

INSTITUTO TECNOLÓGICO SUPERIOR DEL SUR DE GUANAJUATO



“Análisis y desarrollo de un controlador PID para un horno académico”

Opción 2 Titulación Integral – Tesis profesional

Elaborada por:

FRANCISCO JAVIER MENDOZA DIAZ

Que presenta para obtener el título de:

INGENIERO EN SISTEMAS AUTOMOTRICES

Asesor:

M.T.W David Morales Orozco

“Análisis y desarrollo de un controlador PID para un horno académico”

Elaborada por:

FRANCISCO JAVIER MENDOZA DIAZ

Aprobado por..... MTW. David Morales Orozco
Docente de la carrera de Ingeniería en Sistemas Automotrices
Asesor de la Tesis Profesional

Revisado por..... Ing. Rogelio Orozco Martínez
Docente de la carrera de Ingeniería en Sistemas Automotrices
Revisor de la Tesis Profesional

Revisado por..... M.C Mariano Braulio Sánchez
Docente de la carrera de Ingeniería en Sistemas Automotrices
Revisor de la Tesis Profesional

DEPARTAMENTO ACADEMICO	CLAVE:11EIT0002E ISA-EGR-AD2023/15
---------------------------	---------------------------------------

Uriangato, Guanajuato, **20/octubre/2023**

Asunto: Aprobación de impresión de trabajo profesional

C. Francisco Javier Mendoza Diaz

PRESENTE:

Por medio de este conducto, le comunico a usted que después de haber sido revisado su trabajo de titulación bajo la del cual se derivó la Monografía Titulada:

“Análisis y desarrollo de un controlador PID para un horno académico”

La comisión revisora, ha tenido a bien aprobar la impresión de este trabajo.

ATENTAMENTE

*“Excelencia en Educación Tecnológica”
“Tecnología y Calidad para la Vida”*



M.C. Mariano Braulio Sánchez

Jefe de División de Ingeniería en Sistemas Automotrices



Instituto Tecnológico Superior
del Sur de Guanajuato
COORDINACIÓN INGENIERÍA
EN SISTEMAS AUTOMOTRICES

c.p Unidad de Servicios Escolares
C.c.p División de Ingeniería en Sistemas Automotrices
C.c.p Archivo Consecutivo



Ave. Educación Superior No. 2000, Col. Juárez, Uriangato, Guanajuato, C.P. 38982
Tels. (445) 45 7 74 68 al 71 Ext. *144, e-mail: automotrices@itsur.edu.mx
tecnm.mx | surguanajuato.tecnm.mx





LIBERACIÓN DE PROYECTO PARA LA TITULACIÓN INTEGRAL

Uriangato, Gto. 19/ octubre 2017

Asunto: Liberación de proyecto para la titulación integral

M.I José Gabriel Agullera González
Director Académico
ITSUR
PRESENTE

Por este medio informo que ha sido liberado el siguiente proyecto para la titulación integral:

Nombre de estudiante y/o egresado(a): Francisco Javier Mendoza Díaz	
Carrera: Ingeniería en Sistemas Automotrices	Núm. de control: T18120099
Nombre del proyecto: Análisis y desarrollo de un controlador PID para un horno académico	
Producto: Tesis Profesional	

Agradezco de antemano su valioso apoyo en esta importante actividad para la formación profesional de nuestras y nuestros egresados.

ATENTAMENTE

M.C. Mariano Braulio Sánchez
Jefe de División de Ingeniería en Sistemas Automotrices
ITSUR



La comisión revisora ha tenido a bien aprobar la reproducción de este trabajo.

Instituto Tecnológico Superior
del Sur de Guanajuato
COORDINACIÓN INGENIERÍA
EN SISTEMAS AUTOMOTRICES

M.T.W David Morales Orozco Asesor de la Tesis Profesional	Ing. Rogelio Orozco Martínez Revisor de la Tesis Profesional	M.C. Mariano Braulio Sánchez Revisor de la Tesis Profesional

c.c.p. - Expediente

Julio 2017

“Análisis y desarrollo de un controlador PID para un horno académico”

Resumen:

El controlador PID creado se basa en la introducción de sus tres valores (constantes de acciones de control) para implementarse en un horno académico creado por dos estudiantes de la carrera de sistemas automotrices para poder regularle la temperatura y sea capaz de elevar el horno a temperaturas considerablemente elevadas de más de 800°C para poder fundir aluminio de la serie 3xxx (latas de aluminio de jugos, refresco, cerveza, etc.), el cual por medio de una campaña de recolección en el ITSUR, será reciclado para su posterior fundición.

El controlador PID cuenta con un circuito eléctrico muy específico debido a que, se pensó para trabajar con una corriente alterna de 127V, pero por la deficiente instalación del ITSUR en cuanto a falta de corriente bifásica o trifásica, se tuvo que trabajar a 220VAC en una fuente externa capaz de generar ese voltaje, ya que, las pastillas termomagnéticas de la instalación eléctrica que se tienen en la escuela son de muy bajo amperaje, es decir, solamente tenían pastillas de 20 amperes que complicaban el óptimo funcionamiento del encendido de las resistencias eléctricas que contenía el horno, por la sencilla razón de que se votaba la pastilla cada que se conectaba más de una resistencia, por lo que se tuvo que conectar a la corriente de 220VAC para poder realizar pruebas del controlador PID final y con ello, poder ver si lograba estabilizar una temperatura no tan elevada, la cual fue de 190°C para ver si respondía de excelente manera regulando la temperatura sin oscilaciones tan bruscas, sino que, presentara oscilaciones pequeñas que se fueran desapareciendo en muy poco tiempo.

Palabras claves.

Controlador PID, sistemas en retroalimentación, control automático.

Abstract.

The PID controller created is based on the introduction of its three values (constants of control actions) to implement it in an academic oven created by two students of the automotive systems career to be able to regulate the temperature and be able to raise the oven to temperatures considerably elevated temperatures of more than 800°C to be able to melt aluminum of the 3xxx series (aluminum cans for juices, soft drinks, beer, etc.), which, through a collection campaign at the ITSUR, will be recycled for subsequent melting .

The PID controller has a very specific electrical circuit because it was thought to work with an alternating current of 127V, but due to the deficient installation of the ITSUR, it had to work at 220V since the pills that are in the school They are of very low amperage, that is to say, they only had 20 amp tablets, therefore, the optimal functioning of the ignition of the electrical resistances that the oven contained was complicated for the simple reason that the tablet was voted every time more than a resistance, so it had to be connected to the 220VAC current to be able to carry out tests of the final PID controller and with this to be able to see if it managed to stabilize a not so high temperature which was 190°C, to see if it responded excellently way regulating the temperature without such abrupt oscillations, but rather presenting small oscillations that would disappear in a very short time.

Keywords.

PID controller, feedback systems, automatic control.

Agradecimientos y Dedicatoria.

Primero que nada, quisiera agradecer a mis papás, mi hermana, mi novia, familiares en general, amigos, compañeros, y hasta algunos desconocidos por confiar en mí, por no dejar que me rindiera y por apoyarme a cumplir mi sueño ya que, siempre me estuvieron apoyando durante la carrera y a su vez, en este trabajo no fue la excepción ya que siempre confiaron en mí y es por ello que me siento afortunado de haber logrado este paso tan importante en mi vida gracias al apoyo de tan maravillosas personas. Muchos somos fruto de una familia de profesionistas y muchos otros, el día de hoy llenamos de orgullo a los nuestros por ser los primeros en lograrlo.

No podía faltar el agradecer a nuestros docentes, creo que cada uno de ellos, tiene algo más que ofrecernos que sólo conocimiento, pues de ellos también nos llevamos un poco de sus valores, de su coraje, de su dedicación para enseñar, de su valentía, su personalidad y de su carisma, aunque más allá de empaparnos con su conocimiento, nos impulsaron a ser personas íntegras, cultas y de respeto.

