terminal Unit), entre otros, y además del valor del hardware requiere un software dedicado al control, registro y monitoreo, el servicio de mantención y aveces de operación del sistemas, así como también la licencia del proveedor.

Sin embargo, en equipos a pequeña escala o en laboratorios la implementación de este tipo de sistemas no tiene sentido, debido al gran costo, además existe una diferencia sustancial entre un sistema de control de un equipo de laboratorio y uno industrial: el hecho que, para sistemas de laboratorio traen un sistema de control dentro del equipo, y está diseñado para ser conectado a un computador a través de cable serial o incluso USB, lo que no permite conectar varios equipos a un sólo sistema porque que no están diseñados para ser conectado a un mismo sistema del tipo DCS; en cambio, para la industria, el sistema de control se diseña en conjunto con el proceso, y éste ocupa controladores y sistemas de comunicación hechos bajo ciertas normas para ser agregados a un sistema de control y así a un sistema HMI/SCADA. Por lo tanto, la incorporación de sistemas de uso industrial a pequeña escala

no es una buena opción, ni del punto de vista técnico ni económico. Es por ello que un sistema como el descrito en este documento permitiría cubrir la brecha no atendida por los actuales proveedores de sistemas de control.

Otra ventaja del sistema es la utilización de WiFi para la comunicación entre los elementos de control, debido a que elimina la necesidad de establecer cableados entre los componentes del sistema de control. Esta tecnología a nació en los últimos años, sin embargo problemas de seguridad, como por ejemplo la inyección e intersección de paquetes; y la baja velocidad de transmisión de datos (1Mbps) a fines de los años 90, no permitieron el uso de esta tecnología hasta hace algunos años, donde se mejoró el sistema de encriptación y además la velocidad de transmisión es mucho mayor (cercana a 150Mbps [7]), incluso el estándar IEEE 802.11ac, aprobado en 2014, permitiría alcanzar teóricamente los 1Gbps [7]. Este sistema ya está siendo utilizado a nivel industrial, utilizando el estándar IEEE 802.11n en las frecuencias 2,4 y 5 GHz [19], un ejemplo de esto es el sistema ilustrado en la Figura 5.1.

A pesar de lo anterior, este tipo de comunicación presenta interferencia debido al funcionamiento del proceso, sin embargo, se ha determinado que sobre 1,5GHz no existen armónicos en la industria de procesos [21]. Además, estos sistemas aún poseen una latencia alta (mayor a 10ms) para aplicaciones en tiempo real (RT) lo que convierte a este sistema inviable para control de movimiento, aplicación que requiere un estándar como PROFINET RT o IRT (Isochronous Real-Time) [5]. Incluso, una puerta de acceso para una red inalámbrica permite a lo más conectar 4 dispositivos (sin afectar la latencia del sistema), por lo tanto para desarrollar un proceso con gran cantidad de sensores, actuadores y controladores requerirá cubrir con una gran variedad de puertas de acceso a la red industrial. Pero la facilidad que entrega esta tecnología, eliminando el cableado, entre sensores/actuadores y controlador o entre el controlador y el sistema de control, los costos de mantención y además mejora la movilidad para ciertos procesos (especialmente para la manufactura), permitiendo que la utilización de comunicación inalámbrica sea una tecnología prometedora para los próximos años.

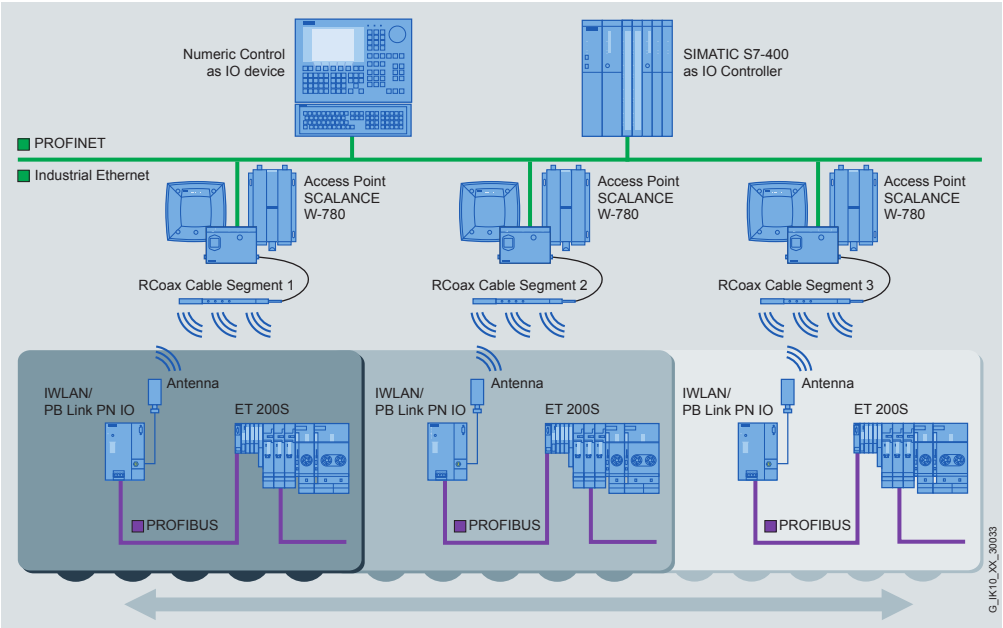


Figura 5.1: Utilización de sistema IWAN(Industrial Wireless LAN) utilizado en monorraíles, figura tomada desde Industrial Wireless Communication, Siemens AG [19].

Por otro lado, la utilización de sistemas embebidos como Arduino y Raspberry permiten diseñar componentes y adaptarse a cualquier tipo de protocolo o estándar, lo cual los convierten en dispositivos muy versátiles y aptos para construir este tipo de sistemas. Sin embargo, carecen de ciertas características deseables para un controlador, como por ejemplo la estabilidad y la baja latencia. En general, la estabilidad del sistema viene dada por el software que posea, como por ejemplo, el hecho que un sistema operativo sea capaz de resolver problemas y pueda recuperarse sin necesidad de tener que reiniciarse. En cambio la baja latencia corresponde a una mezcla entre hardware y software, ya que la velocidad de respuesta depende de los componentes físicos, así como también de la velocidad con la cual es procesado, lo que permitirá que el sistema funcione en tiempo real (RT). En este caso en particular, para programar en el IDE de Arduino se ocupa C++, sin embargo, el programa es compilado a lenguaje de la máquina y se escribe directamente en el microcontrolador, debido a que este dispositivo no posee un sistema operativo. Sin embargo, el gran problema de este sistema es la baja cantidad de recursos (memoria y capacidad de cálculo para sistemas con microcontroladores de 8 bits), lo que limita algunas características, como por ejemplo, la conexión a la base de datos.

Por otro lado, Raspberry Pi si posee un sistema operativo y posee una biblioteca en Python para controlar el GPIO (General Purpose Input/Output), que es similar a los puertos de entrada y salida de Arduino. Además, posee un procesador con algunas instrucciones de 32 bits de tipo ARM lo que le otorga una mayor capacidad de cálculo. Sin embargo, posee una gran desventaja frente a Arduino: su alta latencia. Es por ello, que se escogió utilizar Arduino a nivel de planta (conectado al proceso) y Raspberry Pi para aplicaciones que requiere más capacidades y su tiempo de respuesta no afecta mucho al sistema de control. Sin embargo, hay que indicar que estos dispositivos son placas de entrenamiento y fueron diseñadas para uso doméstico. Por otro lado, sistemas industriales como PLC están diseñadas para uso

industrial: resisten vibraciones, condiciones ambientales más extremas y poseen sistemas de comunicación integrados para los distintos estándares industriales, lo que es justamente lo que Arduino necesita aplicar.

Otro hecho importante, es que el sistema propuesto no incluye un sistema de administración de alarmas, ya que requiere el uso de un dispositivo que actúe de forma paralela al sistema de control y que además, actúe en tiempo real, para una mejor contención de las emergencias. Incluso estos sistemas deben poseer un sistema de gestión aparte del sistema de control, que tenga un tipo de inteligencia que discrimine en poco tiempo el nivel de la alarma, que registre y muestre en el HMI, e incluso pueda desencadenar ciertas acciones para aplacar la alarma, lo que correspondería a utilizar finalmente 2 sistemas (uno de control y otro para alarmas) de forma simultánea.

Cabe destacar, que para la implementación del sistema se utilizó solo un equipo con un solo elemento de control, sin embargo, la elaboración de la base de datos está hecha para conectar una mayor cantidad de estos dispositivos, los que pueden estar asociados a un proceso o equipo en particular, de esta forma para agregar un nuevo equipo o controlar otra sección de un proceso, es necesario registrar este nuevo sistema, y registrar el o los nuevos elementos de control, además de registrar los componentes (sensores, actuadores y parámetros) a la base de datos y relacionarlos con el elemento de control al que pertenecen o están conectados. Sin embargo, la creación del HMI está personalizada para el equipo, por lo tanto, es necesario crear un diagrama o imagen a utilizar, una tabla que contiene los botones y texto a mostrar, y el contenido de los menús al seleccionar un componente del HMI. Para generar un HMI para un nuevo equipo se debe desarrollar una nueva página web, escribiéndola en lenguaje HTML y las acciones en Javascript. No obstante, la creación de un framework podría facilitar la construcción de esta página a través de elaboración gráfica por medio de módulos.

5.2. Implementación

5.2.1. Hardware

Cómo se vio en la sección 4.2.1, la implementación del Arduino Mega 2560 como elemento de control afectó en gran medida el tiempo de ejecución del ciclo de control, debido a que la comunicación con la base de datos toma alrededor de 10 segundos. Luego, al cambiar al Arduino Yún este tiempo disminuyó a cerca de 1 segundo, lo cual es una gran mejora para el sistema de control, pero este valor puede disminuir aún más, ya que el sistema implementado corresponde a una serie de instrucciones secuenciales. Si se divide la tarea de controlar y monitorear se puede mejorar el tiempo de ejecución del controlador, ya que el cuello de botella se encuentra en la comunicación entre el microcontrolador y procesador, y dado que actualmente el sistema se comunica 3 veces por ciclo, éste se podría disminuir a 2 veces por ciclo, donde una comunicación corresponde a obtener la medición del sensor y otra a enviar el orden para modificar el proceso a través del actuador. De esta forma el programa que sigue el ciclo de control tiene sólo 4 ordenes:

1. Solicitar al microcontrolador el valor de la variable medible.
2. Calcular el valor de la variable manipulable.
3. Enviar al microcontrolador el valor de la variable manipulable del lazo de control.
4. Obtener desde la base de datos el valor de los parámetros (en caso de modificación desde el HMI.

Por otro lado, un programa para un ciclo de monitoreo tendría las siguientes órdenes:

1. Solicitar al microcontrolador el valor de las variables medibles.
2. Enviar a la base de datos las mediciones.
3. Obtener desde la base de datos valores de variables manipulables (por ejemplo ventilador).
4. Enviar al microcontrolador los valores de las variables manipulables no incluidas en el lazo de control.

Si bien, esta nueva implementación disminuiría el tiempo del ciclo de control ya que disminuye la cantidad de las operaciones más costosas (en término de tiempo), este sistema requiere de mayor interacciones entre el microcontrolador y el procesador del Arduino Yún, velocidad de respuesta que no está documentada.

Del mismo modo, existe otra configuración que podría seguir disminuyendo el tiempo de control, la cual trata de realizar el loop de control dentro del microcontrolador del Arduino Yún, pero la comunicación con la base de datos la realiza el procesador del mismo dispositivo, además, para facilitar la comunicación entre el microcontrolador y el procesador se utiliza una memoria compartida, de manera de no realizar ejecuciones sincronizadas. Utilizando este lazo de control, permite que el elemento de control se acerque más a un dispositivo en tiempo real. Sin embargo, hay que considerar que el calefactor está conectado a un contactador y a un relé, ambos electromecánicos, por lo tanto disminuir el tiempo del ciclo de control produciría un mayor desgaste en los componentes mecánicos de ambos componentes eléctricos.

A pesar de todas estas posibilidades, sólo se probó un esquema de control, dado que la intención es mostrar el sistema en conjunto y no optimizar el funcionamiento del controlador. Debido a esto, algunas características del controlador están aún en marcha blanca, y algunos problemas pueden aparecer, como por ejemplo, problemas de comunicación entre la base de datos y el elemento de control originaría una baja considerable en el desempeño del controlador debido al aumento considerable del tiempo del ciclo de control (alrededor de 20 segundos por ciclo), problemas que pueden ser solucionados al agregar un temporizador que interrumpa la conexión a la base de datos si no hay comunicación en un cierto tiempo, por ejemplo, 2 segundos.

Otro problema importante es que el sistema no conoce el estado real de los actuadores, ya que no existe una retroalimentación del sistema que indique que luego de una modificación de una variable de entrada cual es el estado que quedó, lo que originaría que en caso de problemas de conexión entre el proceso y el elemento de control, éste último seguiría realizando el ciclo de control sin percatarse de dicho problema, y con ello originar daños en el equipo, como por ejemplo, el caso de encender el calefactor estando la bomba apagada.

Además de lo anterior, es necesario mejorar la conexión entre el sensor de flujo y el elemento de control debido a que las mediciones realizadas no son exactas. Inicialmente, la medición del sensor de flujo lo realiza el Arduino Mega 2560 a través del un programa que cuenta la frecuencia a través de interrupciones, sin embargo, al utilizar el Arduino Yún estas interrupciones causan problemas de comunicación entre el procesador y el microcontrolador, y con ello una detención en ciclo de control. Así que, para tomar esta medición se implementó una función que mide el periodo de la señal de entrada, sin embargo, esta medición posee una baja exactitud y además la relación entre el periodo medido y el flujo real no es linealmente proporcional. Para solucionar es recomendable conectar el sensor a un contador de frecuencia, y con ello medir de forma directa la magnitud entregada por el sensor.

5.2.2. Software

Una de las ventajas del sistema es la utilización de HTML para la construcción del HMI, ya que esto permite que el sistema de monitoreo y control pueda ser utilizado en cualquier computador o dispositivo móvil, sin la necesidad de instalar algún de tipo de programa. Sin embargo, podría existir problemas de seguridad en caso que la red en la cual el sistema de control esté al acceso de cualquier persona, como por ejemplo, que la red que conecta el sistema de control sea la misma que entregue Internet, ya que conectándose al servidor web que soporta el HMI le permitirá abrir y modificar parámetros, y así controlar y obtener información del proceso.

Para solucionar este problema de seguridad, hay algunas soluciones:

- Utilizar la red única y exclusivamente para el sistema de control, y poder acceder a él mediante una puerta de enlace (gateway) para conectarla a una red de uso publico restringiendo el ingreso por usuario o dirección IP.
- Agregar un módulo donde el acceso al HMI requiera utilizar un sistema de identificación de usuario a través un sitio web cada vez que desee ingresar al servidor.

Si bien, la segunda solución es más barata, pero un poco más complicada de implementar, ya que requiere escribir en HTML y PHP más códigos para desarrollar las páginas de acceso; en cambio, la primera solución es más segura, ya que la red es de uso único del sistema lo que mejora la velocidad de respuesta entre los elementos de control y el servidor de control, pero esta solución es más costosa debido a que requiere un router que cree la red y un dispositivo que actúe como gateway, y permita comunicarse con la red del sistema de control. Fuera de eso, el sistema está pensado tener una red aparte, debido a los temas mencionados anteriormente y sólo para la implementación se realizó en la red inalámbrica disponible en el laboratorio.