Quiero agradecer en especial al maestro Rogelio Orozco Martínez, el coordinador de la carrera Mariano Braulio Sánchez y al maestro David Morales Orozco por la confianza que depositaron en uno, para poder llevar a cabo este proyecto y, asimismo, por estar ayudando en cada momento que se presentaba algún problema en el trabajo como también por trabajar a la par en el proyecto dando consejos, observaciones e incluso hasta involucrándose en el mismo.

Asimismo, agradecer de corazón a dos personas muy importantes en mi vida que no están hoy ya conmigo pero que donde quiera que estén yo se que se encuentran orgullosos de lo que hoy estoy logrando, por lo que les dedico de todo corazón este éxito en mi vida a mi abuelo materno y mi tía (hermana de mi abuelo paterno).

Francisco Javier Mendoza Díaz

Índice general

Capítulo 1	12
Introducción.	12
Capítulo 2	14
Marco teórico.	14
Capítulo 3	39
Planteamiento del problema	39
3.1 Identificación.	39
3.2 Justificación.	39
3.3 Alcance.	40
Capítulo 4	42
Objetivos	42
4.1 Objetivos generales.	42
4.2 Objetivos específicos.	42
Capítulo 5	43
Metodología	43
Capítulo 6	59
Resultados	59
Capítulo 7	72
Análisis de Resultados	72
Capítulo 8	77
Conclusiones y trabajo a futuro.....	77
Referencias bibliográficas	80
Anexos	81

Índice de figuras

Figura 1. Función de transferencia $G(s)$	15
Figura 2. Función de transferencia de cualquier sistema de control.....	16
Figura 3. Elementos de un diagrama de bloques.....	18
Figura 4. Diagrama de bloques de un sistema de control de lazo abierto.....	18
Figura 5. Diagrama de bloques de un sistema de control de lazo cerrado.....	18
Figura 6. Elementos que componen un sistema de control.....	19
Figura 7. Esquema de sistema de control de lazo abierto.....	21
Figura 8. Esquema de sistema de control de lazo cerrado.....	22
Figura 9. Algoritmo de un controlador PID.....	25
Figura 10. Esquema de un control PID de una planta.....	25
Figura 11. Diagrama de bloques de un sistema de lazo cerrado simple.....	26
Figura 12. Logotipo de Arduino.....	30
Figura 13. Representación de un Optoacoplador fototransistor EL817.....	31
Figura 14. Puente de diodos rectificador de onda completa KBPC1010.....	32
Figura 15. Representación de un diodo MOC3020.....	33
Figura 16. Representación de un transistor TRIAC BTA16-600B.....	34
Figura 17. Representación del alambre Kanthal A-1.....	34
Figura 18. Termopar tipo K.....	35
Figura 19. Módulo de interfaz MAX6675.....	36
Figura 20. Módulo con interfaz I2C.....	37
Figura 21. Pantalla LCD 16x2.....	38
Figura 22. Convocatoria del Tecnológico Nacional de México.....	46
Figura 23. Primera prueba sin lijar las latas y cortadas en pedazos pequeños.....	47
Figura 24. Segunda prueba lijando las latas y cortadas en pedazos pequeños.....	48
Figura 25. Tercera prueba lijando y utilizando solo el cuerpo de las latas sin la tapa.....	49
Figura 26. Video acerca de los fundamentos de un controlador PID.....	50
Figura 27. Video acerca de un controlador PID con una corriente DC.....	52
Figura 28. Prueba de monitoreo de temperatura en monitor serial con un termopar tipo K.....	55
Figura 29. Pantalla LCD 16x2 mal conectada con sus 14 pines.....	56
Figura 30. Pantalla LCD 16x2 bien conectada con sus 14 pines.....	56
Figura 31. Monitoreo de la temperatura en una pantalla LCD 16x2 con modulo I2C.....	57
Figura 32. Controlador PID conectado a 220V y al horno académico.....	58
Figura 33. Programa con termopar tipo K que muestra la temperatura en el monitor serial.....	59
Figura 34. Prototipo de horno eléctrico en funcionamiento.....	60
Figura 35. Conexión del Arduino con el horno eléctrico con el termopar tipo K.....	61
Figura 36. Temperatura máxima censada (538°C) dentro del prototipo del horno eléctrico.....	61
Figura 37. Mensaje de inicio al comenzar a censar la temperatura.....	62
Figura 38. Pantalla LCD mostrando el aumento de temperatura.....	63
Figura 39. Pantalla LCD evidenciando el correcto funcionamiento del termopar tipo K.....	63
Figura 40. Componentes del controlador PID.....	64
Figura 41. Funcionamiento de comunicación de Arduino con el controlador PID.....	65
Figura 42. Controlador PID en funcionamiento con la placa de Arduino Mega2560.....	66
Figura 43. Implementación del controlador PID con un prototipo de horno a 220VAC.....	67
Figura 44. Conexión del controlador PID con el horno académico.....	68
Figura 45. Ajuste del SetPoint en el programa de Arduino previo a su carga a la placa.....	68
Figura 46. SetPoint y temperatura real dentro del prototipo del horno académico.....	69
Figura 47. Controlador PID oscilando por debajo de la temperatura del SetPoint.....	70
Figura 48. Controlador PID oscilando con una falla de 1°C por arriba del SetPoint.....	70
Figura 49. Controlador PID estabilizando la temperatura real a temperatura del SetPoint.....	71
Figura 50. Impresión del monitoreo de temperatura con el monitor serial.....	72
Figura 51. Dirección del LiquidCrystal para que funcionara la pantalla LCD.....	73

Figura 52. Primer programa para medir temperatura y mostrarlo en el monitor serial.	81
Figura 53. Segundo programa para monitorear temperatura desde un LCD 16x2 (1/3).	82
Figura 54. Segundo programa para monitorear temperatura desde un LCD 16x2 (2/3).	82
Figura 55. Segundo programa para monitorear temperatura desde un LCD 16x2 (3/3).	83
Figura 56. Programa final del controlador PID (1/6).	84
Figura 57. Programa final del controlador PID (2/6).	84
Figura 58. Programa final del controlador PID (3/6).	85
Figura 59. Programa final del controlador PID (4/6).	85
Figura 60. Programa final del controlador PID (5/6).	86
Figura 61. Programa final del controlador PID (6/6).	86

Índice de tablas

Tabla 1. Ventajas de los sistemas de lazo abierto y lazo cerrado.	23
Tabla 2. Desventajas de los sistemas de lazo abierto y lazo cerrado.	24
Tabla 3. Efectos de aumentar la constante K_p en un sistema de control PID.	27
Tabla 4. Efectos de aumentar la constante K_i en un sistema de control PID.	28
Tabla 5. Efectos de aumentar la constante K_d en un sistema de control PID.	29

Capítulo 1

Introducción.

Un controlador PID es considerado como el sistema de control industrial más utilizado, esto debido a, su excelente desempeño y su manera tan sencilla de diseñar e implementar un controlador basado en un sistema de lazo cerrado. Las siglas de controlador PID se derivan de las tres constantes que forman parte de él; la acción proporcional K_p , una acción integral K_i y una acción derivativa K_d . La constante de acción proporcional, se encarga de ajustar la ganancia con relación a la magnitud que lleva el error de una forma directa. La constante de acción integral, tiene la capacidad de eliminar el error cuando se encuentra en un estado estacionario, y por último la constante de acción derivativa, es la encargada de mejorar la velocidad de respuesta en el sistema para anticipársele al error.

En este trabajo se hablará del análisis y diseño de un controlador PID para la implementación en un horno académico, el cual será aprovechado especialmente por los estudiantes de la carrera de Ingeniería en Sistemas Automotrices para poder realizar fundiciones y tratamientos térmicos, los cuales son temas muy importantes en el sector automotriz y que por cuestiones de que disponibilidad es que no se puede aprovechar al máximo esta herramienta, ya que, solo hay un horno tipo mufla en la carrera, por lo que no es posible que los alumnos puedan adquirir esos aprendizajes valiosísimos, ya que, cada que alguien va a fundir prácticamente se lleva un día por la sencilla razón que esos procesos se llevan a cabo a altas temperatura y para seguridad de la mufla y de los estudiantes, se debe dejar enfriar la mufla en su totalidad para volver hacer uso de ella una vez que ya se hayan expulsado los gases que se pudieron haber generado en la fundición previamente realizada.