Ahora, si por algún motivo se desea cambiar el controlador debido a la manera en que está construido el sistema, no debería existir grandes cambios. Por el lado de la base de datos, es necesario agregar estos parámetros como componentes del sistema. Luego, es necesario cambiar el menú que se despliega en el HMI. Para ello, es necesario crear el menú en el archivo *getParam.php*, para que muestre las nuevas entradas para modificar los parámetros

del controlador, y en el archivo principal del equipo, hl630.php, cambiar el nombre del componente que envía al apretar el botón del controlador.

Además, se puede realizar una anulación del controlador del sistema, y ser reemplazado por un controlador externo y remoto, como por ejemplo, utilizando Simulink de Matlab. Para esto, es necesario dejar el controlador del sistema en modo manual y mediante el uso de REST API, conectarse al elemento de control que posee el controlador a reemplazar, el cual debe estar conectado directamente a la misma red de control, y utilizar el usuario y contraseña utilizada para conectarse al Arduino. De esta forma, se puede modificar todos los actuadores conectados a este elemento de control mediante la conexión de la url *ipArduino/arduino/setParam/x*, y medir a través de todos los sensores conectados al elemento de control por medio de la dirección *ipArduino/arduino/measureAll/1*, donde para ambos casos *ipArduino* es la dirección IP entregada al elemento de control. De esta manera, se puede diseñar, probar y estudiar controladores más complejos, como sistemas basados en modelos, redes neuronales, lógica difusa, entre otros.

5.2.3. Sintonización

Como se nombró en la Sección 4.3.3, se utilizó un modelo empírico para obtener un modelo del sistema y con ello sintonizar el controlador. Si analizamos el experimento realizado para obtener el modelo, por equilibrio termodinámico existe un único estado estacionario para el sistema en lazo abierto, bajo las condiciones descritas en la Sección 3.1.3, el cual corresponde a que la temperatura medida sea la temperatura ambiente. Si se realiza en otro momento esta prueba, digamos con otra temperatura ambiental, la respuesta sería similar debido a que la pérdida de calor en el radiador y en las cañerías es proporcional a la diferencia de temperatura entre el fluido y el ambiente, por lo tanto un pulso de energía de la misma duración entregaría una misma cantidad de calor, por lo tanto la diferencia de temperatura con respecto al estado estacionario sería similar, y ya que la velocidad de transferencia de calor sería la misma (porque el gradiente con el exterior es el mismo), entonces la curva sería similar. Cabe destacar que esto se debe a que el calor específico del agua en estas condiciones es prácticamente constante, así como también la densidad y viscosidad que producen que el flujo dentro del sistema sea parecido.

En el desarrollo del modelo empírico se determinó que una buena función de transferencia que describe el sistema corresponde a aquella de orden-(3,2), ya que la pendiente de la respuesta es mayor que cero, y además esta función de transferencia describe de buena manera el sistema, ya que presenta la misma oscilación, la cual se observa sólo cuando el fluido pasa a través del radiador. Esto se puede deber a 2 puntos:

- La presencia del radiador produce un aumento en la distancia que debe recorrer el fluido, lo produce una gran pérdida de carga, y esto conlleva a una disminución del flujo. Además, el paso a través del radiador también produce una mayor pérdida de calor en relación a que el fluido no pase por él. Debido a esto, el fluido se caliente de forma poco homogénea originando que existan zonas más calientes, y que, al recorrer el circuito producen la oscilación en la temperatura medida. De lo contrario, cuando el

fluido no pasa a través del radiador, la temperatura es más homogénea y asi, la curva obtenida es más suave.

- El efecto del radiador produce cambios más drásticos de temperatura, ya que cuando el fluido no fluye a través de él, la única pérdida de calor es por medido de la transferencia de calor a lo largo de los tubos que conducen el agua, por lo tanto caliente no se enfría tanto al recorrer el circuito, y como se mencionó anteriormente produce que la temperatura se caliente y se enfríe de forma más gradual.

Cómo se observa en el experimento realizado para conocer cualitativamente la respuesta del sistema frente a cambios en las variables de entrada (Ver Figura 4.11), se comprobó que la oscilación mostrada en la Figura 4.13, sólo ocurre cuando el flujo de agua pasa a través del radiador (caso 1-3) y no así cuando el agua no circula a través de esta (5-6). Esto puede ser explicado por distintos factores:

- Al cerrar el circuito de agua a través del radiador aumenta la cantidad de agua del sistema, ya que aumenta el largo del circuito y existe agua al interior del radiador, por lo tanto a un mismo flujo de agua, aumenta el tiempo característico (cuociente entre volumen de agua y flujo).
- El paso del agua a través del radiador produce una pérdida de carga en el sistema, ya que hace más tortuoso el paso del agua en relación a cerrar el ciclo a través de tubos. Es por ello que a una misma potencia de la bomba se produce una disminución de flujo en el sistema al estar conectado al radiador, y con ello se potencia más el punto mencionado anteriormente.
- Al utilizar el radiador la cantidad de calor transferida al ambiente es mayor, lo cual no ocurre homogéneamente a diferencia del caso sin radiador.

Bajo estas suposiciones, al aplicar un pulso al sistema cuando el radiador está desconectado del circuito ocurre que el líquido se calienta de una forma más homogénea debido a que se calienta en un punto pero es enfría a lo largo del recorrido del agua. Por otro lado, al estar conectado el radiador, el sensor recibe inicialmente una sección de agua que pasó solo por el calefactor por lo tanto se encuentra más caliente que el resto, luego un flujo que pasó por el radiador y el calefactor que se encuentra a una temperatura menor,y estas oscilaciones continúan pero con una menor amplitud. Esta disminución puede ser explicado porque el tiempo característico aumenta, lo que produce que estos líquido no pasen nuevamente por el calefactor encendido, disminuyendo la transferencia de calor, y a una mala mezcla al interior del circuito que no permite transferir calor entre distintas secciones del agua a través del avance por el circuito.

Frente a esto, es estima que es necesario tener un modelo fenomenológico o empírico del sistema, de manera de conocer bien su comportamiento y respuesta frente a perturbaciones y cambios en las variables de entrada, lo cual afectará notoriamente el desempeño del controlador. Sin embargo, la utilización de un modelo, ya sea empírico o fenomenológico no sirve para la sintonización mediante métodos clásicos para el controlador implementado, ya que el efecto de la modulación de ancho de pulso (PWM) afecta en el desempeño del controlador, como por ejemplo el factor desestabilizante producto del ciclo de encendido y apagado del actuador. Un método de sintonización no aplicado en este proyecto corresponde

a analizar la respuesta del sistema para un controlador de tipo on/off [10] al aplicar una entrada oscilatoria de periodo τ_n, de forma de obtener la razón entre la amplitud de entrada vs la amplitud de salida, con ello se define P_u como:

$$P_u = 100\frac{\pi}{4}\frac{A_c}{A_u} \tag{5.1}$$

Donde, A_c es la amplitud en la variable de entrada y A_u es la amplitud en la variable de salida. Con ello se utiliza la Tabla 5.1, para sintonizar el controlador. Donde, PID_i corresponde a un controlador de tipo interactivo y PID_n es de tipo no interactivo [10].

Tabla 5.1: Regla de sintonización usando ciclos proporcionales [10].

Controlador	P	τ_I	τ_D
PI	$1{,}70P_u$	$0{,}81\tau_n$	
PID_n	$1{,}30P_u$	$0{,}48\tau_n$	$0{,}11\tau_n$
PID_i	$1{,}80P_u$	$0{,}38\tau_n$	$0{,}14\tau_n$

Se decidió analizar como afecta el tamaño del período de la señal de referencia del modulador de ancho de banda τ_{PWM}. En la Figura 4.19 se observa una comparación entre 2 respuestas del controlador, utilizando los mismos parámetros, pero cambiando el valor de P_{PWM} de 10 a 20 segundos. Dado que el ciclo de control tiene el mismo tiempo (máximo 2 segundo), al tener un periodo de 10 segundos la variable manipulable se puede discretizar en 6 valores (0, 1/5,..., 1) y en el caso contrario, al aumentar este ciclo a 20 segundos, la variable manipulable se discretiza en 11 valores (0, 1/10, ... , 1); por lo tanto es esperable que mientras mayor es el tiempo del período mejor será el desempeño del controlador debido a que la diferencia entre la potencia del controlador PID continuo versus la salida de tipo ON/OFF sea menor, y con ello la respuesta será menos oscilatoria.

Sin embargo, debido al tipo de modulador empleado (ver Anexos G) , el uso de altos periodos favorece la aparición de armónicos similar a la onda de referencia ya que al inicio favorece al estado OFF y al final del periodo al ON, lo que se observó experimentalmente en la Figura 4.19. Por otro lado, en la misma experiencia se observó una disminución de la frecuencia de cambios de estados del elemento final de control al aumentar el periodo de la señal de referencia, por lo tanto, aumentar este valor disminuye la cantidad de cambios de estados del relé, y con ello la vida útil de este componente.

Al igual que el caso anterior, se vio que el valor límite de la suma de los errores acumulador $I_{max} = (\sum_{\text{i}=1}^{k-2} \varepsilon(\text{i})$ afecta al sistema, debido a que si este valor es muy alto genera mayor número de oscilaciones, ya que aumenta el efecto proporcional del controlador, pero disminuyen el efecto integral. Esto se observa en la Figura 4.20, donde se aplican los mismos parámetros excepto para el valor de I_{max}, el cual para el primer caso es 1000 y en el segundo 100. Incluso, la disminución de este valor casi elimina el efecto integral debido al alto tiempo de estabilización e incluso no mostrar indicios de eliminar el offset.

Capítulo 6

Conclusiones

Un sistema de control distribuido corresponde a un sistema donde los lazos de control de un proceso no dependen de un único controlador, sino que se encuentran repartidos a lo largo del proceso y se comunican a través de una red de control. Esta memoria permitió diseñar e implementar una infraestructura para sistemas similares al descrito anteriormente, utilizando componentes embebidos y de bajo costo, incluyendo la utilización de una red inalámbrica para conectar dichos componentes y permitir la manipulación del sistema de forma remota, y con ello cumplir el objetivo general del proyecto.

El diseño del sistema se basó en una simplificación de un sistema de control distribuido, donde se redujo componentes que no son necesarios dado la escala del sistema, así como también se sustituyeron componentes, produciendo una disminución de la eficiencia y robustez del sistema, como por ejemplo utilizar una red inalámbrica de uso doméstico frente al empleo de un *Data Highway*, como por ejemplo fibra óptica.

De la misma forma, se utilizaron dispositivos como Arduino y Raspberry Pi, como componentes del sistema debido a su amplio uso, bajo costo y gran versatilidad que permite reemplazar a un controlador industrial o un PLC, sin embargo, estos componentes no poseen las características físicas tal que permitan ser utilizados en la industria, por ejemplo, resistencia a condiciones más agresivas, como vibraciones, temperatura y humedad.

Por otro lado, fue necesario utilizar herramientas como bases de datos, lo que permitió almacenar de forma eficiente las mediciones y registros del sistema de control, apoyado por la normalización de tablas, y con ello estandarizar el manejo de la información a través de lenguaje SQL, en los distintos componentes que realizan consultas o inserción de datos.

Así también, se diseñó el interfaz humano-máquina como una aplicación web mediante el uso de herramientas como AJAX, HighChart/HighStock y PHPExcel, lo que quita la necesidad de instaladores, elimina problemas de compatibilidad en distintos sistemas operativos y con ello, mejora la versatilidad lo que permite monitorear el sistema incluso desde un dispositivo móvil, y otorga características como la visualización de datos registrados en líneas de tiempo o la exportación de datos a hojas de cálculo.

Este sistema se implementó en el equipo HL630, que contiene un circuito de refrigeración compuesto por un radiador con ventilador y un calefactor, el cual incluye un controlador, el que fue reemplazado por el elemento de control, ya que el controlador que incluye el equipo no permite ser manipulado de forma remota. Así también, se conectó los sensores y actuadores a través de circuitos o dispositivos electrónicos como relés, lo que permitió conectar el sistema de control al equipo de forma satisfactoria a través del elemento de control, y con esto, manipular variables de entrada a través del HMI, así como también modificar parámetros del controlador.

Por motivos operacionales no es posible realizar el método de la curva de reacción, por lo que fue necesario obtener un modelo del sistema para su sintonización. Se planteó un modelo fenomenológico, el cual no fue implementado por limitantes temporales, y por esto, se realizó un modelo empírico, el cual presentó un buen grado de ajuste con respecto a la prueba experimental realizada.

Con esto, la sintonización del controlador se realizó a través de un análisis de sensibilidad, tomando como referencia valores obtenidos mediante la optimización de la respuesta del sistema frente al problema de regulación y servo-control entregados por PID Tuner, que utilizó una aproximación continua del controlador. Debido a lo anterior, estos valores no pudieron ser utilizados directamente, ya que experimentalmente se observó gran inestabilidad del sistema, producida por el bajo valor del parámetro τ_I.

Además, para probar el sistema se realizaron 2 pruebas de sensibilidad con 2 parámetros: el periodo de la señal de referencia P_{PWM} y el valor máximo del error acumulado I_{max}, destinado al uso del anti-reset windup. En el primer caso, se observó que al aumentar al doble este periodo no afecta la eficacia del controlador, pero se aumenta el efecto de armónicos en la respuesta, lo que se observa como una oscilación constante en la salida, pero a su vez, produce una disminución en la frecuencia de cambios de estados del relé, aumentando su vida útil. De forma contraria, al disminuir el valor de I_max disminuye la frecuencia y la amplitud de las oscilaciones, pero aumenta el tiempo de estabilización a estado estacionario e incluso se aleja del control perfecto al observarse la presencia de un offset.

Debido a sus características, este sistema permite ser utilizado en aplicaciones de automatización domésticas, como por ejemplo sistemas de regadíos automáticos, donde el elemento de control se conectaría a sensores de humedad en el terreno y con ello abrirían o cerrarían válvulas para abrir los aspersores que riegan un prado, lo está integrado al sistema de control, el cual incluye el HMI y el registro en la base de datos. Así, este sistema permitiría gobernar toda una casa, y con ello controlar sistemas de luces, alarmas, calefacción, entre otros desde cualquier computador o dispositivo móvil.

En el ámbito de laboratorio, este sistema permitiría crear un sistema unificado para la recolección de datos y con ello extraer desde cualquier computador conectado a la red del laboratorio los datos medidos, y además poder conectarse de forma remota al sistema de control para manipular cualquier equipo conectado al sistema. Incluso, como se mencionó en el cuerpo de este documento, este sistema permitiría crear lazos de control y con ello automatizar equipos que no traen incorporado este tipo de mecanismos, así como también utilizar algoritmos de control más complejos.

6.1. Recomendaciones

A continuación de detallan algunos puntos importantes que pueden ser desarrollados o estudiados de manera de mejorar el desempeño, la seguridad y agregar características al sistema.

6.1.1. Implementación

- **Continuar marcha blanca del sistema**:
  Se recomienda fuertemente, operar el equipo bajo vigilancia para encontrar errores que produzcan una disminución del desempeño del controlador o fallas de seguridad, ya sea por intrusión de usuarios no deseados o problemas que pongan en riesgo la operación del equipo.
- **Utilizar una red exclusiva para el sistema de control**:
  Esto mejora la seguridad del sistema, ya que restringe el acceso al servidor web desde una red externa a través de una puerta de enlace (gateway) con muro cortafuegos (firewall), además de eliminar posibles problemas de latencia en la comunicación entre los componentes del sistema.
- **Crear un sistema de registro de acceso**:
  Utilizar una registro de acceso permite que solo personas autorizadas permitan ingresar al HMI, y con ello se puede restringir el acceso total, o dejar en modo de solo lectura, o también, tener acceso solo a algunos equipos conectados al sistema.
- **Utilizar un contador de frecuencia para el sensor de flujo**:
  Esta solución mejoraría la medición de flujo, debido a que Arduino Yún presenta problemas al realizar esta medición y actualmente realiza una medición indirecta que provoca que los resultados sean poco exactos.
- **Modificar la estructura del controlador**:
  Como se vio en la sección de discusiones, como el Arduino Yún posee un microcontrolador y un procesador, se pueden probar distintas configuraciones sobre quién realiza el algoritmo de control, por ejemplo, ejecutar el ciclo de control y de monitoreo de forma separada y corriendo en paralelo en el procesador; o realizar el ciclo de control en el microcontrolador y el de monitoreo en el procesador.
- **Creación de software para la elaboración más amigable del HMI**:
  Actualmente el HMI se elabora a través de código HTML, Javascript y PHP para la elaboración del interfaz que muestra el proceso, ya que requiere seleccionar una imagen, ubicar botones, bloques de texto y que interactuen con el usuario. Se podría crear un programa que permita construir esto de forma gráfica y que sea más amigable con el usuario.

6.1.2. Controlador

- **Crear una biblioteca para manipulación externa**:
  La creación de la biblioteca mejoraría la prestación del sistema, y se recomienda crearla en un software externo, por ejemplo Matlab, para realizar el controlador desde un computador remoto y con ello, probar y estudiar otros tipos de controladores.
- **Realizar más pruebas para la sintonización del controlador**:
  Utilizar otros métodos de sintonización que puedan mejorar el desempeño, por ejemplo disminuir la oscilación, o disminuir la velocidad de respuesta.

Bibliografía

[1] Arduino. Arduino - arduinoboardmega2560. [En línea] `http://arduino.cc/en/Main/ArduinoBoardMega2560`. [consulta: 04 de agosto de 2014].

[2] Arduino. Arduino - arduinoboardyun. [En línea] `http://arduino.cc/en/Main/ArduinoBoardYun`. [consulta: 04 de agosto de 2014].

[3] Arduino. Arduino - arduinowifishield. [En línea] `http://arduino.cc/en/Main/ArduinoWiFiShield`. [consulta: 04 de agosto de 2014].

[4] Arduino. Arduino yún in arduino store. [En línea] `http://store.arduino.cc/index.php?main_page=product_info&products_code=A000008`. [consulta: 08 de marzo de 2015].

[5] Siemens Industry Automation. Easy profinet implementation. SIEMENS, 2012.

[6] Pablo Gonzalez Carlos Espinosa. *Manual para la realización de PCBs*. Laboratorio de mecatronica, Universidad de Chile, 2.01 edition, 2005.

[7] LitePoint Corporation. *IEEE 802.11ac: What Does it Mean for Test?* LitePoint, 2013.

[8] ENERGIT Electrónica de Potencia S.A. Capítulo 6: Filtro de linea-en qué consisten, de qué y cómo protengen.

[9] Element14. Raspberry pi b+ in element14.com. [En línea] `http://www.element14.com/community/community/raspberry-pi/raspberry-pi-bplus`. [consulta: 08 de marzo de 2015].

[10] Don W. Green and Robert H. Perry. *Perry's Chemical Engineers' Handbook*. McGraw-Hill, 8th edition, 2008.

[11] E. Heinzle J. E. Prenosil J. B. Snape J. Ingham, I. J. Dunn. *Chemical Engineering Dynamics: An Introduction to Modelling and Computer Simulation*. WILEY-VCH Verlag GmbH & Co. KGaA, 3rd edition, 2007.

[12] Kobold. Modelo dpl medidor de caudal tipo rotativo de bajo caudal para líquidos.

[13] Béla G. Lipták. *Instrument Engineers' Handbook: Process Contol and Optimization, Vol 2*. CRC Press, 4st edition, 2005.

[14] Microensamble. Calculadora de ancho de pista vs corriente. [En línea] `http://www.microensamble.com/site/index.php?option=com_content&view=article&id=105:calculadora-de-ancho-de-pista&catid=62&Itemid=197`. [consulta: 06 de septiembre de 2014].

[15] Babatunde A. OGUNNAIKE and W. Harmon RAY. *Process Dynamics, Modeling and Control*. Oxford University Press, 1st edition, 1994.

[16] Pförtner. *Transmitter for Mounting Rails for Thermocouples and Resistance thermometers: Datasheet 84.9*. Pförtner Messtechnik GmbH & Co. KG, 1.5 edition, 2009.

[17] Prozeß-und Maschinen-Automation GmbH PMA. Ks40, compact industrial controller.

[18] Raspberry. Raspberry pi 1 model b+. [En línea] `http://www.raspberrypi.org/products/model-b-plus/`. [consulta: 08 de marzo de 2015].

[19] Siemens AG Industry Sector. *Industrial Wireless Communication*. Siemens AG, 2012.

[20] Carlos A. Smith and Armando B. Corripio. *Control Automático de Procesos: Teoría y Práctica*. Editorial Limusa S.A., 1st edition, 1991.

[21] Richard Steigmann and Jan Endresen. Introduction to wisa - wireless interface for sensor and actuators. ABB Corporate Research, July 2006.

[22] George STEPHANOPOULOS. *Chemical Process Control: An Introduction to Theory and Practice*. Prentice Hall International, 1st edition, 1984.

[23] Jian Sun. *Dynamics and COntrol of Switched Electronic Systems, Chapter 2: Pulse-Width Modulation*. Springer London, 1st edition, 2012.

[24] Luis Vargas. *Apuntes de electromagnétismo*. Departamento de Ingeniería eléctrica, Facultad de Ciencias Físicas y Matemáticas, 2rd edition, 2010.

Anexos

Anexo A

Comparación de precios

Las Tablas A.1, A.2 y A.3 muestran los distintos costos de hardware y software para PLC, un sistema de control distribuido (DCS) y los principales componentes utilizados.

Tabla A.1: Costo de PLC y componentes asociados [13].

	Characteristics	Price Range ($US)
Nano PLC Hardware	Basic PLC with CPU, 12 I/O points	99 to 200
Small PLC Hardware	Basic PLC with CPU, 2-Kb RAM, 20-40 I/O points	200 to 1,000
Medium PLC Hardware	Basic System with 256 I/O points	5,000 to 8000
Medium PLC Hardware	Basic System with 512 I/O points	10,000 to 20,000
Medium PLC Hardware	Basic System with 1024 I/O points	14,000 to 28,000
Large PLC Hardware	cost driven by network and information display requirements	15,000 to 70,000
Software		50-100 % of hardware cost
Engineering		25-50 % of hardware cost
Handheld programmers		100 to 500
PC-supported programming software		2,500 to 5,000

Tabla A.2: Costo de componentes de un DCS sin considerar software [13].

Description	Price Range ($US)	Typical ($US)
Control unit, w/carrier, power supply	5300 to 6600	5500
Redundant control units	11,700 to 14,300	13,000
Operator's console w/21"monitor (1200 I/O)	9,800 to 10,300	10,000
Engineer's station w/21"monitor (1200 I/O)	20,000 to 22,000	21,000
Analog input (AI) card, 8 channel w/HART	1,400 to 1,800	1,500
Analog input (AI) card, 8 channel w/HART redundant	2,500 to 3,000	2,800
Analog output, 4 to 20 mA, 8 channel w/HART	1,700 to 1,900	1800
Analog output, 4 to 20 mA, 8 channel w/HART, redundant	2,700 to 3,300	3000
Foundation Fieldbus interface card w/2 segments	2,200 to 2,900	2500
Discrete input, 24 Vdc, 8 channel	520 to 630	550
Discrete output, 24 Vdc, 8 channel	1,000 to 1,150	1,100
Discrete input, 24 Vdc, 8 channel as fieldbus I/O	820 to 930	850
Discrete output, 24 Vdc, 8 channel as fieldbus I/O	1,000 to 1,150	1000
Discrete input, 24 Vdc, dry contact, 8 channel, redundant	1,000 to 1,200	1,100
Discrete output, 24 Vdc, hi-side, 8 channel, redundant	1,500 to 1,700	1,600
Carrier for 8 cards	600 to 800	700
Bulk power supply	900 to 1,200	1,000

Tabla A.3: Costo de componentes utilizados para el sistema implementado.

Componente	Precio ($US)	Referencia
Arduino Yún	57	[4]
Raspberry Pi B+	30	[9]

Anexo B

Conceptos básicos

B.1. Tipos de variables

Si se representa un proceso genérico como el que se observa en la Figura B.1, éste se puede describir como un mecanismo, equipo o *caja* donde, por un extremo entra materia prima y energía, y por el otro lado salen los productos y subproductos.

Figura B.1: Representación de un proceso genérico.

Imaginemos que esta caja corresponde a una caldera, la que será utilizada como ejemplo para las definiciones de proceso y los distintos tipos variables. La materia prima de este sistema corresponde al combustible, el comburente (por ejemplo, aire) y el agua que será transformada en vapor. Por otro lado, el producto será el vapor generado y como subproductos los gases de combustión. Esta transformación de agua en vapor no ocurre espontáneamente, ya que a partir de balances de masa y energía se determina cuánto es el combustible y comburente necesario para que la combustión se realice de la forma más completa posible, y por lo tanto, menos contaminante y más segura. Sin embargo, todos estas relaciones requieren que se fijen algunos valores para que sean completamente descritas, y con ello, poder resolverlas. Para determinar la cantidad de variables que se quieren fijar es necesario realizar un análisis de grados de libertad.

Estas materias primas, productos y subproductos deben ser descritas mediante variables, como por ejemplo la temperatura y la composición de los gases de salida, flujo de combustible, entre otros. Estas variables se pueden describir como de entrada o salida dentro proceso, donde aquellas que afectan al proceso se denominan variables de entrada; en cambio aquellas que se generan debido al actuar del proceso se denominan variables de salida.

En particular, para la caldera descrita anteriormente, tenemos las siguientes variables:

- Flujo, presión y temperatura del aire (F_A, P_A y T_A) y del combustible (F_c, P_c y T_c).
- Flujo, presión y temperatura del agua de alimentación (F_w, P_w y T_w).
- Presión y temperatura de operación de la caldera (P y T).
- Composición, presión y temperatura de los gases de combustión (x_{i}, P_g y T_g).
- Flujo, presión y temperatura del vapor generado (F_v, P_v y T_v).

Muchas de estas variables pueden ser descritas a partir del proceso, ya que la inclusión de un lazo de control produce que variables de salida se conviertan en variables de entrada, como por ejemplo un lazo que controle la presión del vapor que es generado por la caldera produce que esta variable no sea de salida, sino que de entrada, ya que el valor que toma la variable afectará el estado del proceso.

Tabla B.1: Categorización de variables como de entrada y salida

Tipo	Variables								
Variable de entrada	F_A	P_A	T_A	F_c	P_c	T_c	F_w	P_w	T_w
Variable de salida	P	T	x_{i}	P_g	T_g	F_v	P_v	T_v	

En este caso particular, la presión y temperatura de la caldera vienen dados por el balance de masa y energía del sistema.

Además de esta división, las variables de entrada se pueden catalogar como manipulables y perturbaciones [22]:

- Variables manipulables: Son las variables de entrada que su valor puede ser ajustado libremente por la acción humana o un mecanismo de control.
- Perturbación: Son aquellas variables de entrada que no pueden ser ajustadas y pueden ser medibles o no medibles.

Por otro lado, las variables de salida se pueden catalogar como [22]:

- Variable de salida medible: Son aquellas que se pueden medir directamente.
- Variable de salida no medible: Por el contrario, son aquellas que no se pueden medir.

Otro término que se utiliza con frecuencia es el de variable de desviación (x'), el cual para una variable x, se define como:

$$x' = x - x_{ss}$$

Donde x_{ss} corresponde al valor de la variable x en estado estacionario.

B.2. Modelamiento de procesos

Conocer el proceso es una buena estrategia para elegir un controlador apropiado, así como también para elegir el valor de parámetros más apropiado, tema que será visto más adelante. Existen varias formas de representar un proceso, como por ejemplo, mediante el uso de una o más ecuaciones de distinto tipo, como sistemas algebraicos o sistemas de ecuaciones diferenciales, así como también sistemas más complejos como redes neuronales. Esta representación o abstracción de la realidad se le denomina modelo matemático. Dependiendo de la finalidad del modelo, será definida la complejidad que se quiera dar. De esta forma, un modelo más complejo puede mostrar más fidedignamente el comportamiento del proceso, pero esto, a su vez, requiere un mayor gasto de tiempo y recursos para desarrollarlo.

A partir de distintos fenómenos que ocurren en el proceso, se pueden obtener modelos, éstos son llamados fenomenológicos o teóricos, los cuales requieren un profundo conocimiento del sistema. Por otro lado, los modelos empíricos son aquellos que se forman a partir de la observación y la experiencia, si bien no poseen un sustento teórico como el caso anterior, pueden llegar a representar de buena forma un modelo bajo ciertas condiciones y supuestos, e incluso tener una forma más simple que un modelo fenomenológico.

Para describir un proceso o un sistema y su comportamiento se requiere de [22]:

- Una serie de variables cuyos valores describen el estado natural de un sistema.
- Una serie de ecuaciones con las variables recién nombradas, las cuales describen cómo cambia el estado natural de un sistema en el tiempo.

A las variables antes mencionadas se les denomina variables de estado, y a este set de ecuaciones se les llama ecuaciones de estado. Para definir estas variables y ecuaciones se utilizan relaciones de masa, energía y momemtum. Estas relaciones se pueden describir en variables como la densidad, concentración, temperatura, presión y flujo. La obtención de ecuaciones del sistema se puede determinar a partir del principio de conservación (Ver Ecuación B.1) y con ello realizar balances de masa global o por especie, balances de energía y de momento. Otra fuente de ecuaciones son las relaciones usadas para expresar equilibrios termodinámicos, velocidades de reacción, tasas de transporte de masa, energía y momento. Otros fenómenos, como tiempos muertos, o el tiempo que tarda en pasar una sustancia por una tubería se pueden modelar como un retardo, fenómeno que trae algunos inconvenientes en la elección del controlador [22].

$$\text{Acumulacion} = \text{Entrada} - \text{Salida} + \text{Generacion} - \text{Consumo} \tag{B.1}$$

Para describir el modelo matemático, se puede definir de 4 maneras:

1. En el estado de espacios (state-space).
2. En el dominio de las transformadas (de Laplace o transformada-z).

3. En respuesta de frecuencias (o variable compleja).
4. En respuesta de impulso (o convolución).

Todas estas formas están relacionadas y para pasar de una a otra requiere de una transformación o cálculo (Ver Figura B.2). En este documento, todo el análisis dinámico será descrito en el dominio de las transformadas.

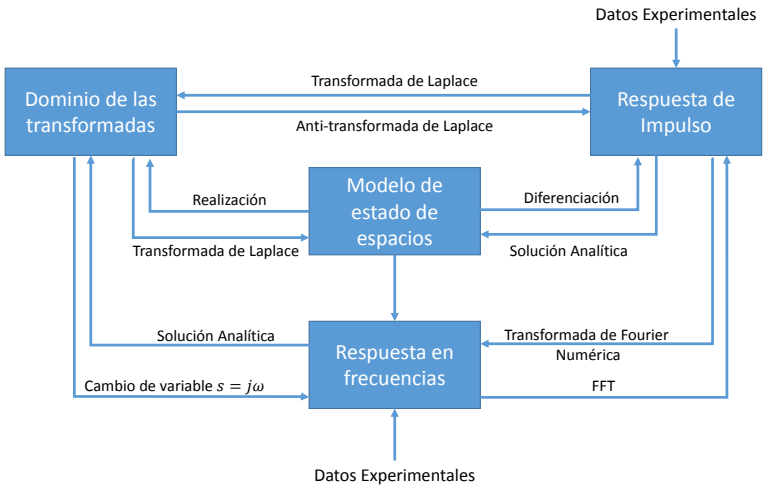


Figura B.2: Interacción entre tipos de representaciones de un modelo [15].