Por lo tanto, en este trabajo se mostrará paso a paso desde el análisis del controlador PID hasta la construcción del mismo.

Capítulo 1. Introducción.

Comenzando desde el capítulo 2 donde se muestra una breve explicación de cada uno de los componentes a utilizar en el circuito eléctrico del controlador, seguido del planteamiento del problema ubicado en el capítulo 3, donde se explicará la identificación del problema como su justificación y el alcance que se esperaría tener. Después de esta sección nos encontramos con el capítulo 4, donde se describen los objetivos del proyecto tanto general como específicos. En la siguiente sección del trabajo una vez mostrados los objetivos, pasamos al capítulo 5, en el cual se muestra la metodología realizada para llegar al objetivo trazado desde el inicio del proyecto. Pasando la sección de metodología, nos encontraremos con el capítulo 6 donde se mostrarán los resultados obtenidos y siguiendo al siguiente capítulo, el cual es el 7 se mostrarán los análisis de los mismos resultados obtenidos. Finalmente, se encontrará el último capítulo, el cual es el 8 donde se muestran las conclusiones del trabajo seguido de sus propuestas de trabajo a futuro.

Capítulo 2

Marco teórico.

Sistemas de control.

Antes de hablar de un sistema de control se debe de hablar de ¿qué es una planta?, una planta puede ser desde una pieza de equipo o tal vez un conjunto de partes de una máquina funcionando de manera conjunta cuyo propósito es la de realizar una operación en particular y que, a su vez cabe recalcar que cualquier objeto físico que pueda ser controlado se le puede considerar como una planta. Una vez dicho esto, se puede hablar acerca de los sistemas de control, sin embargo, primero se debe definir lo que es por sí solo un sistema para poco a poco comprender de mejor manera a lo que se refiere con sistemas de control. Cuando se habla de un sistema se le puede definir como aquella combinación de componentes que actúan juntos para llegar a un cierto objetivo.

Finalmente, se puede definir que un sistema de control es un conjunto de elementos que permite regular las variables dentro de él para obtener el valor de respuesta esperado, de manera que se puedan cambiar las variables externas o también llamadas variables de entrada para que éstas a su vez influyan en el valor de las variables de salida del sistema.

Para poder analizar y diseñar un sistema a de control primero se debería tener en cuenta el modelado matemático de los procesos controlados. Básicamente, cuando se cuenta con un proceso que tiene un cierto control, primero se debe tomar en cuenta todas las variables que puedan estar involucradas dentro del sistema, es decir, aquellas variables que sirven para describir el funcionamiento dinámico del proceso. Sin embargo, para explicar de mejor manera a que se refiere el tomar en cuenta todas las variables del proceso, es necesario basarse en el ejemplo que se muestra en [1], en el cual dice que considerando un motor utilizado para fines de control, sus variables del sistema se pueden determinar como el voltaje aplicado al propio motor ,la corriente que se está circulando por el embobinado de mismo motor, el par que se le pueda estar desarrollando al propio eje del del rotor, así como

Capítulo 2. Marco teórico (Antecedentes).

también el desplazamiento angular que se pueda presentar y por último y no menos importante sería la velocidad del rotor. Estas variables no están nomas por que sí, ya que están sustentadas por leyes físicas que ya están consolidadas y a su vez éstas nos encaminan a las ecuaciones matemáticas que nos describen en este caso la dinámica con la que cuenta el motor. Sin embargo, las ecuaciones de un sistema de control no siempre serán las mismas, ya que pueden ser lineales o no lineales, variantes o invariantes con la variable del tiempo y todo esto depende de la condición de operación del sistema que se esté trabajando, así como del análisis del modelado matemático.

En este caso el trabajo se un enfocó en los sistemas lineales y a su vez invariantes en el tiempo, por lo que, la manera más utilizada para realizar el modelado matemático de estos sistemas es apoyándonos en funciones de transferencia para poder representar la relación que existe de entrada-salida en un sistema de control.

Función de transferencia

Para ello se debe definir a la función de transferencia tal como se muestra en [2], la cual nos dice que, “la función de transferencia de un sistema descrito mediante una ecuación diferencial, lineal e invariante en el tiempo, se define como el cociente entre la transformada de Laplace de salida (función de respuesta) y la transformada de Laplace de la entrada (función de excitación) bajo la suposición de que todas las condiciones iniciales son cero”.

Se sabe que $G(s)$ es quien denota la función de transferencia en un sistema de control con una entrada y una salida, en la cuales se tiene a la entrada como $u(t)$ y a la salida como $y(t)$ y a su vez la respuesta del impulso denotada como $g(t)$, por lo tanto, la función de transferencia se conoce como se puede ver la Figura 1:

$$G(s) = L[g(t)]$$

Figura 1. Función de transferencia $G(s)$.

Fuente: Elaboración propia.

Por lo tanto, una vez que se sabe que la función de transferencia $G(s)$ tiene una relación con la transformada de Laplace de la respectiva entrada y salida del sistema se puede llegar a la función de transferencia de cualquier tipo de sistema de control, como se puede ver en la Figura 2, donde tenemos todas las condiciones iniciales con un valor de cero y que a su vez, se puede observar que $Y(s)$ es la transformada de Laplace de $y(t)$ (salida del sistema) y $U(s)$ es la transformada de Laplace de $u(t)$ (entrada del sistema).

$$G(s) = \frac{Y(s)}{U(s)}$$

Figura 2. Función de transferencia de cualquier sistema de control.

Fuente: Elaboración propia.

Finalmente se pueden deducir 5 grandes puntos a destacar los cuales son los siguientes:

1. La función de transferencia que se tiene en un sistema de control, se le conoce como modelo matemático por la sencilla razón de que forma parte de un método operacional para representar la ecuación diferencial que nos relaciona las dos variables principales del mismo que son la de salida con la variable de entrada.
2. La función de transferencia también se le puede considerar como una característica de un sistema, sin tomar en cuenta la magnitud y la propia naturaleza de la variable de entrada o a la vez función de excitación.
3. La función de transferencia cuenta con las herramientas necesarias para poder relacionar la entrada con la salida del sistema, pero no nos provee la información a tomar en cuenta de la estructura física del sistema. Esto nos quiere decir que se pueden tener sistemas físicos totalmente diferentes y contar con la misma función de transferencia.

4. Para poder comprender la naturaleza de un sistema, es necesario conocer la función de transferencia para estudiar cada una de las salidas/respuestas de una cantidad considerable de entradas.
5. Si en dado caso no se conoce la función de transferencia de un sistema, se puede llegar a conocer de manera experimental, es decir, que se tienen que introducir entradas previamente conocidas y por consecuente estudiar la salida del sistema y una vez que se llega a la formulación de la función de transferencia, se debe proveer una descripción completa de las propiedades dinámicas del sistema como se describió anteriormente.

Diagrama de bloques

El diagrama de bloques de un sistema de control es una representación gráfica de las funciones que lleva a cabo cada componente y el flujo de señales y que a su vez gracias a su sencillez y lo fácil de diseñar, son empleados con una gran frecuencia por usuarios dedicados a modelar cualquier tipo de sistema de control que vayan a crear. Estos diagramas muestran las previas relaciones existentes ya mencionadas en los párrafos anteriores, entre los diversos elementos que lo componen. Este tipo de diagramas se diferencian de gran manera de las representaciones de cálculos matemáticos abstractos debido a que, un diagrama de bloques cuenta con la ventaja de visualización más realista en cuanto al flujo de las señales involucradas en el sistema como tal.

En un diagrama de bloques se enlazan todas las variables del sistema con ayuda de los llamados bloques funcionales, los cuales sirven como un símbolo o esquema que nos ayuda a representar una operación matemática que actúa sobre la señal de entrada para producir la salida.

La función de transferencia del sistema se puede encontrar dentro de los bloques funcionales, que como bien se sabe se conectan con ayuda de flechas que nos sirven para señalar la dirección del flujo de señales.

En pocas palabras, la señal de un sistema de control solamente tiene la capacidad de fluir en la dirección que indican las flechas, así como se muestra en la Figura 3.