B.3. Transformada de Laplace

La utilización de la transformada de Laplace es una herramienta que permite resolver fácilmente EDOs Lineales. La transformada de Laplace $\bar{f}(s)$ de la función $f(t)$ se define como:

$$\mathcal{L}\left[f(t)\right] = \bar{f}(s) = \int_0^\infty f(t)\mathrm{e}^{-st}\mathrm{d}t$$

Algunas propiedades para la transformada de Laplace son [15]:

1. No todas las funciones $f(t)$ poseen transformada de Laplace. Esto se debe a que la función debe ser finita para que la transformada exista.
2. La transformada de Laplace no contiene información sobre el comportamiento de $f(t)$ para $t < 0$ debido a que el límite de la integral es 0. Esto, sin embargo, no es un problema debido a que las aplicaciones en la Ingenería de Procesos generalmente se está interesado sobre lo que ocurre desde un tiempo igual a t_0, el cual puede ser cambiado arbitrariamente a cero.
3. La función $f(t)$ y su correspondiente transformada $\bar{f}(s)$ se denominan *transform pair*. Lo importante es que estos pares de funciones son únicas. Si una función $f(t)$ y $g(t)$ poseen la misma transformada de Laplace, entonces $f(t) = g(t)$.
4. La transformada de Laplace es una operación lineal, es decir:

$$\mathcal{L}\left[c_1 f_1(t) + c_2 f_2(t)\right] = c_1\mathcal{L}\left\{f_1(t)\right\} + c_2\mathcal{L}\left\{f_2(t)\right\} \tag{B.2}$$

La transformada inversa o antitransformada de Laplace corresponde a:

$$f(t) = \mathcal{L}^{-1}\left\{\bar{f}(s)\right\} = \frac{1}{s\pi j}\int_{\mathbb{C}} \mathrm{e}^{st}\bar{f}(s)\mathrm{d}s \tag{B.3}$$

Por definición, para $f(t) = 1$:

$$\bar{f}(s) = \mathcal{L}\{1\} = \int_0^t \mathrm{e}^{-st}\mathrm{d}t = -\left.\frac{\mathrm{e}^{-st}}{s}\right|_{t=0}^{t=\infty} = \frac{1}{s} \tag{B.4}$$

Para las derivadas, la transformada de Laplace para estas funciones es:

$$\mathcal{L}\left\{\frac{\mathrm{d}^n f}{\mathrm{d}t^n}\right\} = s^n\bar{f}(s) - \sum_{k=0}^{n-1} s^k f^{n-1-k}(0) \tag{B.5}$$

Donde, $f^{(\mathrm{i})}(0)$ se define como la i-ésima derivada de $f(t)$ evaluada en $t = 0$. Además, si las condiciones iniciales de $f(t)$ y todas sus derivadas son cero, la Ecuación B.5 es:

$$\mathcal{L}\left\{\frac{\mathrm{d}^n f}{\mathrm{d}t^n}\right\} = s^n\bar{f}(s) \tag{B.6}$$

La transformada de Laplace para integrales corresponde a:

$$\mathcal{L}\left\{\int_0^t f(t)\mathrm{d}t)\right\} = \frac{1}{s}\bar{f}(s) \tag{B.7}$$

Otra propiedad de la transforma de Laplace es el desplazamiento de la variable compleja, donde:

$$\mathcal{L}\left\{\mathrm{e}^{at}f(t)\right\} = \bar{f}(s-a) \tag{B.8}$$

De esta propiedad se desprende que:

$$\mathcal{L}^{-1}\left\{\frac{1}{s+a}\right\} = \mathrm{e}^{-at} \tag{B.9}$$

En cambio, si este desplazamiento es temporal:

$$\mathcal{L}\left\{f(t-a)\right\} = \mathrm{e}^{-as}\bar{f}(s) \tag{B.10}$$

Teorema del valor inicial

$$\lim_{t\to 0} f(t) = \lim_{s\to\infty} s\bar{f}(s) \tag{B.11}$$

Teorema del valor final

$$\lim_{t\to\infty} f(t) = \lim_{s\to 0} s\bar{f}(s) \tag{B.12}$$

Algunas de las transformadas más comunes son:

Tabla B.2: Lista de transformadas de Laplace más comunes.

Función	$f(t)$	$\mathcal{L}\{f(t)\}$
Retraso ideal	$\delta(t-\tau_\mathrm{d})$	$\mathrm{e}^{\tau_\mathrm{d}s}$
Impulso unitario	$\delta(t)$	1
Escalón unitario	$f(t) = \begin{cases} 0 & t<0 \\ 1 & t\geq 0 \end{cases}$	$\frac{1}{s}$
Pulso tamaño A	$f(t) = \begin{cases} 0 & t<0 \\ \frac{A}{b} & 0\leq t\leq b \\ 0 & t>b \end{cases}$	A
Seno	$f(t) = sin(\omega t)$	$\frac{\omega}{s^2+\omega^2}$
Coseno	$f(t) = cos(\omega t)$	$\frac{s}{s^2+\omega^2}$

Como se mencionó, la transformada de Laplace permite resolver EDOs lineales, por ejemplo, analicemos un proceso modelado con la siguiente expresión [15]:

$$\tau\frac{\mathrm{d}y}{\mathrm{d}t} + y = Ku(t)$$

Considerando $u(t)$ como un escalón unitario, luego aplicando la transformada de Laplace y sus propiedades, se obtiene:

$$\tau s\bar{y}'(s) + \bar{y}'(s) = \frac{K}{s}$$

Reordenando, se obtiene por solución:

$$\bar{y}'(s) = \frac{K}{(\tau s+1)}\frac{1}{s}$$

Utilizando fracciones parciales:

$$\bar{y}'(s) = \frac{K}{s} - \frac{K\tau}{\tau s+1}$$

Aplicando la antitransformada de Laplace a cada término:

$$y'(t) = K - K\mathrm{e}^{-\frac{t}{\tau}} = K\left(1-\mathrm{e}^{-\frac{t}{\tau}}\right)$$

B.4. Sistemas lineales y no lineales

Un sistema se dice lineal si es descrito por ecuaciones que sólo contienen funciones lineales. Por el contrario, si estas ecuaciones poseen algún término no lineal, será denominado como sistema no-lineal. Un sistema se define de primer orden si él es descrito con ecuaciones diferenciales de primer orden, así mismo, para un sistema descrito con una ecuación diferencial orden n-ésimo, se dice que es un sistema de n-ésimo orden.

Existen 2 propiedades básicas que caracterizan el comportamiento de un sistema lineal [15]:

1. **Principio de superposición**
   Si la respuesta de un proceso para una entrada I_1 en R_1 y la respuesta para una entrada I_2 es R_2, entonces según el principio de superposición, la respuesta a (I_1+I_2) es (R_1+R_2) si el sistema es lineal.
   En general, la respuesta para un sistema lineal con suma de N entradas será la suma de cada entrada de forma individual.

2. **Independencia entre la forma de la respuesta dinámica y las condiciones del proceso**.
   La forma de una respuesta frente a un cambio en la entrada es independiente al estado estacionario al momento en que se hizo el cambio, sólo si el sistema es lineal. Por ejemplo, para un mismo cambio escalón, la respuesta será la misma pero desplazadas según el estado estacionario inicial, antes de el cambio en la entrada. Por otro lado, para un una misma entrada pero con signo contrario, se obtendrá la misma respuesta pero en sentido distinto, similar a lo que se observa en la Figura B.3.

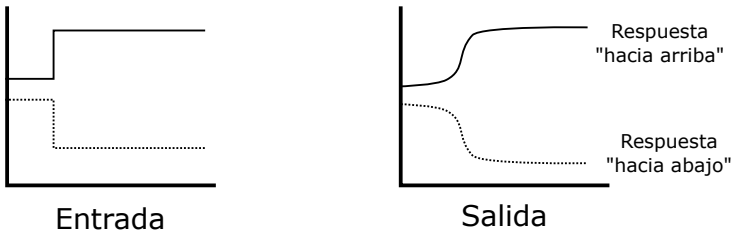


Figura B.3: Representación de sistema lineal al realizar el mismo cambio en la variable de entrada, pero de distinto signo [15].

Para analizar un sistema representado por un sistema no lineal, las siguientes son algunas estrategias que se pueden tomar [15]:

1. Utilizar una análisis analítico
2. Linealización exacta por cambio de variable
3. Análisis numérico
4. Linealización del sistema

Un análisis analítico puede ser utilizado sólo si el sistema puede ser resuelto de forma analítica. Si bien, son pocas las ecuaciones no lineales que pueden ser resueltas, también son pocos los métodos para analizar de forma cualitativa los resultados. Otra forma de analizar el sistema es mediante la transformación de términos no lineales en variables, de forma que el sistema quede como un sistema lineal. Sin embargo, no existe un procedimiento general para transformar cualquier sistema no lineal, por lo que usualmente no es una estrategia muy utilizada.

En general, para resolver un sistema con ecuaciones diferenciales, en especial para las ecuaciones de tipo no lineal, se obtiene una solución numérica, la cual no entrega una solución analítica, sino que a través de una simulación es posible observar curvas o tablas que contienen la solución.

Por otro lado, la linealización de un sistema se define como la aproximación de un sistema no lineal a través de un sistema lineal. La forma más común de realizar esta aproximación es a través de la serie de expansión de Taylor (Ver Ecuación B.13), la que al ser realizada en un primer orden se obtiene una aproximación lineal.

$$f(x) \sim f(x_s) + f'(x_s)(x - x_s) \tag{B.13}$$

En general, se utiliza como punto de expansión el valor en estado estacionario, sin embargo, existen otros métodos que permiten realizar la linealización por tramos, de forma que la esta aproximación sea más cercana a la respuesta del sistema original.

Se define la función de transferencia de un proceso como:

$$G(s) = \frac{\text{Transformada de Laplace de salidas}}{\text{Transformada de Laplace de entradas}}$$

Donde, la respuesta de un sistema con M entradas corresponde a:

$$y(t) = \sum_{\mathrm{i}=1}^{M} \mathcal{L}^{-1}\left\{G(s) \cdot \bar{f}'_{\mathrm{i}}(s)\right\}$$

B.5. Estrategias de Control

En algunos procesos, la variable controlada varía del setpoint a causa de perturbaciones. El término de regulación se refiere a los sistemas diseñados para compensar las perturbaciones. Sin embargo, cuando el cambio del setpoint afecta de mayor manera al sistema que las perturbaciones en sí, existen sistemas diseñados para contrarrestar el efecto donde el setpoint puede variar en el tiempo, a éstos se les denomina control servo. A pesar de esta diferencia, ambos sistemas pueden ser analizados mediante la misma metodología, así como también, elegir el tipo de controlador y su sintonización. La estrategia de control corresponde a la

forma en que el sistema afronta cambios en el proceso y con ello, la forma en que actúa el sistema.

Una estrategia de control de tipo Feedback corresponde a aquella donde, a partir de un cambio en una variable de salida medible, el sistema actúa sobre el sistema a través de una entrada manipulable (Ver Figura B.4). La ventaja de este tipo de estrategia es su simplicidad, ya que no requiere necesariamente un modelo o estimación de la salida del sistema, y además, es capaz de contrarrestar todas la perturbaciones que afectan al sistema. Sin embargo, la gran desventaja de este tipo de sistema de control es que el sistema no puede adelantarse a perturbaciones o cambios en las variables de entrada, ya que sólo actúa una vez que el sistema fue afectado. Pero el tiempo en que el sistema es afectado y vuelva al setpoint dependerá de las características del controlador, las cuales, a su vez, dependen del comportamiento dinámico del proceso.

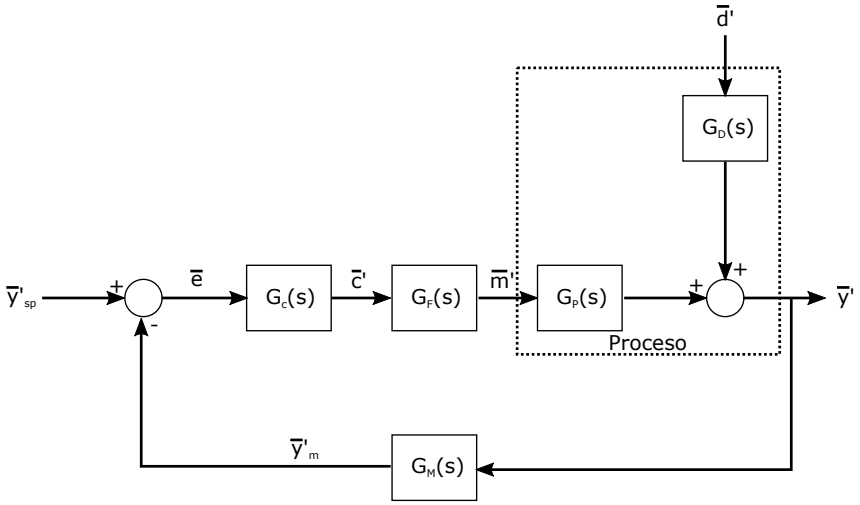


Figura B.4: Representación de un controlador de tipo Feedback [15].

Por otro lado, el sistema de control Feedforward es aquel que utiliza la medición de cambios de perturbaciones medibles y a partir de una estimación del proceso afecta a éste a través de variables manipulables (Ver Figura B.5). A pesar que este esquema requiere de un mayor conocimiento del sistema para realizar la acción precalculada, además de ser más rápido, ya que el sistema actúa antes que exista un efecto en la salida, tiene un gran inconveniente: esta estrategia de control no es capaz de enfrentar perturbaciones no medidas.

En búsqueda de un mejor controlador algunos sistemas de control mezclan estas 2 estrategias, de manera que la parte Feedforward enfrente a las perturbaciones medibles antes que afecten al proceso, y la parte Feedback actúe en aquellas que no fueron detectadas por la otra parte.

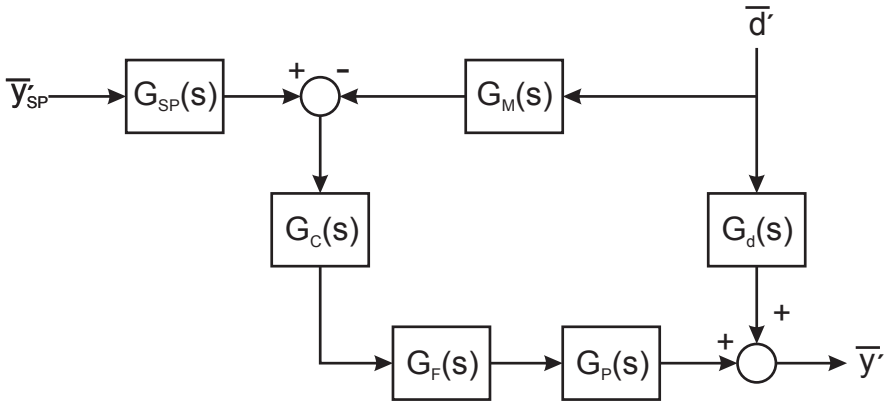


Figura B.5: Representación de un controlador de tipo Feedforward [15].

B.6. Estabilidad

Un aspecto muy importante del comportamiento dinámico de un proceso es cómo actúa el sistema luego de que ocurre un cambio en la variable de entrada de éste. Si el sistema se encontraba en estado estacionario antes de este cambio, el sistema tiende a responder de las siguientes formas:

- Pasa a un estado estacionario y se mantiene ahí.
- No se mantiene en estado estacionario, ya que su salida crece indefinidamente.
- No se mantiene en estado estacionario, ya que la respuesta oscila indefinidamente con una amplitud constante.

De estas respuestas, sólo la primera puede ser considerara como estable. Algunas definiciones más formales sobre estabilidad son [15]:

Definición 1 Un estado estacionario x_s se dice estable si para cada posible región de radio $\varepsilon > 0$ alrededor del estado estacionario, existe un estado inicial x_0 en $t = t_0$ dentro de una bola de radio $\delta > 0$ centrada en x_s, tal que la trayectoria del sistema x permanecerá en el interior de la región $\|x - x_s\| < \varepsilon \ \forall t > t_0$.

Definición 2 Un estado estacionario x_s se dice asintóticamente estable si es estable y además existe una bola de condiciones iniciales de radio $\delta_0 > 0$ centrada en x_s para la cual el sistema se aproxima a x_s para $t \to \infty$.

Definición 3 Un estado estacionario x_s se dice inestable si no es estable.

Definición 4 Un sistema dinámico se dice estable en un dominio M, si para toda región de radio $\varepsilon > 0$ en el dominio existe una región de radio $\delta > 0$ en el dominio tal que las trayectorias del sistema están acotadas en ε, si la condición está acotada en δ.

Definición 5 Un sistema se considera BIBO estable si para toda entrada acotada el sistema poduce una trayectoria dinámica acotada para $t \to \infty$.

Un sistema lineal será estable si y sólo si todos sus polos tienen parte real negativa, de otra forma será inestable [15]. Para un sistema no lineal, y para conocer si un estado estacionario es estable o inestable, es necesario analizar un modelo linealizado. Si el modelo linealizado indica estabilidad del estado estacionario, entonces el correspondiente sistema no lineal también será estable en un vecindario de este estado. De forma contraria, si el modelo linealizado muestra inestabilidad del estado estacionario, entonces el estado estacionario original del sistema no lineal también será inestable.

Para analizar la estabilidad de un sistema en lazo cerrado, es necesario analizar el sistema a través de la ecuación característica. Otras funciones permiten conocer la estabilidad del sistema como el criterio de estabilidad de Routh.

B.7. Tipos de controladores

La manera en que los controladores de tipo Feedback toman la decisión del valor de la variable manipulable, es mediante el cálculo de la diferencia entre el valor de la variable de salida y el valor deseado (setpoint). El controlador más simple es aquel actúa como switch, tal que:

$$si \begin{cases} \varepsilon > 0 & c = ON \\ \varepsilon < 0 & c = OFF \end{cases}$$

Sin embargo, para disminuir el desgaste de los componentes mecánicos del relé y la presencia de arcos eléctricos, se utiliza una pequeña brecha alrededor del setpoint. Si el error es mayor a h (denominado histeresis) el actuador se apagará, en cambio si el error está bajo -h se encenderá. El tipo de respuesta que se obtiene de este controlador se observa en la Figura B.6.

El controlador proporcional es el más simple, con excepción del controlador de 2-puntos, donde la salida del controlador será proporcional a esta diferencia $\mathrm{e}(t) = y_{sp} - y_m$.

$$c(t) = K_c \cdot \mathrm{e}(t) + C_s$$

$$G_c(s) = K_c$$

Si bien, este controlador es muy simple, el valor del estado estacionario luego de una perturbación no es igual al setpoint, y a esta diferencia se denomina offset. Para solucionar esto se le añade al controlador una acción integral o de reajuste, lo que es denominado controlador PI, el cual tiene por salida:

$$c(t) = K_c \cdot \mathrm{e}(t) + \frac{K_c}{\tau_I} \int_0^t \mathrm{e}(t)\mathrm{d}t + C_s$$

$$G_c(s) = K_c \left(1 + \frac{1}{\tau_I s}\right)$$

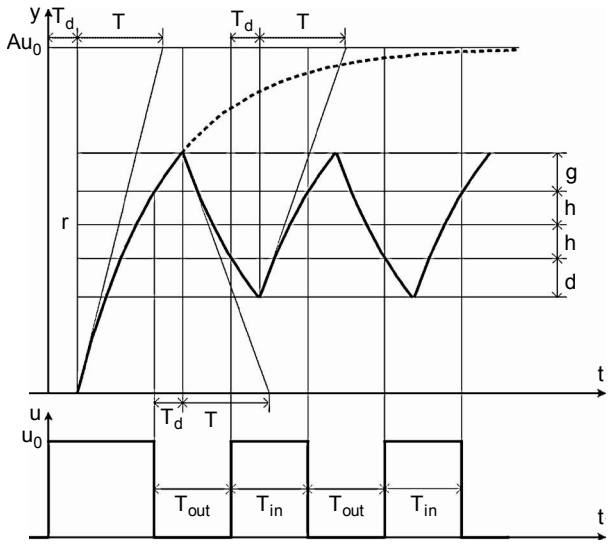


Figura B.6: Señal de salida de un controlador de tipo on/off (u) que produce una respuesta oscilatoria (y) [13].

El controlador PID añade otro término, que tiene como propósito anticipar el proceso mediante la observación del cambio del error a través de su derivada. Este controlador se describe como:

$$c(t) = K_c \cdot \mathrm{e}(t) + \frac{K_c}{\tau_I} \int_0^t \mathrm{e}(t)\mathrm{d}t + K_c \tau_D \frac{\mathrm{d}}{\mathrm{d}t}\mathrm{e}(t) + C_s$$

$$G_c(s) = K_c \left(1 + \frac{1}{\tau_I s} + \tau_D s\right)$$

Para un proceso de tiempo característico corto, donde son rápidos y suceptibles al ruido, este controlador presenta algunos problemas debido a que el término derivativo amplifica el error. [20]

B.8. Elección del controlador y sintonización

Para mantener una salida del proceso en un determinado setpoint utilizando algún tipo de controlador, requiere como primer paso elegir el tipo de controlador según algún criterio, controlador que puede ser de tipo P, PI, PID, entre otros. Una vez seleccionado el controlador, es necesario darle un valor a los parámetros.

En general, un controlador proporcional (P) acelera la respuesta del sistema de control, pero produce que el sistema tenga offset. Al agregar el efecto integral para un controlador PI se elimina el offset, pero la respuesta se vuelve oscilatoria, así como también grandes valores de K_C producen inestabilidad. Por otro lado, un controlador del tipo PD acelera aún más la respuesta del sistema, pero aún sigue produciendo offset en la respuesta. Finalmente, un

controlador de tipo PID, elimina el offset y se generan oscilaciones debido a la aceleración producto del factor derivativo, y además es afecto al ruido.

Para considerar cual son los mejores parámetros, es necesario especificar cuál es el criterio con el que se juzga el desempeño del sistema de control. Estos criterios pueden ser:

- **Criterio de estabilidad**: Indica si la respuesta es estable o no.
- **Criterio de estado estado estacionario**: Indica la presencia de offset en la respuesta.
- **Criterio de la respuesta dinámica**: Indica cómo el sistema responde luego de una perturbación, como oscilaciones, velocidad a la que se acerca al estado estacionario, entre otros.

Otros criterios cuantitativos se pueden describir mediante el uso de las siguientes ecuaciones:

1. Integral Absolute Error (IAE):

$$IAE = \int_0^\infty \|\varepsilon(t)\| \, \mathrm{d}t$$

2. Integral Squared Error (ISE):

$$ISE = \int_0^\infty \varepsilon^2(t) \mathrm{d}t$$

3. Integral Time-weighted Absolute Error(ITAE):

$$ITAE = \int_0^\infty t \, \|\varepsilon(t)\| \, \mathrm{d}t$$

4. Integral Time-weighted Squared Error (ITSE):

$$ITSE = \int_0^\infty t\varepsilon^2(t) \mathrm{d}t$$

Donde, tanto el ITAE y el ITSE penalizan los errores cercanos al estado estacionario.

Para la elección de los valores de los parámetros del controlador, denominada sintonización del controlador, generalmente se requiere saber o tener algo de conocimiento del proceso, ya sea a través de la experiencia o la utilización de modelos de manera que la respuesta del sistema se asemeje a lo esperado según el criterio elegido. Las metodologías utilizadas para la sintonización pueden ser divididas en 2 grandes grupos: Los que requieren de un modelo y las que no.

Por el lado de aquellas que requieren del modelo, se pueden mencionar las que se basan directamente del modelo, o que utilizan márgenes de estabilidad.

El método de síntesis directa cae dentro de los mecanismos basados en modelos que buscan la sintonización de parámetros de un controlador. Este método busca que la respuesta del

sistema sea $y(s) = q(s)y_{\mathrm{d}}(s)$, donde el controlador será:

$$g_c(s) = \frac{1}{g}\left(\frac{q}{1-q}\right)$$

Dependiendo de la trayectoria buscada, el controlador puede ser o no ser del tipo PID. Otro tipo de controlador basado en el modelo del sistema, es el IMC (Internal Model Control), el que dependiendo de la función de transferencia y el orden del filtro puede producir un controlador del tipo PID.

Por otro lado, entre las metodologías que utilizan márgenes para asegurar la estabilidad se puede nombrar Ziegler-Nichols (Ver Tabla B.3), método que requiere conocer valores como K_u y P_u, los que se obtienen al obtener la frecuencia de crossover, que corresponde a una frecuencia tal que al ingresar una onda sinusoidal de frecuencia ω_{co} al sistema en lazo abierto, la respuesta será una onda periódica.

Tabla B.3: Sintonización de parámetros utilizando el margen de estabilidad de Ziegler-Nichols [15].

Tipo	K_C	τ_I	τ_D
P	$0.5K_u$	-	-
PI	$0.45K_u$	$P_u/1.2$	-
PID	$0.6K_u$	$P_u/2$	$P_u/8$

Por otro lado, hay métodos que no requieren un modelo del sistema, pero la sintonización la realizan a partir de una aproximación a éste. Por ejemplo, el método de la curva de reacción (Cohen y Coon) describe el proceso a partir de 3 parámetros K, τ y α, los que se obtienen experimentalmente luego de realizar un cambio escalón en la variable de entrada (Ver Figura B.7). Las reglas de sintonización fundamentadas en el criterio de error de integración mínimo, se basan en minimizar los valores de IAE, ISE, ITAE o ITSE.

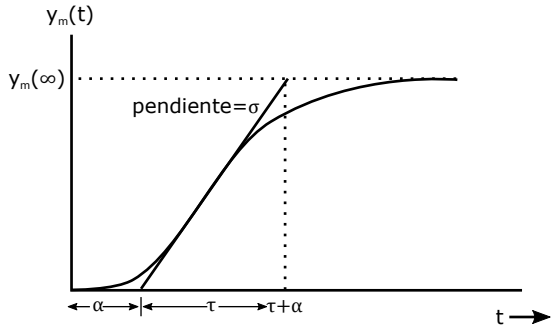


Figura B.7: Parámetros del método de la curva de reacción producto de la respuesta del sistema frente a un cambio escalón [15].

Anexo C

Filtro de Línea y Jaula de Fadaray

C.1. Filtro de Línea

Los filtros de línea protege contra ruidos eléctricos y contra sobretensiones transitorias. Este sistema utiliza dos mecanismos, un circuito RLC que actúa como filtro de alta frecuencia, y en caso que el circuito no actúe por completo actúa un varistor, que es un componente eléctrico, similar a los diodos, que suprime energías transitorias. El circuito de este filtro se observa en la Figura C.1.

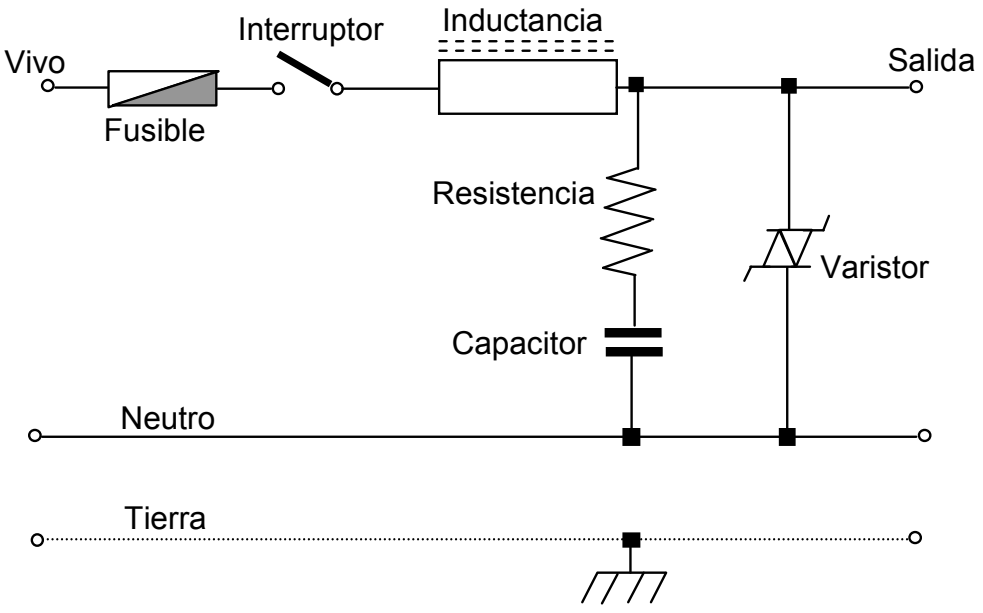


Figura C.1: Diagrama de un circuito de un filtro de Línea [8]

C.2. Jaula de Fadaray

La jaula de Fadaray corresponde al efecto que un objeto conductor cerrado en equilibrio (por ejemplo, conectado a tierra) produce un campo electromagnético nulo en su interior debido a la Ley de Gauss [24]. Esto produce que cualquier se genere un campo eléctrico en sentido contrario a un campo electromagnético en su interior. Debido a esto, la utilización de este efecto en componentes eléctrico y electrónicos produce que interferencia electromagnética (EMI) se elimine en el interior de este objeto.

Anexo D

Calibraciones

D.1. Temperatura

Para la temperatura, debido a que la salida del transmisor de temperatura es de 0 a 10V y la de entrada de elemento de control es hasta 5V, se utilizaron circuitos que transformen este voltaje. Se decidió probar el primer circuito hecho con Amplificadores Operacionales (ver Figura 4.9a). Para ello, se probó en primera instancia la relación entre voltaje de entrada vs voltaje de salida del circuito (ver Figura D.1), lo cual se hizo utilizando una fuente de poder.

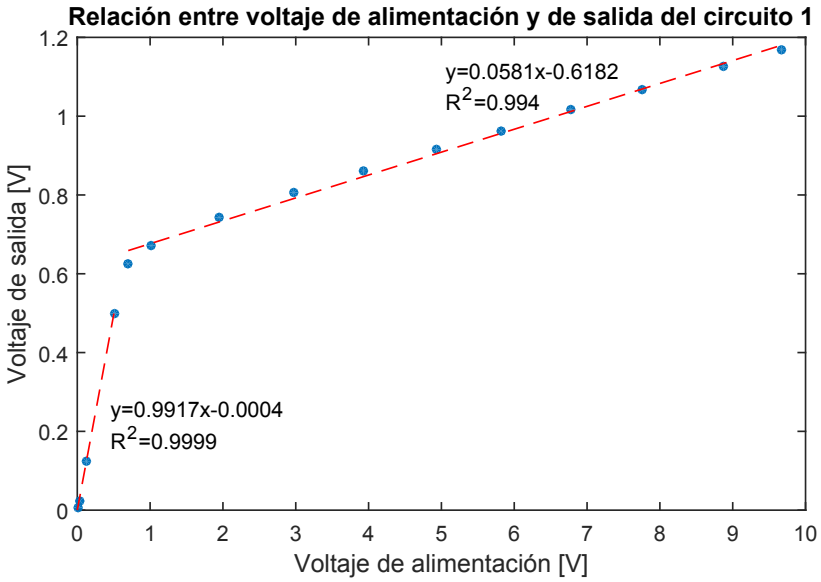


Figura D.1: Relación entre voltaje de alimentación y de salida del circuito.