Capítulo 2. Marco teórico (Antecedentes).



Figura 3. Elementos de un diagrama de bloques.

Fuente: [2].

El uso de los diagramas de bloques cuenta con unas grandes ventajas, como, por ejemplo, que es muy sencilla su creación y diseño, por lo que la operación de funcionamiento del sistema se puede observar con mayor facilidad al inspeccionar dicho diagrama que al tratar de inspeccionar el sistema físico como tal, ya sea de un sistema de control en lazo abierto como se ve en la Figura 4.

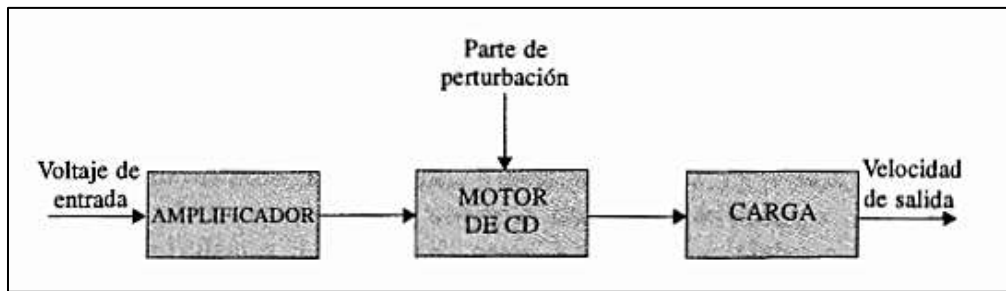


Figura 4. Diagrama de bloques de un sistema de control de lazo abierto.

Fuente: [1].

De la misma manera si es un sistema de control de lazo cerrado también se puede observar a simple vista, tal como se muestra en la Figura 5.

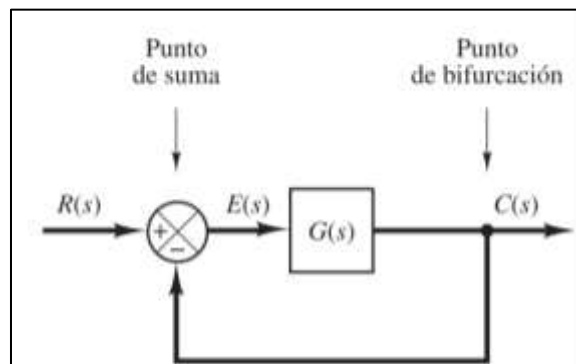


Figura 5. Diagrama de bloques de un sistema de control de lazo cerrado.

Fuente: [2].

Tipos de sistemas de control

Existen en grandes rasgos dos tipos de sistemas de control basados en bucles de control o también conocidos como lazos de control, el cual consta de todos aquellos componentes físicos del controlador y sus propias acciones de control necesarias para reestablecer o conservar el valor de alguna variable del proceso medida (PV) a su valor deseado, o punto de ajuste conocido como Set Point (SP).

Cabe mencionar que para lograr que este lazo de control funcione se necesita de tres partes primordiales las cuales son:

- Objetivos de control (entradas o acciones de control).
- Componentes del sistema de control (actuadores del controlador).
- Resultados o salidas (salidas o variables controladas).

Asimismo, se puede observar en la Figura 6, alguna referencia de los elementos que conforman un sistema de control de retroalimentación.

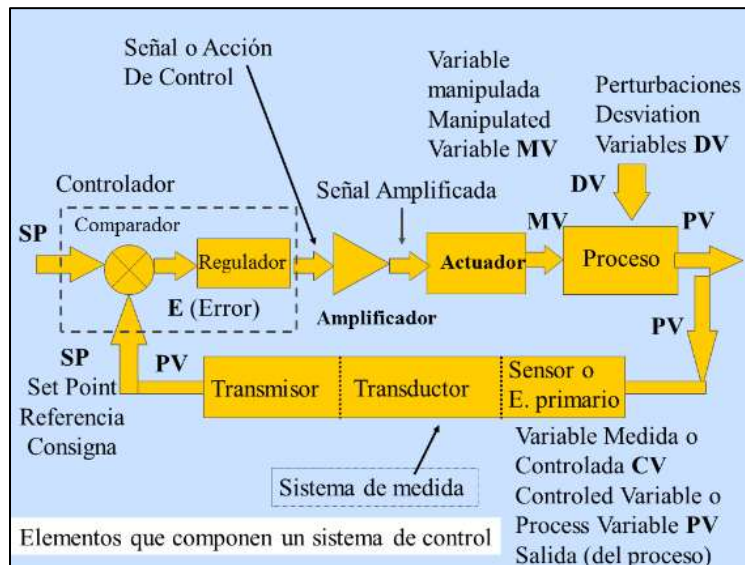


Figura 6. Elementos que componen un sistema de control.

Fuente: [3].

Sistemas de control en lazo abierto.

Este tipo de sistemas se puede resumir a aquellos en los cuales no existe una comparación del valor de salida con el de la entrada, es decir que la variable física de salida que tenga el sistema no va a llegar a tener efecto sobre el elemento que produzca la acción de control. Es por eso que este tipo de sistemas solamente es utilizado cuando se conoce la relación que pueda haber entre la salida y la entrada y también si existe algún tipo de perturbación ya sea interna o externa.

Como bien se menciona en [2], un ejemplo muy claro para describir mejor los sistemas de lazo abierto sería una lavadora, la cual tiene sus tres principales modos: lavado, enjuague y centrifugado, los cuales tienen algo en común y es el modo de operación basado en un intervalo de tiempo, es decir que, la lavadora no mide la salida del proceso que sería la del nivel de limpieza de la ropa, sino simplemente opera en una base de tiempo en cada uno de los modos sin importar esta variable de salida.

A parte de este ejemplo se tienen muchos más sobre sistemas de lazo abierto que se pueden entender con tan solo mencionarlos sin tener que explicar porque después de haber leído anteriormente en qué consisten este tipo de sistemas, algunos de los muchos que existen son:

- Tostadora de pan.
- Impresora de inyección de tinta.
- Un motor paso a paso.
- Secador de manos eléctrico el cual no mide la temperatura.

En pocas palabras en cualquier tipo de sistema de lazo abierto, no existe comparación entre la entrada y salida del sistema, por lo que a cada entrada de referencia le corresponde cierta condición de funcionamiento fija, por lo que el sistema tendrá una precisión muy buena o muy baja de acuerdo al nivel de calibración de dicho sistema, es decir, que si se cuenta con un sistema de lazo abierto que no está del todo calibrado y aun así existen perturbaciones ya sean

Capítulo 2. Marco teórico (Antecedentes).

internas o externas, no se podrá realizar la tarea deseada, como se puede en la Figura 7.

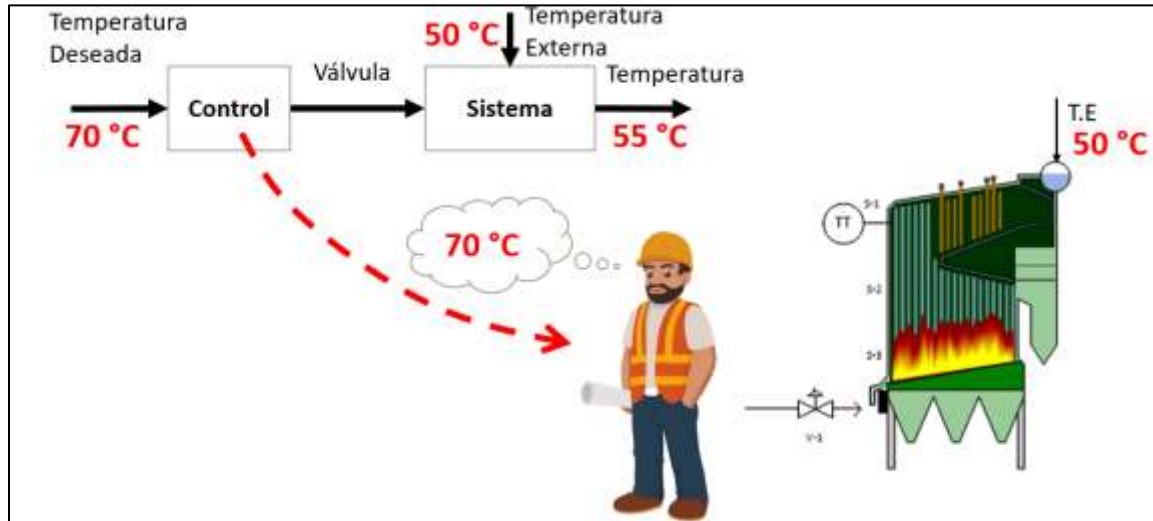


Figura 7. Esquema de sistema de control de lazo abierto.