Luego de eso, se hizo una relación entre la temperatura medida por el controlador con el voltaje medido en el transmisor(ver Figura D.2).

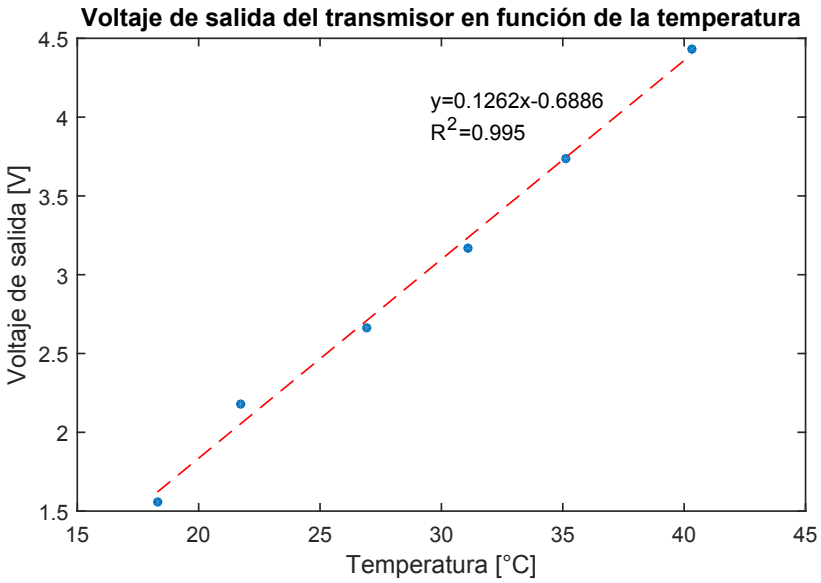


Figura D.2: Voltaje de salida del transmisor en función de la temperatura.

Dado que no se está ocupando la medición de temperatura en el mismo punto, la proximidad entre el sensor de controlador y el que se registra en el transmisor se asumen iguales. Con estas dos curvas se relacionó el voltaje de salida del circuito y la temperatura medida (Ver Figura D.3).

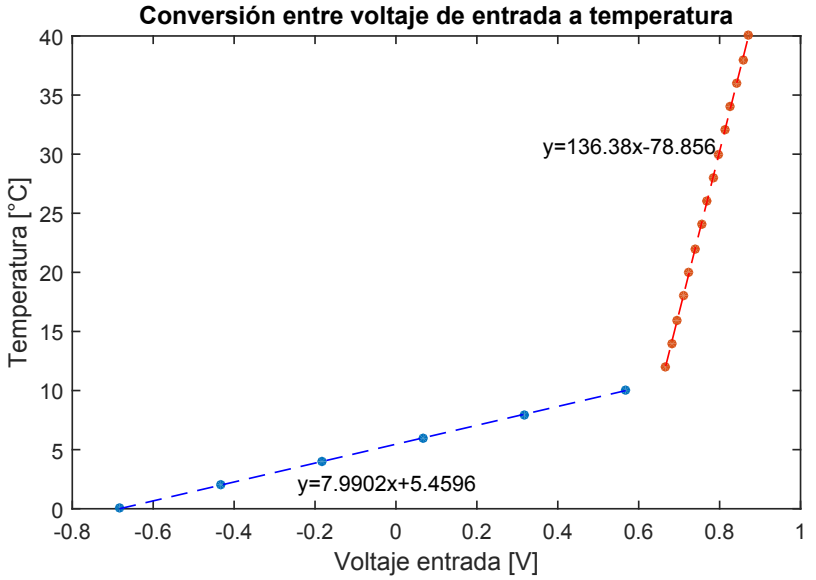


Figura D.3: Conversión de voltaje medido a temperatura usando curvas de Figura D.1 y D.2.

Se probó esta transformación y se observó que no existía un patrón regular en la medición (Ver Figura D.4), y midiendo la señal desde el sensor y hasta el elemento de control, se observó que el problema se encontraba entre la salida del transmisor y el circuito. Dado que al medir la salida del transmisor no se observó nada irregular, se decidió por cambiar el

circuito y reemplazarlo por una versión más fácil que permita disminuir el voltaje a un nivel deseado.

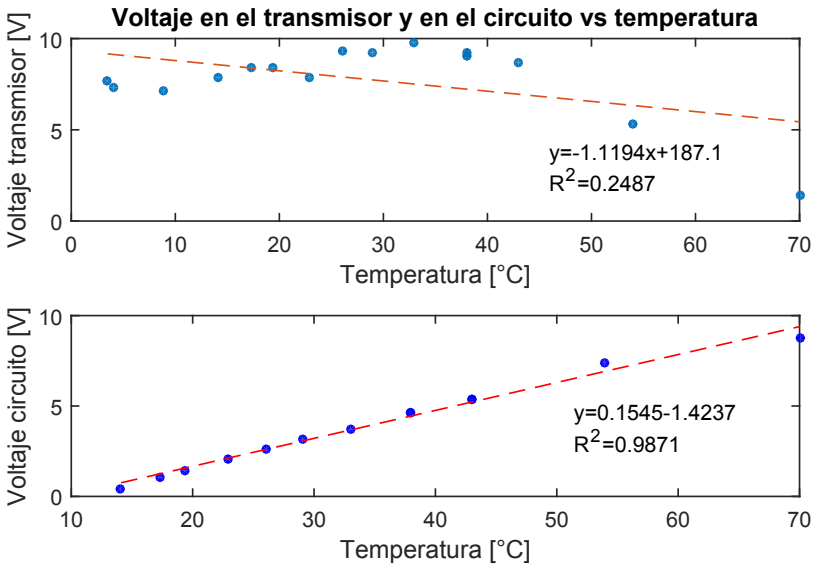


Figura D.4: Voltaje a la salida del transmisor y a la salida del circuito en función de la temperatura.

Una vez reemplazado con el circuito que posee resistencias (Ver Figura 4.9b), se observó nuevamente el transmisor de temperatura y se observó la presencia de ruido. Se midió nuevamente la relación voltaje de salida del transmisor vs temperatura, pero para ser más preciso se medió la resistencia del sensor y a partir de tablas este valor se transformó a temperatura, lo que se observa en la Figura D.5.

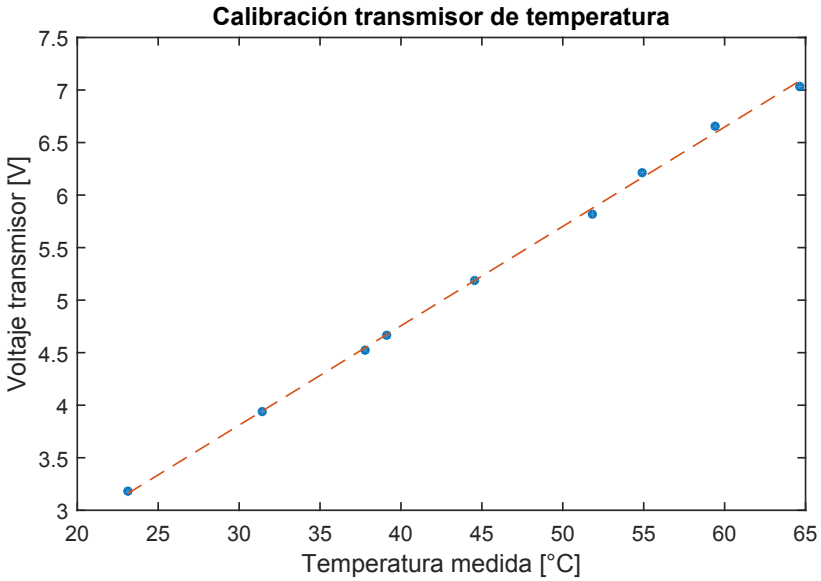


Figura D.5: Calibración de transmisor de temperatura.

Luego, se hizo la calibración entre temperatura medida a través de resistencia y el valor medido en el Arduino en la entrada análoga, lo que se observa en la Figura D.6.

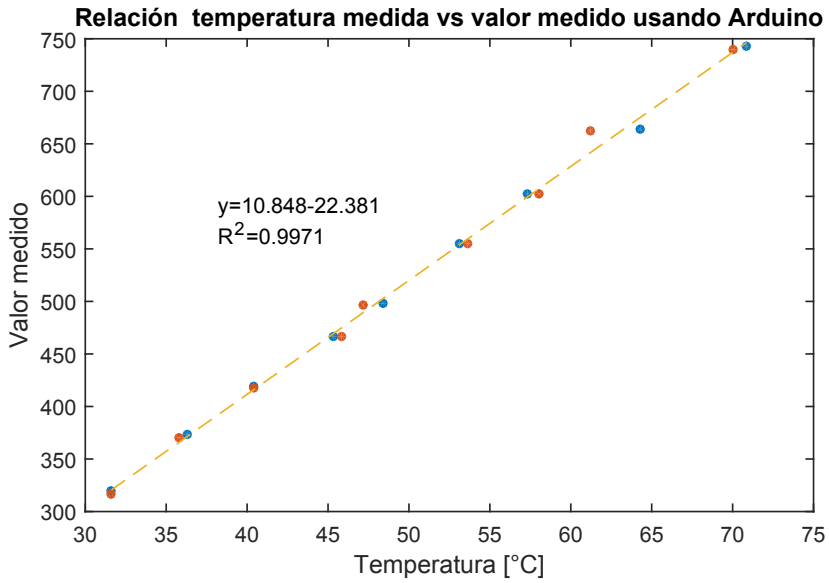


Figura D.6: Valor medido en Arduino en función de temperatura.

D.2. Flujo

En el caso de la temperatura, se utilizó un estanque graduado y el sensor se alimentó con agua proveniente de la red del laboratorio. Con ello, se registrando el volumen en el tiempo, para obtener el flujo. El sensor, se encontraba conectado al Arduino y se realizó un conteo de vueltas del álabe de la miniturbina del sensor (Ver Figura D.7).

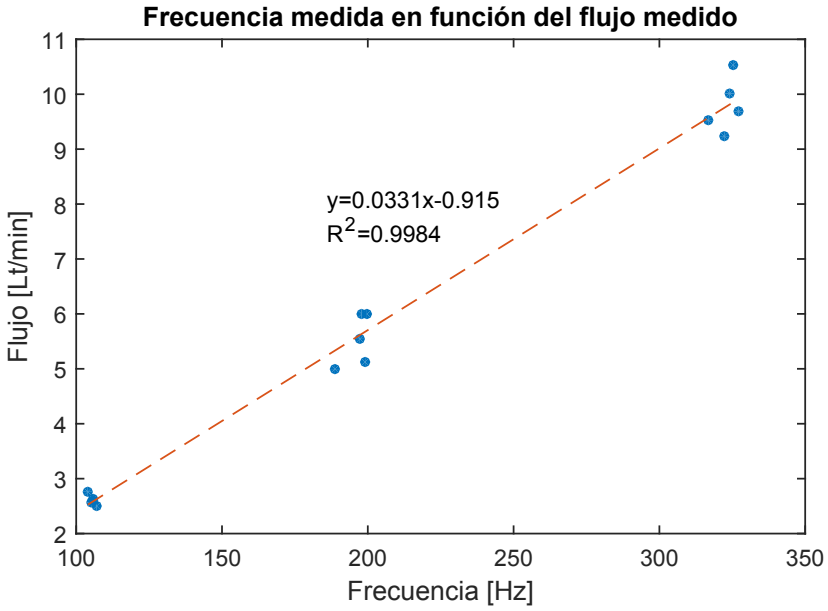


Figura D.7: Frecuencia en función del flujo medido.

Sin embargo, como se mencionó en el Capitulo 4 al realizar el cambio del elemento de control no fue posible realizar el contador de frecuencia y con ello se utilizó una función que mide periodos, la relación entre el periodo calculado y el flujo se observa en la Figura D.8.

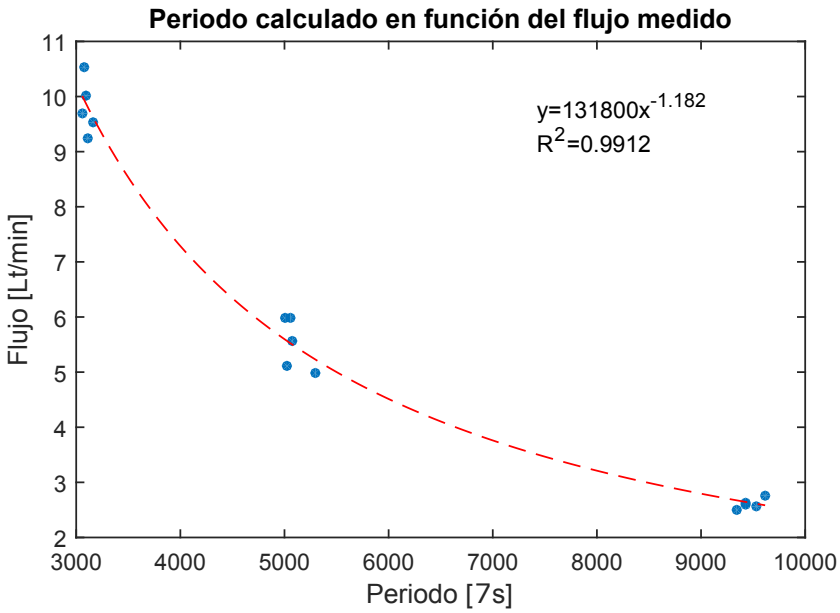


Figura D.8: Periodo calculado en función del flujo medido.

Anexo E

PCB

Para conectar los distintos componentes eléctrico y electrónicos para la adaptación de las señales de entrada y salida que van hacia el proceso se diseño y elaboró una placa de circuito impreso (PCB).

El proceso de elaboración de PCB, que es mostrado a continuación, corresponde a la eliminación de cobre de una lámina mediante la oxidación del cobre utilizando cloruro férrico (Ver Ecuación E.1). La idea de este método es que el cobre sea eliminado de la placa con la excepción de algunas zonas, las cuales son los lugares donde se sueldan los componentes o corresponden a pistas que actúan como conductores para comunicar distintas zonas.

$$2FeCl_3 + Cu^0 \leftrightharpoons 2FeCl_2 + CuCl_2 \tag{E.1}$$

Aquellas zonas donde no se desea remover el cobre se cubren con tóner, el cual actúa como una agente limitante de manera que no exista contacto entre la solución de $FeCl_3$ con el cobre.

Para la elaboración del circuito, se hizo utilizando el software *PCB Wizard*, el que permite dibujar la posición y grosor de agujero y pistas, así como también plantillas para distintos componentes electrónicos. Este software permite la creación del circuito e impresiones por capa, para la elaboración de circuitos impresos más complejos, pero en este caso sólo se utilizará la capa de cobre, la cual entrega el positivo fotográfico espejo para ser transferida a una placa de cobre.

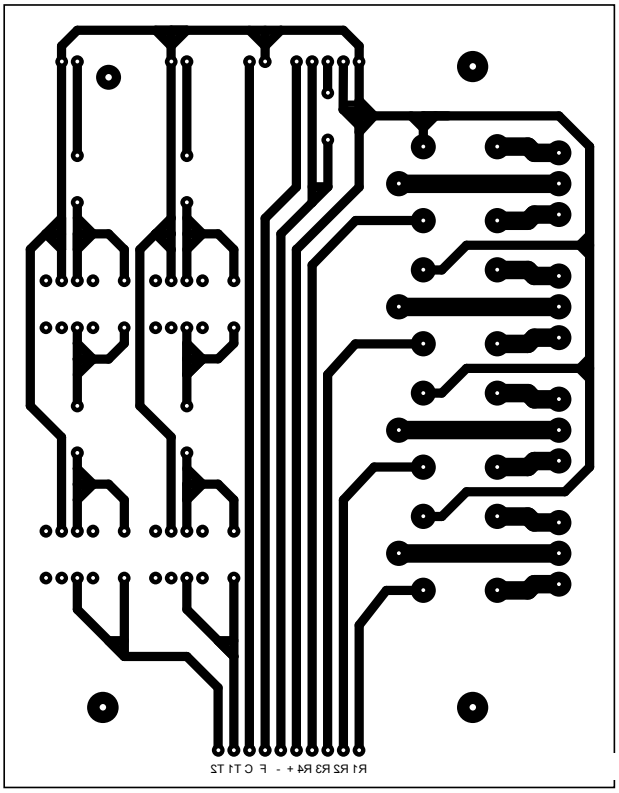


Figura E.1: Imagen de capa de cobre del PCB para la primera versión del circuito con PCB.

La siguiente lista contiene los distintos materiales utilizados para la elaboración:

- **Placa de cobre**: Corresponde a una placa de fibra de vidrio o de algún polímero, que posee una fina capa de cobre.
- **Cloruro férrico**: Solución ácida utilizada para eliminar el cobre de la a través de la oxidación de este elemento a través de una reacción redox.
- **Papel de transferencia**: Papel utilizado para transferir el circuito impreso hacia la placa de cobre, de preferencia debe ser papel fotográfico.
- **Impresora láser o fotocopiadora**.
- **Plancha**.
- **Lija fina**.
- **Lápiz permanente**.
- **Agente limpiador**: Se recomienda alcohol isopropílico o acetona.
- **Taladro pequeño**.

Los pasos para la elaboración del circuito son [6]:

1. Imprimir en el papel de transferencia la imagen espejo del circuito dibujado en el software.
2. Preparar la placa de cobre, lijando con un papel de lija fino y luego eliminar la suciedad con el agente limpiador.

3. Transferir el circuito a la placa de cobre: Colocar la imagen impresa sobre la placa de cobre de forma que el tóner esté sobre el cobre. Calentar el papel con la plancha a máxima temperatura utilizando otro papel de forma que la plancha no esté en contacto directo con la impresión.
4. Sacar el papel impreso de la placa de cobre, observando como el tóner que estaba en la hoja se traspasó a la placa de cobre. Para que la calidad de la transferencia sea buena, se recomienda sacar cuidadosamente el papel, incluso se puede utilizar agua para facilitar su remoción.
5. Para corregir detalles en el traspaso a la placa de cobre se puede utilizar el plumón permanente para rellenar los espacios no cubiertos por el tóner de la impresión.
6. Sumergir la placa de cobre en la solución de cloruro férrico utilizando elemento de protección personal para el trabajo con soluciones corrosivas. La velocidad del reacción dependerá de la pureza de la solución y temperatura. Se recomienda agitar la solución para aumentar la velocidad.
7. Limpiar con abundante agua la placa una vez que el cobre fue removido de la placa (con excepción de las partes cubiertas por el tóner) para eliminar los resto de la solución.
8. Eliminar el tóner que se encuentra en la placa utilizando una lija suave o el agente limpiador.
9. Taladrar los agujeros de la placa, se recomienda utilizar una broca de 0,8mm o 1mm.

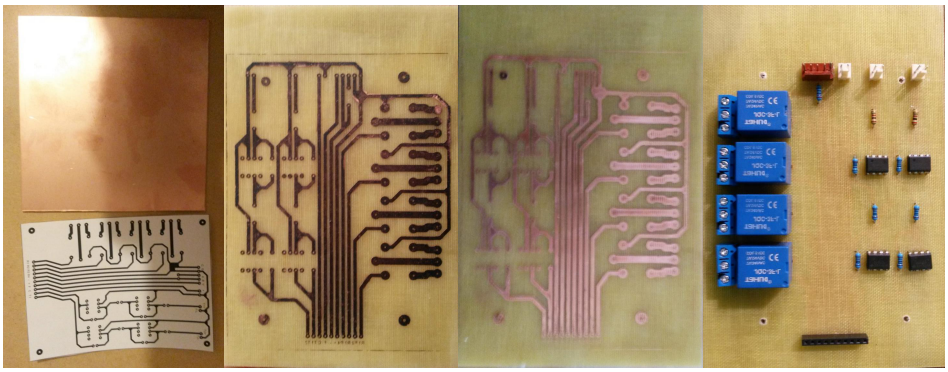


Figura E.2: Distintas etapas de la construcción del PCB para el circuito elaborado.

Anexo F

Arduino y Raspberry Pi

F.1. Arduino Mega 2560 + Arduino WiFi Shield

Las características del Arduino Mega 2560 se observan en la Tabla F.1.

Tabla F.1: Características de Arduino Mega 2560 [1].

	Descripción
Microcontrolador	ATmega2560
Voltaje Operativo	5 V
Voltaje de alimentación	5-12V
Voltaje de alimentación (límite)	6-20V
Puertos I/O digitales	54 (15 con salida PWM)
Entradas Análogas	16
Intensidad de corriente por I/O	40 mA
Intensidad de corriente 3.3V	50 mA
Memoria Flash	256 KB (8 KB reservadas para bootloader)
SRAM	8 KB
EEPROM	4 KB
Frecuencia de reloj	16 MHz
Serial conecction	USB Type B

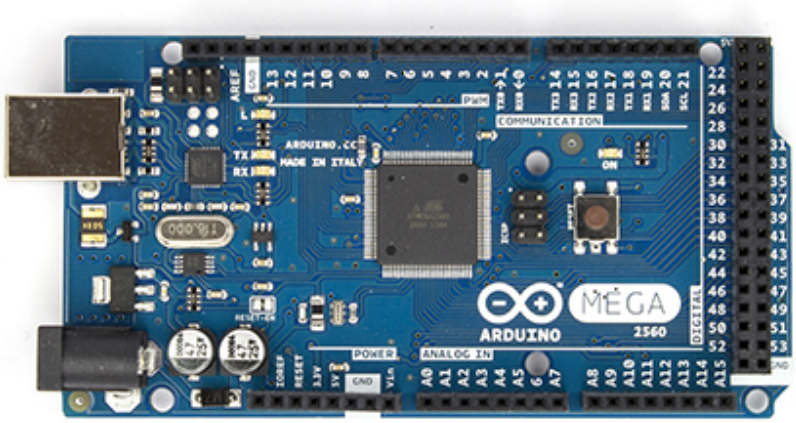


Figura F.1: Imagen Arduino Mega 2560 [1].

Para que el Arduino Mega tenga conectividad WiFi, se utiliza una tarjeta de expansión (Shield), cuyas características se observan en la Tabla F.2.

Tabla F.2: Características de Arduino WiFi Shield [3].

	Descripción
Voltaje operativo	5V (entregado por Arduino Board)
Conexión WiFi	802.11b/g
Encriptación	WEP/WPA2 Personal
Conexión a Arduino	Vía puerto SPI
Tarjeta de memoria	Micro-SD
Debugging	vía conexión serial FTDI
Updating	vía mini-USB

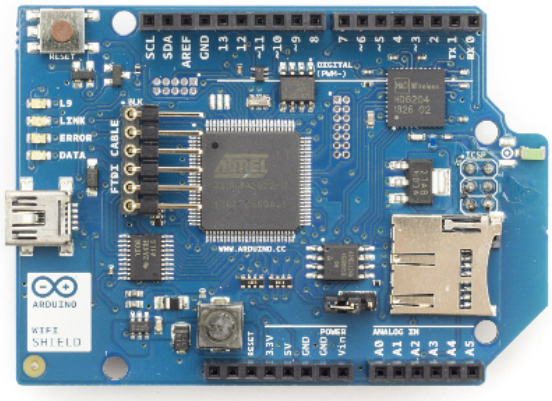


Figura F.2: Imagen Arduino WiFi Shield [3].

F.2. Arduino Yún

Las características del Arduino Yún se observan en la Tabla F.3.

Tabla F.3: Características de Arduino Yún [2].

AVR Arduino microcontroller	
Microcontrolador	ATmega32u4
Voltaje Operativo	5V
Voltaje de alimentación	5V
Puertos I/O digitales	20 (7 con salida PWM)
Entradas Análogas	12
Intensidad de corriente por I/O	40 mA
Intensidad de corriente 3.3V	50 mA
Memoria Flash	32 KB (4 KB reservadas para bootloader)
SRAM	2,5 KB
EEPROM	1 KB
Frecuencia de reloj	16 MHz
Linux processor	
Microcontrolador	Atheros AR9331
Arquitectura	MIPS@400MHz
Voltaje Operativo	3,3V
Ethernet	IEEE 802.3 10/100 Mbits/s
WiFi	IEEE 802.11b/g/n
USB Type-A	2.0
Lector de tarjetas	Micro-SD
RAM	64 MB DDR2
Memoria flash	16 MB

Figura F.3: Imagen Arduino Yún [2].

F.3. Raspberry Pi B+

Las características de Raspberry Pi B+ se muestran en la Tabla F.4.

Tabla F.4: Características de Raspberry Pi B+ [18].

	Descripción
Microprocesador	ARM 1176JZF-S a 700 MHz (ARM11)
GPU	Broadcom VideoCore IV
Memoria RAM	512 MB
Puertos USB 2.0	4
Salida vídeo	RCA, HDMI
Salida audio	3.5mm, HDMI
Tarjeta de memoria	Micro-SD
Conectividad de red	10/100 Ethernet (RJ-45)
Consumo energético	3 W
Voltaje de alimentación	5V vía USB
OS soportados	Raspbian (Debian), Pidora (Fedora), ArchLinux ARM (ArchLinux), RISC OS

Figura F.4: Imagen Raspberry Pi B+ [18].

Anexo G

Modulación por ancho de pulsos (PWM)

La modulación de ancho de pulsos (PWM) es la base para el control en sistemas de potencia [23]. Existen varias formas de realizar la modulación, por ejemplo mantener la frecuencia constante donde la señal PWM se obtiene al comparar una señal de referencia $r(t)$ con la señal portadora $c(t)$. La salida binaria producida por el PWM se obtiene matemáticamente como:

$$b_{PWM}(t) = \begin{cases} r(t) \geq c(t) & 0 \\ r(t) < c(t) & 1 \end{cases} \tag{G.1}$$

Los tres tipos de señales más comunes utilizadas como referencias son:

1. **Diente de sierra**: Debido a la forma de la señal, produce que el sistema generalmente inicie el periodo en ON. Este método se denomina constant-frequency trailing-edge modulation.
2. **Diente de sierra invertido**: A diferencia del caso anterior, la posición de encendido ocurre al final del periodo, y se denomina constant-frequency leading-edge modulation.
3. **Onda triangular**: Al utilizar este tipo de señal, la modulación combina el efecto de poseer una función que crece y decrece, lo que produce que el la posición ON se encuentre al centro del periodo. En este caso, a este método se denomina constant-frequency double-edge modulation.

Si bien los 2 primero métodos tienen un funcionamiento similar y el efecto de las posición ON en el periodo no afecta al sistema, el uso de una onda triangular es beneficioso debido a que elimina armónicos [23].

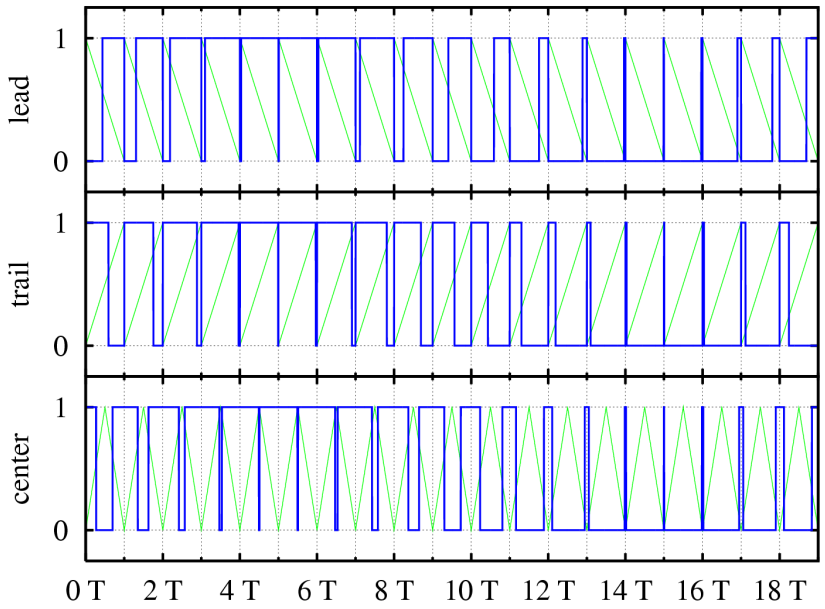


Figura G.1: Respuesta para modulación de ancho de pulso según señal portadora.

Otros métodos de modulación menos populares corresponden a sistemas con frecuencia variable, pero manteniendo el tiempo de apagado constante, el tiempo de encendido constante, o control con histéresis.

Anexo H

Tablas creadas para BD

La Figura H.1 muestra las relaciones entre las tablas de la base de datos implementada. El icono similar a una llave indica que es una clave primaria, el icono rojo que es una llave foránea y el icono azul es no nula.

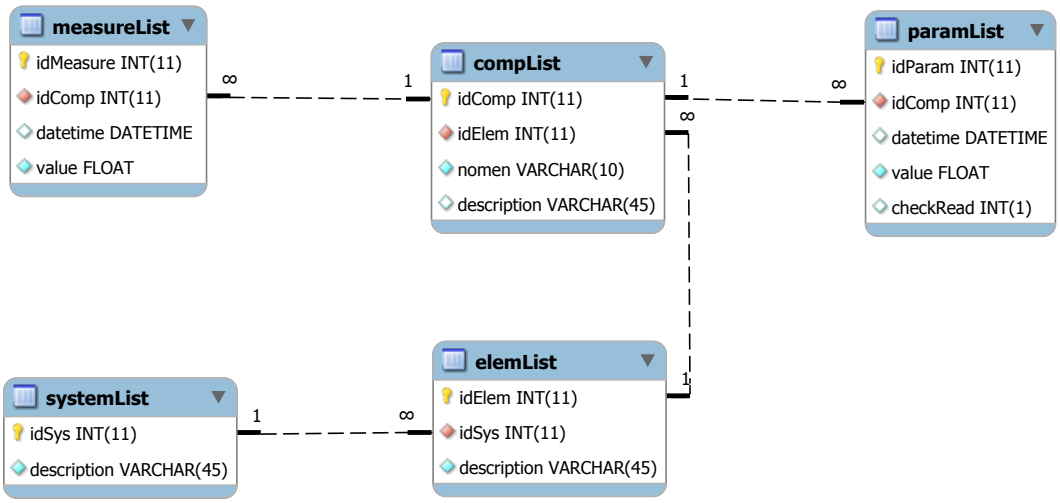


Figura H.1: Diagrama entidad-relación que representa la relación de datos entre las tablas.