Fuente: [4].

Sistemas de control en lazo cerrado

También conocidos como sistemas retroalimentados, es decir, que cuando se habla de este tipo de sistemas, se refiere a aquella relación estrecha entre una salida y una entrada deseada y estas a su vez son comparadas para su posterior uso en la resta de ambas variables y utilizar esta diferencia como el medio de control necesario para su posterior acción de control.

Como bien se menciona un ejemplo en [2], en el cual tenemos un sistema de control de temperatura en alguna habitación. Este sistema de control mide la temperatura real la cual sería la variable de salida del sistema y después la compara con la temperatura deseada que se conocería como la variable de entrada o variable de referencia, y en este caso el elemento de control es el encargado de tomar esa acción de control para poder regular la temperatura o tratar de llevar la temperatura real de la habitación a la temperatura deseada con ayuda de un termostato que pueda activar o desactivar el sistema de calefacción para elevar la temperatura si la

Capítulo 2. Marco teórico (Antecedentes).

temperatura real se encuentra por debajo de la temperatura deseada, o también puede activar o desactivar un sistema de enfriamiento si la temperatura real está por arriba de la temperatura deseada.

Un sistema de control de lazo cerrado no tiene un uso exclusivo en la ingeniería, es decir que no se limita a usarse sólo en el ámbito ingenieril, debido a que el propio cuerpo humano sería prácticamente un sistema de control retroalimentado muy avanzado por la sencilla razón de que al simple hecho de salir a entornos diferentes el cuerpo humano siga funcionando correctamente sin ningún problema debido a la retroalimentación fisiológica con la que contamos los humanos, y con ello se pueden controlar más sistemas que forman parte del cuerpo como la misma temperatura corporal al estar en un ambiente con la temperatura más baja o por encima de la corporal, siempre trata de mantenerla en un estado correcto para mantener esa retroalimentación constante, asimismo pasa con otros sistemas del cuerpo como lo sería la presión sanguínea.

En pocas palabras y como se menciona en [2], un sistema de control en lazo cerrado siempre implica el uso de una acción de control realimentado para reducir el error del sistema, véase en la Figura 8.

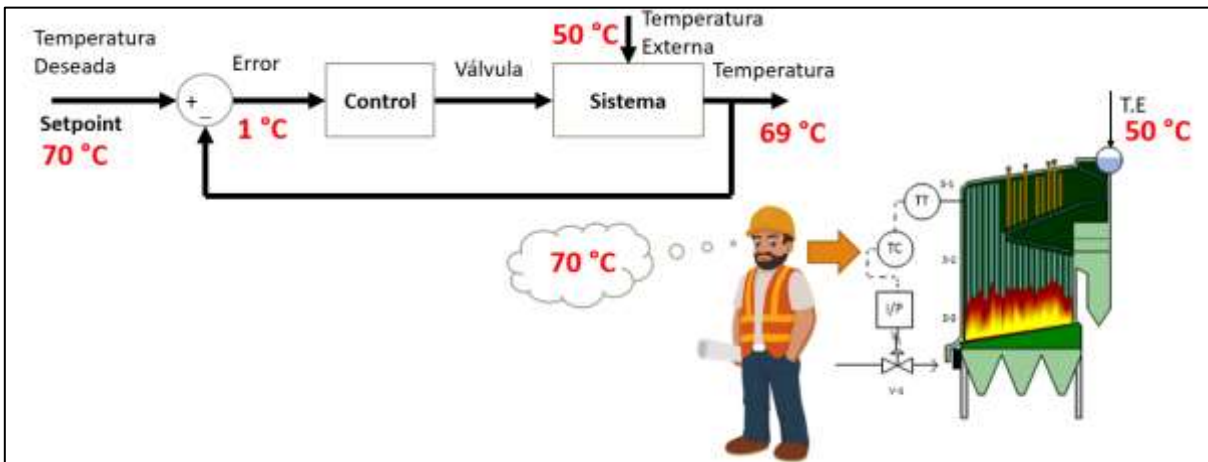


Figura 8. Esquema de sistema de control de lazo cerrado.

Fuente: [4].

Capítulo 2. Marco teórico (Antecedentes).

En pocas palabras se puede afirmar que existen unas grandes ventajas de cada tipo de sistema de control, ya sea de control en lazo abierto o de control en lazo cerrado como se puede ver en la Tabla 1.

Tabla 1. Ventajas de los sistemas de lazo abierto y lazo cerrado.

Fuente: Elaboración propia.

Sistemas de lazo abierto	Sistemas de lazo cerrado
Son muy sencillos de utilizar y poseen de un diseño simple en comparación a los de lazo cerrado.	Tienen una señal de retroalimentación para controlar la salida por lo tanto Son mucho más precisos y están menos involucrados a tener a errores debido a que, cuentan con una señal de retroalimentación con la cual se controlan de mejor manera la salida.
En comparación de otros sistemas de control son mucho más baratos.	Los errores del sistema son corregidos gracias a la señal de retroalimentación.
Son los mejores para cumplir actividades que se basan en tiempos o ciclos.	Son menos propensos a ser afectados por el ruido que llegue a presentar el proceso.
No requieren mucho mantenimiento.	Compatibles con la automatización del proceso.

De la misma manera se puede afirmar que también se cuenta con algunas desventajas al usar este tipo de sistemas de control, ya sea de control en lazo abierto o de control en lazo cerrado como se puede ver en la Tabla 2.

Capítulo 2. Marco teórico (Antecedentes).

Tabla 2. Desventajas de los sistemas de lazo abierto y lazo cerrado.

Fuente: Elaboración propia.

Sistemas de lazo abierto	Sistemas de lazo cerrado
Este tipo de sistema no cuenta con la capacidad de retroalimentación provocando la difícil o nula automatización del proceso.	Su diseño es mucho más complejo, así como el ser ajustado dentro del proceso.
Desgraciadamente son sistemas que son propensos a ser inexactos esto a su vez provoca que sean un poco precisos a comparación de los de lazo cerrado.	Debido a sus grandes ventajas y complejidades en cuanto al diseño del sistema suelen ser algo caros.
Este tipo de sistemas son muy propensos a ser afectados por las perturbaciones o ruidos que se lleguen a presentar de manera externa dentro del proceso.	Partiendo de que son sistemas más complejos, como consecuencia requieren a su vez de mayor mantenimiento.
Los sistemas de este tipo no tienen la capacidad de corregir las desviaciones de la salida de manera automática, debido a que no cuentan con la capacidad de la retroalimentación de un sistema de control.	Estos sistemas se les pueden desencadenar una serie de oscilaciones dentro del proceso si no se llegan a ajustar de manera correcta.

Controlador PID.

Este tipo de controlador se dice que es el más común en cuanto al algoritmo de control se trata, debido a que la gran mayoría de sistemas de control de lazo cerrado o de retroalimentación su control es basado en un controlador PID, aunque tal vez hay otras veces que se utilizan otros tipos de controladores, pero se siguen basando en el algoritmo del controlador PID. Sin embargo, para entender mejor en que se basa el algoritmo del controlador PID, véase en la Figura 9.

$$u(t) = K \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right)$$

Figura 9. Algoritmo de un controlador PID.

Fuente: [5].

Es importante mencionar que existe una gran ventaja en el uso del controlador PID, debido a que, si es posible obtener un modelado matemático de la planta o sistema, se puede hacer uso de algunas técnicas de diseño del mismo controlador, con el simple objetivo de conocer los parámetros del mismo y gracias a esto se pueda cumplir con las especificaciones que se requieran, ya sea en el estado tanto transitorio como en el estado estacionario del sistema en retroalimentación. Sin embargo, si el sistema o planta es algo compleja, es decir, que no es tan sencillo el poder obtener su modelado matemático, es muy probable que tampoco sea posible la existencia de un método analítico para el diseño de un controlador PID, por lo que en este tipo de casos se debería de apoyarse en métodos experimentales de la planta.

Para comprender de mejor manera el diseño de un controlador PID se puede ver en la Figura 10.

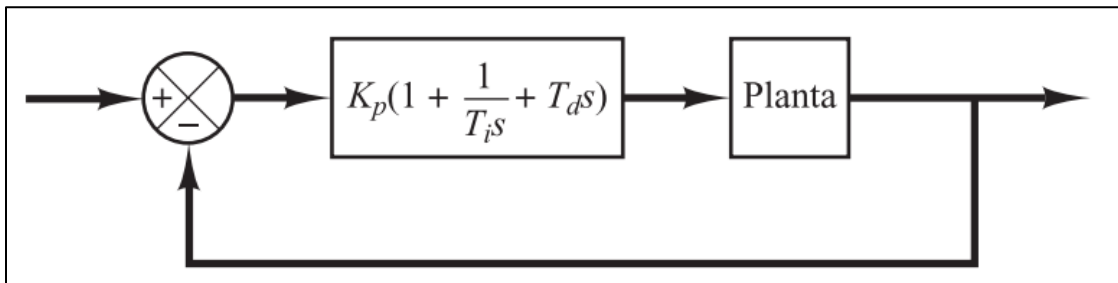


Figura 10. Esquema de un control PID de una planta.

Fuente: [2].

También se puede observar un sistema de retroalimentación simple con los elementos básicos que conlleva un diagrama de bloques en un controlador PID, y como se menciona en [5] para su mejor explicación, “u es la señal de control y e se

Capítulo 2. Marco teórico (Antecedentes).

entiende como el error de control ($e = y_{sp} - y$). La señal de control es así una suma de tres términos: el término P (que es proporcional al error), el término I (que es proporcional a la integral del error), y por último el término D (que es proporcional a la derivada del error). Los parámetros del controlador son la ganancia proporcional K , el tiempo integral T_i , y el tiempo derivativo T_d .”, todo esto se puede ver en la Figura 11.

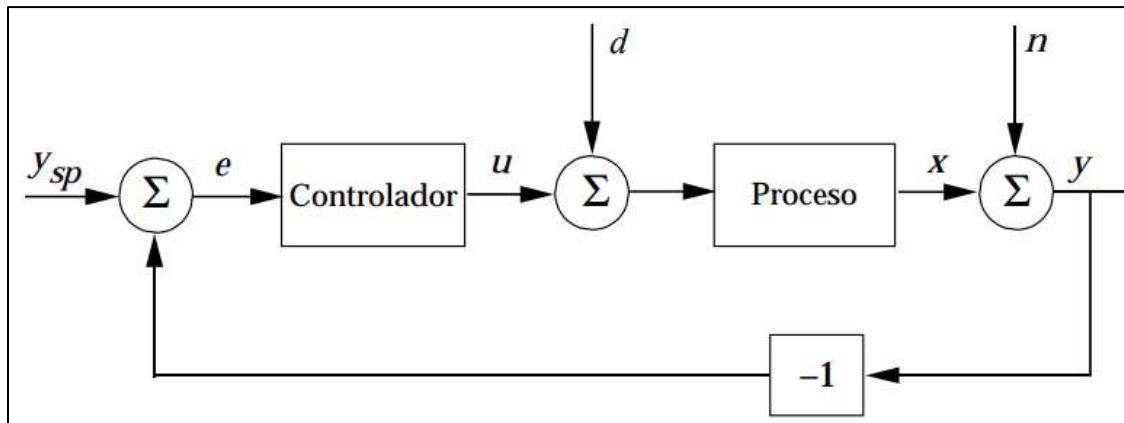


Figura 11. Diagrama de bloques de un sistema de lazo cerrado simple.

Fuente: [5].

Acción proporcional

Esta acción tal y como su nombre nos lo indica, es proporcional a la misma señal de error $e(t)$. Su funcionamiento se basa en que la acción proporcional multiplica la señal de error $e(t)$ por una constante conocida como K_p , la cual se encarga de determinar la cantidad en cuanto a magnitud de acción proporcional que necesitará el controlador.

En resumen, se puede afirmar que esta acción proporcional trata de reducir el error del sistema, es decir, cuando el error es pequeño, es más que claro que la acción de control es pequeña también. De igual manera si el error en el sistema es más grande, se entiende que la acción de control también será grande y con ello tratar de disminuir este error.

Al tratar de sintonizar el controlador PID y aumentamos la constante de acción

Capítulo 2. Marco teórico (Antecedentes).

proporcional K_p , vamos a obtener las siguientes consecuencias como se muestran en la Tabla 3.

Tabla 3. Efectos de aumentar la constante K_p en un sistema de control PID.

Fuente: Elaboración propia.

Acción (Aumentar K_p / Disminuir K_p)	Acción de control proporcional K_p
Aumentar K_p	Se disminuirá el error del sistema en un régimen permanente.
Aumentar K_p	Se aumentará la velocidad de respuesta del sistema.
Aumentar K_p	Es posible que se aumente la inestabilidad del sistema.

De manera objetiva se puede ver como a las primeras dos consecuencias como ventajas y la tercera sería una desventaja, ya que lo que se trata es que se estabilice el sistema por lo que debemos disminuir esa inestabilidad provocada. Sin embargo, si se aumenta la acción proporcional, va a llegar un punto de estabilidad, en donde se puede conseguir la rapidez de respuesta del sistema necesaria y a su vez, la reducción del mismo error, sin tener el problema de que el sistema sea inestable. Por lo que, no se debe aumentar la acción proporcional de más para no producir una inestabilidad no deseada en el sistema. Asimismo, si se mira del otro extremo, al reducir la acción proporcional nos aseguraremos de minimizar la velocidad de respuesta del sistema y también se aumentaría su error.

Acción Integral

Esta acción de control, es similar a la de acción proporcional en cuanto al nombre, ya que tal como su nombre lo menciona, es la encargada de calcular la integral de la señal de error $e(t)$ y a su vez la multiplica por la constante denominada por las letras K_i . Como bien se sabe una integral es conocida como el área bajo la curva y en este caso para mejor comprensión se puede entender como la acumulación de

Capítulo 2. Marco teórico (Antecedentes).

la señal de error o también se puede ver como una sumatoria de la misma señal de error, esto nos quiere decir que conforme avanza el tiempo aquellos errores se van a ir sumando para lograr que la acción integral cada vez sea mayor. Logrando así disminuir el error del sistema en un régimen permanente. La desventaja de utilizar la acción integral consiste en que esta añade una cierta inercia al sistema y, por lo tanto, le hace más inestable.

De la misma manera que la acción proporcional, al tratar de sintonizar el controlador PID y aumentamos la constante de acción proporcional K_i , vamos a obtener las siguientes consecuencias como se muestran en la Tabla 4.

Tabla 4. Efectos de aumentar la constante K_i en un sistema de control PID.

Fuente: Elaboración propia.

Acción (Aumentar K_i / Disminuir K_i)	Acción de control proporcional K_i
Aumentar K_i	Se aumentará la inestabilidad del sistema.
Aumentar K_i	Se aumentará en una cantidad pequeña la velocidad del sistema.
Aumentar K_i	Esta acción de control servirá para disminuir el error en un régimen permanente.
Aumentar K_i	Se disminuirá el error del sistema en régimen permanente.