Anexo I

Modelo fenomenológico: método explícito

Modelando el sistema como un tubo de diámetro constante y utilizando los supuestos mencionados en la Sección 4.3.3, como por ejemplo que la dinámica del calefactor es instantánea, se obtuvo la Ecuación 4.2. Para resolver esta ecuación diferencial parcial, se utilizó un método explicito que es mostrado a continuación. Para ello, se discretizó las derivadas parciales como diferencias centrales en torno a un punto (i, n) para las derivadas espaciales y como diferencias hacia adelante para las derivadas temporales, como se observa en las Ecuaciones I.1 y I.2.

$$\left.\frac{\partial T}{\partial t}\right|_{\mathrm{i},n} = \frac{1}{\triangle t}\left(T_{\mathrm{i},n+1} - T_{\mathrm{i},n}\right) + O(\triangle t) \tag{I.1}$$

$$\left.\frac{\partial T}{\partial z}\right|_{\mathrm{i},n} = \frac{1}{2\triangle z}\left(T_{\mathrm{i}+1,n} - T_{\mathrm{i}-1,n}\right) + O(\triangle z^2) \tag{I.2}$$

Reemplazando las Ecuaciones I.1 y I.2 en la Ecuación 4.2, donde la condición de borde corresponde a $T_{0,n} = T_{\mathrm{end},n}\ \forall n$ y la condición inicial es $T_{\mathrm{i},0} = T_{amb}$.

$$\frac{1}{\triangle t}\left(T_{\mathrm{i},n+1} - T_{\mathrm{i},n}\right) = \frac{-v}{2\triangle z}\left(T_{\mathrm{i}+1,n} - T_{\mathrm{i}-1,n}\right) + \begin{Bmatrix} \frac{q}{A_c \rho C_p} & z \leq L_0 \\ -\frac{4U}{\mathrm{d}\rho C_p}\left(T_{\mathrm{i},n} - T_{amb}\right) & z > L_0 \end{Bmatrix} \tag{I.3}$$

Despejando el término $T_{\mathrm{i},n+1}$ de la Ecuación I.3, donde la parte superior de los vectores corresponden a $z_\mathrm{i} \leq L_0$ y la inferior a $z_\mathrm{i} > L_0$.

$$T_{\mathrm{i},n+1} = \frac{-v\triangle t}{2\triangle z} T_{\mathrm{i}+1,n} + \begin{Bmatrix} 1 \\ 1 - \frac{4U}{\mathrm{d}\rho C_p} \end{Bmatrix} T_{\mathrm{i},n} + \frac{v\triangle t}{2\triangle z} T_{\mathrm{i}-1,n} + \begin{Bmatrix} \frac{q}{A_c \rho C_p} \\ \frac{4U}{\mathrm{d}\rho C_p} T_{amb} \end{Bmatrix} \tag{I.4}$$

Como se observa en la Ecuación I.4 no existe ningún término ni relación de tal manera que la ecuación cumpla la regla de positividad, ya que el término $\frac{-v\triangle t}{2\triangle z}$ siempre será menor que 0.

Anexo J

Imágenes HMI

En la siguiente sección se muestran imágenes del HMI implementado para el equipo, las que son descritas en la Tabla J.1.

Tabla J.1: Descripción imágenes de HMI

Figura	Descripción
J.1	Imagen del HMI utilizado en el equipo HL630 con la vista de los componentes del equipo.
J.2	Imagen del HMI utilizado en el equipo HL630 con la vista del diagrama de flujo del equipo.
J.3	Imagen de aplicación del sistema que permite observar el historial de una variable medida a través de un gráfico.
J.4	Imagen de aplicación para obtener datos registrados desde la base de datos.
J.5	Menú desplegable para el sensor de flujo, bomba y controlador.
J.6	Menú desplegable para el sensor de temperatura superior y ventilador.
J.7	Menú desplegable para el sensor de temperatura inferior y calefactor.

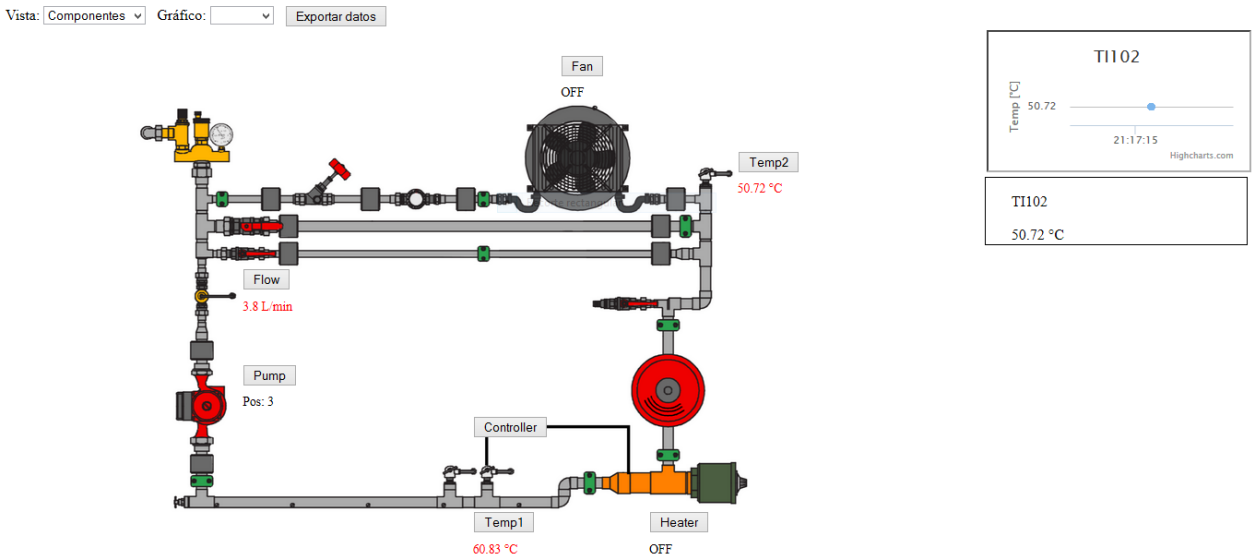


Figura J.1: Imagen del HMI utilizado en el equipo HL630 con la vista de los componentes del equipo.

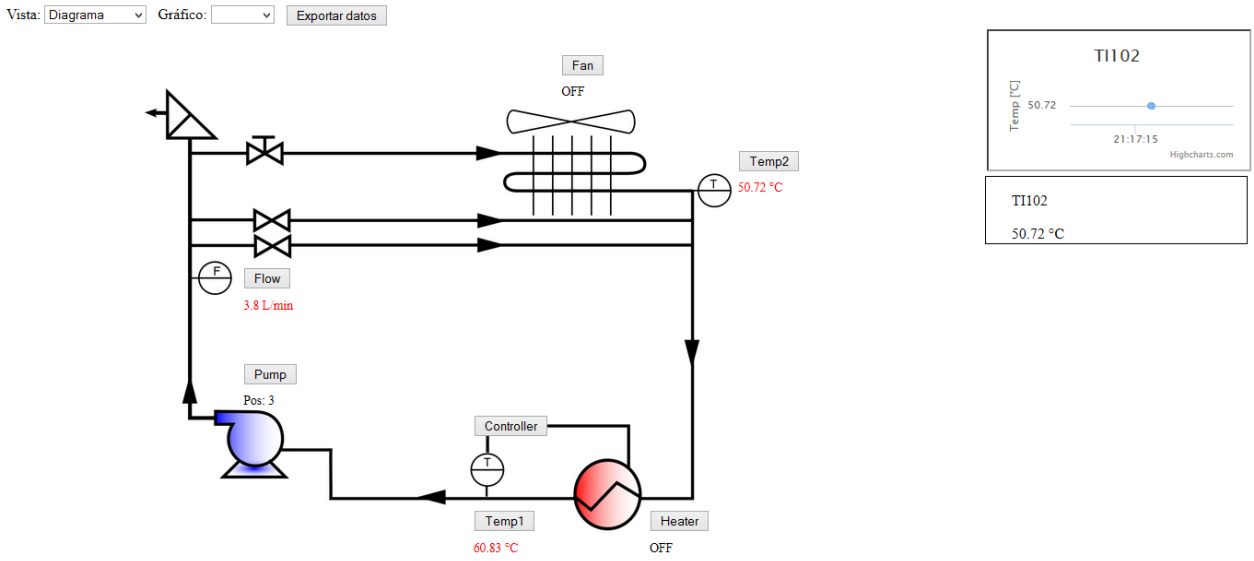


Figura J.2: Imagen del HMI utilizado en el equipo HL630 con la vista del diagrama de flujo del equipo.

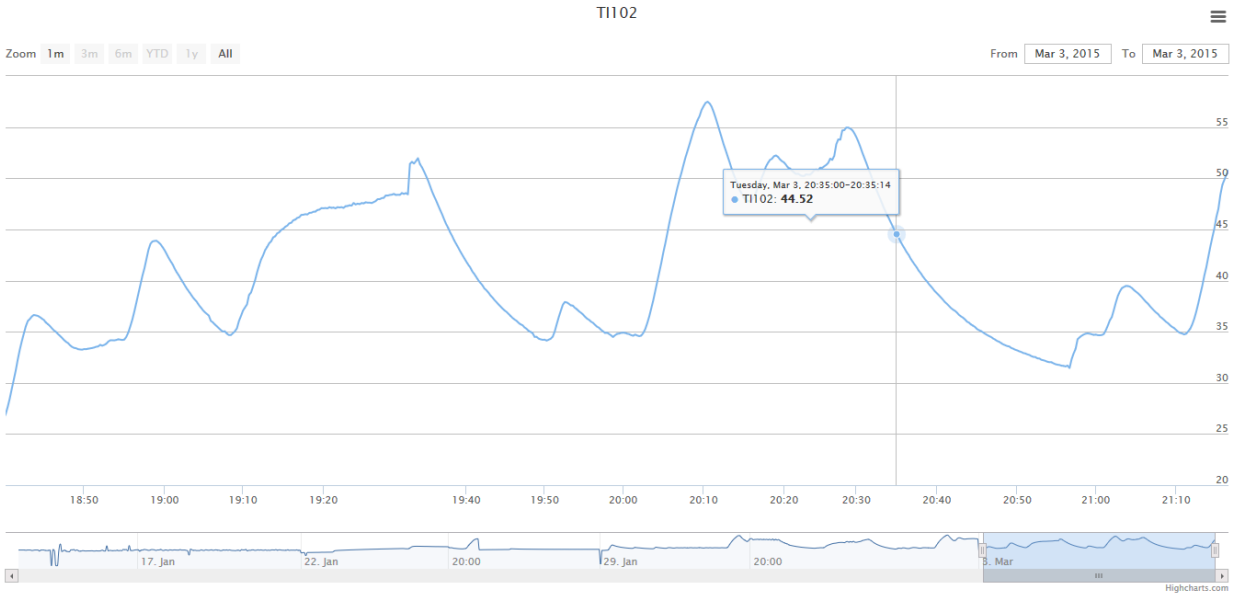
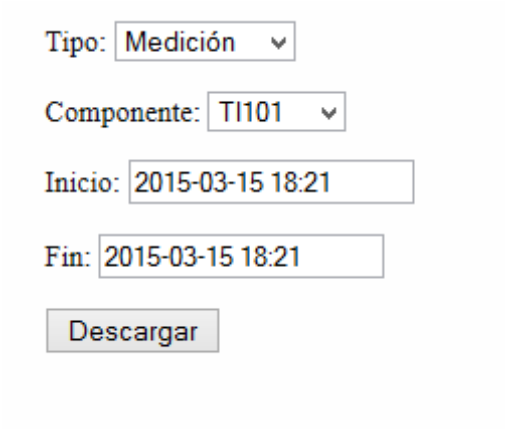
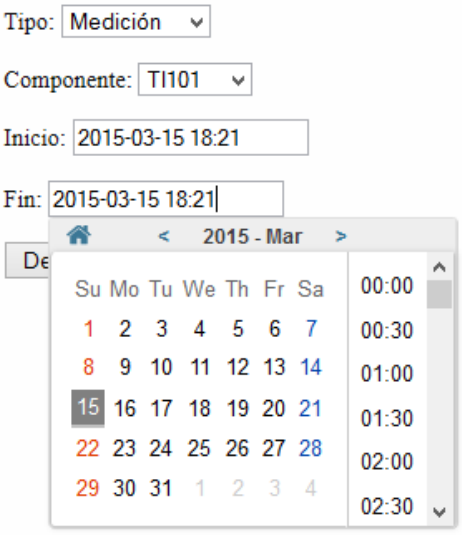


Figura J.3: Imagen de aplicación del sistema que permite observar el historial de una variable medida a través de un gráfico.

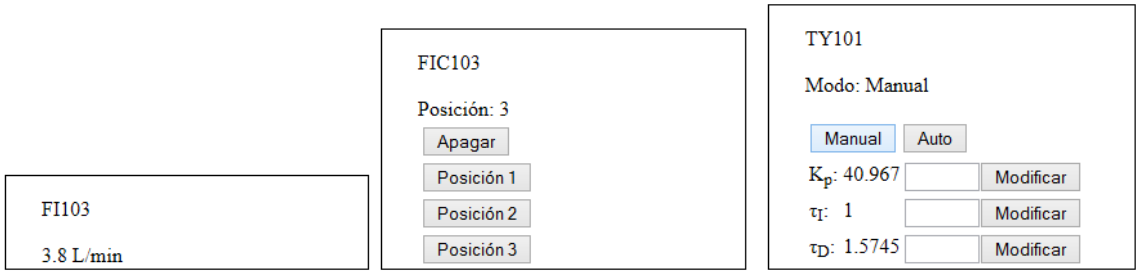


(a) Aplicación para obtener datos registrado en la BD.



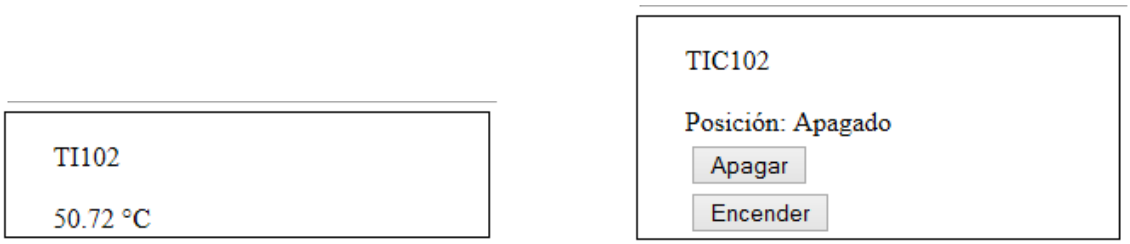
(b) Menú para facilitar el ingreso de fechas.

Figura J.4: Imagen de aplicación que genera una hoja de cálculo (Excel 2007) de una variable o parámetro entre 2 fechas. En la Figura J.4b se observa un selector de fechas utilizando DateTimePicker Control.



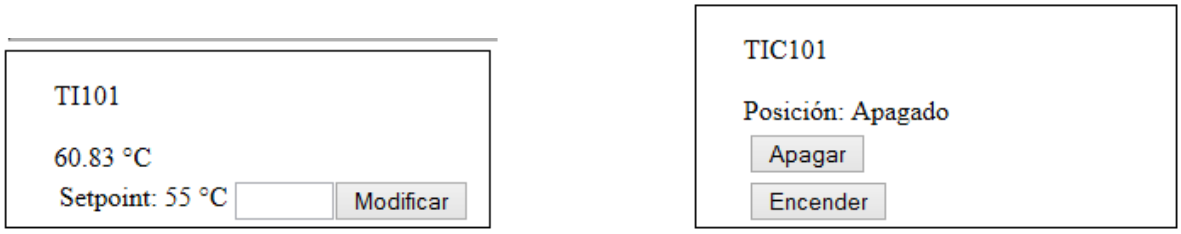
(a) Menú FI103. (b) Menú FIC103. (c) Menú Controlador.

Figura J.5: Opciones al desplegar el menú del sensor de flujo (Figura J.5a), bomba (Figura J.5b) y controlador (Figura J.5c).



(a) Menú TI102 (b) Menú TIC102.

Figura J.6: Opciones al desplegar el menú del sensor de temperatura superior (Figura J.6a) y el radiador con ventilador (Figura J.6b).



(a) Menú TI101. (b) Menú TIC101.

Figura J.7: Opciones al desplegar el menú del sensor de temperatura inferior (Figura J.7a) y calefactor (Figura J.7b).