Acción derivativa

Tal como su nombre lo menciona, la acción de control derivativa es meramente proporcional a la derivada de la señal de error $e(t)$ y a su vez multiplicada por la constante denominada por las letras K_d . La derivada del error tiene otra forma muy peculiar de conocerse y es la de la "velocidad" del error, por lo que es muy importante conocer esta velocidad, ya que, si tomamos en cuenta que cuando el valor real se encuentra por debajo del valor deseado, la acción proporcional se

Capítulo 2. Marco teórico (Antecedentes).

encarga de aumentar para poder subir ese valor y llegar a la posición correcta. Pero el problema se da cuando se toman en cuentas las inercias que puede llegar a tener el sistema debido a la acción integral, es decir que, cuando el sistema está moviendo a una alta velocidad con dirección al punto de referencia, el sistema se pasará de ese punto de referencia como consecuente de la inercia en el propio sistema. Por lo tanto, al tener esa presencia de inercia en el sistema se puede producir un sobre pulso y oscilaciones en alrededor de la referencia. Es por ello que, para poder evitar este problema, el controlador tiene que tener la capacidad de reconocer la velocidad a la que el sistema se va acercar a la señal de referencia para poder tener ese frenado de la señal con anticipación conforme va llegando a la referencia deseada y así poder evitar que se sobrepase el valor real del valor deseado.

De la misma manera que la acción proporcional e integral, al tratar de sintonizar el controlador PID y aumentamos la constante de acción proporcional K_d , vamos a obtener las siguientes consecuencias como se puede ver en la Tabla 5 :

Tabla 5. Efectos de aumentar la constante K_d en un sistema de control PID.

Fuente: Elaboración propia.

Acción (Aumentar K_d / Disminuir K_d)	Acción de control proporcional K_d
Aumentar K_d	El error en régimen permanente se quedará en el mismo valor.
Aumentar K_d	Se aumenta la estabilidad del sistema que se esté controlando.
Aumentar K_d	Esta acción de control será de gran ayuda para poder estabilizar una respuesta que presente demasiadas oscilaciones.
Aumentar K_d	Se logrará disminuir en una pequeña cantidad la velocidad del sistema.

Arduino IDE

El Arduino IDE es denominado así debido a sus siglas en inglés las cuales significan Integrated Development Environment, que en español se conocería como entorno de desarrollo integrado. El Arduino IDE no es más que un programa con fines informáticos que es conformado por un conjunto de herramientas de programación. Este programa cuenta con la capacidad de enfocarse en su totalidad a un solo lenguaje de programación o en el mejor de los casos puede ser utilizado para varios.

El Arduino IDE es también conocido por ser un ambiente de programación, que desde su creación ha sido utilizado como un tipo de programa de aplicación, es decir, que su creación involucra en primer lugar a un editor de código, un compilador del programa, así como un depurador y un creador de la interfaz gráfica. Por lo tanto, en el caso de utilizar el microcontrolador de Arduino, el IDE es el encargado de agregar las herramientas para poder editar y luego cargar el programa ya compilado, el cual es guardado en la memoria flash del propio hardware del microcontrolador. Es por ello que, El IDE de Arduino es la herramienta encargada de trabajar a la par con las placas de hardware de Arduino, por lo tanto, llevan el mismo logotipo, el cual se puede ver en la Figura 12 .



Figura 12. Logotipo de Arduino

Fuente: [6].

Optoacoplador EL817

En pocas palabras este es uno de los elementos electrónicos que se utilizó en la creación del circuito eléctrico, el cual se puede observar en la Figura 13. Su utilización fue gracias a que su función es la de servir como un interruptor, para lo cual se puede aislar de manera electrónica el circuito eléctrico, gracias a que es un fototransistor en el cual, lleva acoplado un diodo infrarrojo.

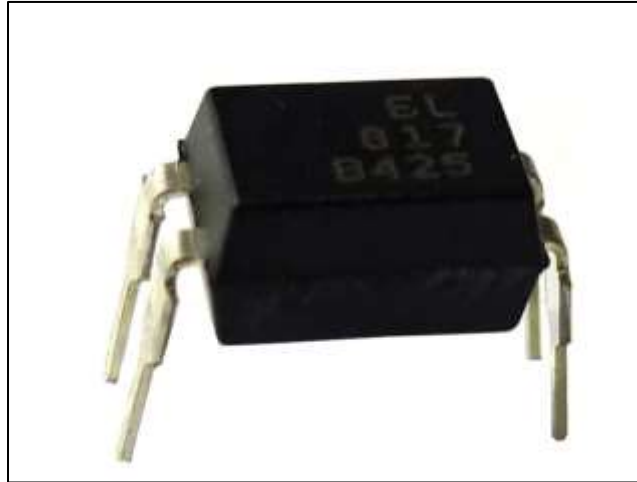


Figura 13. Representación de un Optoacoplador fototransistor EL817.

Fuente: [7]

Puente de diodos de onda completa kbpc1010

Este elemento se le conoce también como puente rectificador de diodos y se puede ver en la Figura 14, el cual funciona como un circuito rectificador de onda completa, debido a que, se encarga de transformar una señal de voltaje senoidal, es decir, con presencia en la señal positiva y negativa a la vez, en una señal únicamente de carácter positivo. Este puente de diodos es diferente su aprovechamiento a lo de un solo diodo, debido a que, un solo diodo es capaz de quedarse con la parte de carácter y desechar la parte negativa, pero el puente es capaz de utilizar y darle ese aprovechamiento a la parte negativa.

El puente rectificador es un componente electrónico que es capaz de transformar una señal de corriente alterna a una señal de corriente directa, pero para ello se

Capítulo 2. Marco teórico (Antecedentes).

debe de recalcar que un puente rectificador no puede realizar esto por sí solo, por lo tanto, para lograr esa conversión de corriente alterna a corriente directa, se deben de utilizar otros componentes, que al juntarlos y trabajando a la par del puente rectificador de diodos, ayudarán a que se cumpla esta transformación de la señal.



Figura 14. Puente de diodos rectificador de onda completa KBPC1010.

Fuente: [8]

Diodo MOC3020

El MOC3020M es un diodo conformado de arseniuro de galio (GaAs) tal como se puede ver en la Figura 15 el cual tiene la capacidad de emitir infrarrojos que son ópticamente acoplados a un interruptor bilateral de silicio, por lo que es activado por luz y funciona como un TRIAC. Este diodo se puede considerar como aquel dispositivo que fue creado para su implementación en aplicaciones que requieran de un disparo en un TRIAC aislado.

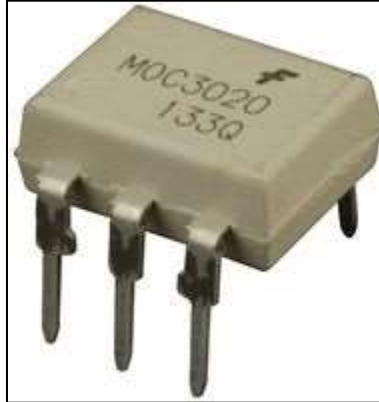


Figura 15. Representación de un diodo MOC3020.

Fuente: [9]

Transistor TRIAC BTA16-600B

Este tipo de elemento electrónico, el cual se puede ver en la Figura 16, es un transistor que se utiliza para controlar el flujo de corriente hacia una carga, con la particularidad de que es capaz de trabajar en los dos sentidos de la señal y a su vez, puede llegar a ser bloqueado y eso se logra invirtiendo la tensión o simplemente al minimizar la corriente a un valor que se encuentre por debajo del valor de mantenimiento. Por lo tanto, el TRIAC por tener la capacidad de trabajar en los dos sentidos de la señal, se puede activar su disparo sin importarle la polarización de compuerta, gracias a la corriente tanto negativa como positiva.

Este tipo de transistor es conocido como un TRIAC de propósito general, por lo que es muy común y correcta su utilización para actuar como dispositivo de conmutación en la corriente alterna. Sin embargo, puede utilizarse como un switch de encendido y apagado, así como también para variar la cantidad de potencia que se le suministra a la carga que se le haya conectado, gracias a que se le puede ir variando su ángulo de disparo.

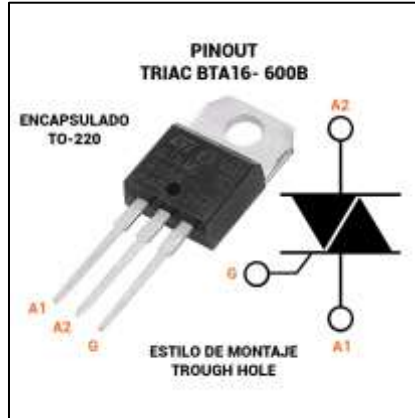


Figura 16. Representación de un transistor TRIAC BTA16-600B.

Fuente: [8].

Alambre de resistencia Kanthal A-1.

El alambre Kanthal A-1 es una aleación ferrítica compuesta por hierro, cromo y aluminio (FeCrAl), el cual se puede llegar a utilizar usar a altas temperaturas de hasta los 1400 grados Centígrados, que en grados Fahrenheit son 2550 grados. Este alambre se caracteriza por tener una alta resistividad y una muy buena resistencia a la oxidación gracias a la aleación que lo conforma. Este tipo de alambre Kanthal A-1, tiene generalmente su campo de aplicación en elementos de calentamiento eléctrico como si fueran resistencias dedicadas a los hornos de alta temperatura en las industrias dedicadas a los tratamientos térmicos, fundiciones y trabajos de vidrio, acero, cerámica y por supuesto en aplicaciones de electrónica y se puede ver el tipo de alambre que es en la Figura 17.



Figura 17. Representación del alambre Kanthal A-1.

Fuente: [10].

Termopar tipo K

El termopar tipo K también se le conoce como termocupla, el cual consta de un sensor de temperatura que funciona como un tipo transductor debido a que, funciona con dos metales distintos, los cuales producen una diferencia de potencial, es decir, la diferencia de temperatura que existe entre los dos metales, ya que, uno de los extremos se le denomina como punto caliente y el otro se le denomina con el nombre de punto frío. Este termopar es capaz de medir altas temperaturas, es decir, desde los 0 hasta los 1300°C, pero todo depende del calibre del cable del termopar puede llegar a medir menos temperatura. Sin embargo, se puede presentar de muchas maneras y en la se muestra una de ellas.



Figura 18. Termopar tipo K.

Fuente: [11].

Módulo MAX6675

El módulo de interfaz MAX6675 se utiliza específicamente para el termopar tipo K, el cual nos da la capacidad de acondicionar la señal analógica entregada por algún tipo de sensor, el cual sería de tipo transductor como el termopar tipo K, para que esta señal pueda ser captada y a su vez interpretada de forma digital y así se pueda utilizar algún microcontrolador como el Arduino.

Dentro del módulo del MAX6675, se localiza la electrónica necesaria para poder amplificar, compensar y transformar la señal analógica a una señal digital el voltaje

Capítulo 2. Marco teórico (Antecedentes).

que haya sido demandado por el termopar, por lo que se hace muy sencilla la manera de conectar un sensor de temperatura a algún un microcontrolador. El módulo de interfaz MAX6675 para termopar tipo K es una buena herramienta que nos facilita medir altas temperaturas con la ayuda de cualquier tipo de microcontrolador. Un ejemplo de este módulo se puede observar en la Figura 19.



Figura 19. Módulo de interfaz MAX6675.

Fuente: [12].

Módulo con interfaz I2C

Este módulo con interfaz serial I2C, te permite controlar una pantalla LCD de una forma muy sencilla, por ejemplo, en lugar de conectar 14 pines en una LCD 16x2, solamente tendrías que conectar 4 pines; GND, VCC, SDA, SCL. Por lo tanto, en el hecho de que no usarán tantos pines para conectar este módulo con una pantalla LCD, sobrarán muchos pines en la placa de Arduino, por ejemplo.

Este módulo puede llegar a hacer que un proyecto se convierta en algo único y bastante sencillo de montar, por lo que, si el proyecto se basa en el microcontrolador de Arduino, el cual tendrá la capacidad de utilizar este módulo con total confianza.

En este módulo se cuenta con un chip I2C PCF8574, el cual transforma los datos

en serie con comunicación I2C a datos paralelos para una pantalla LCD, un ejemplo de este módulo se puede observar en la Figura 20.

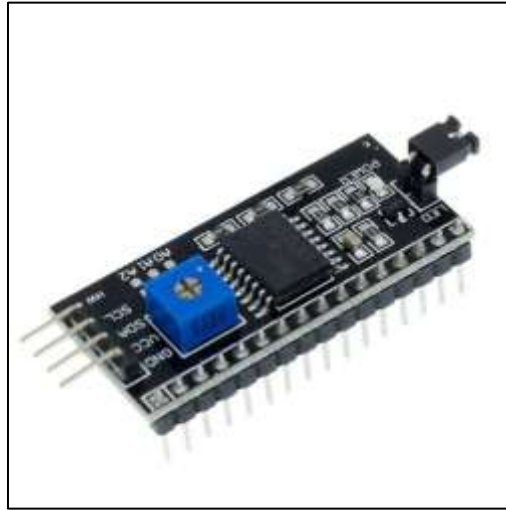


Figura 20. Módulo con interfaz I2C

Fuente: [8].

Pantalla LCD 16x2

Este tipo de pantalla se refiere a un dispositivo con pantalla compuesta de cristal líquido, tal como se muestra en la Figura 21, la cual cuenta con dos filas que contienen dieciséis caracteres cada una de ellas, las cuales se utilizan para mostrar información del programa o proceso que se ejecutando, que por lo general es una información alfanumérica. Este tipo de dispositivos cuenta con unas capacidades muy altas por la sencilla razón de que se puede mostrar información de cualquier tipo prácticamente, es decir, sin importarle qué tipo de caracteres o símbolos sean, el idioma o hasta el tipo de lenguaje, esto debido a que, el sistema es capaz mostrar cualquier tipo de carácter alfanumérico, símbolos y hasta varias figuras, sin embargo, el número de píxeles que puede llegar a tener cada tipo de carácter o símbolo puede variar dependiendo del modelo de la pantalla LCD que se esté utilizando y se le agrega que cada uno de estos dispositivos está siendo controlado por algún un microcontrolador, el cual ya está previamente programado para controlar el funcionamiento y a su vez también la imagen que va a ser presentada en la pantalla.

Capítulo 2. Marco teórico (Antecedentes).

Existen varias ventajas al utilizar estos dispositivos, por ejemplo, que tienen un consumo muy bajo de energía, así como la programación resulta ser muy fácil de realizar además de que, el usuario es quien carga el programa al microcontrolador que lo esté controlando.

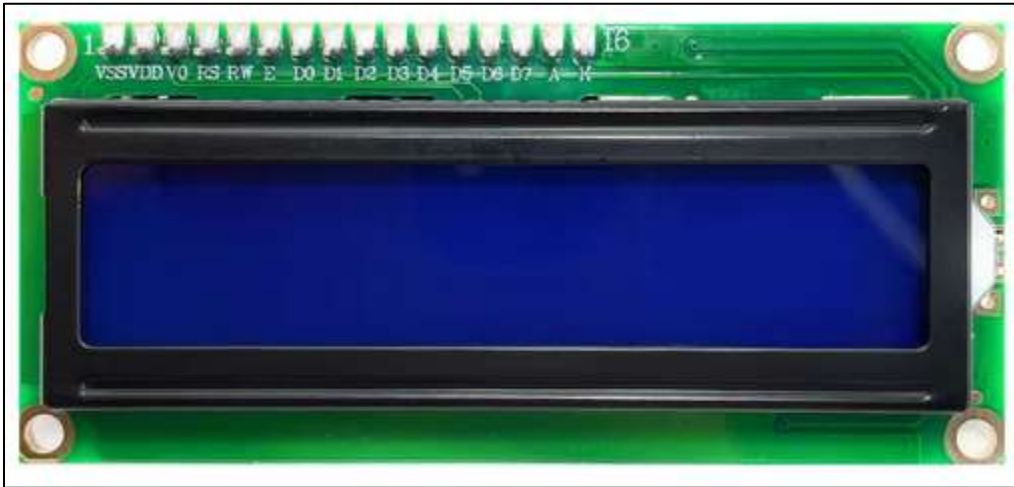


Figura 21. Pantalla LCD 16x2.

Fuente: [13].

Capítulo 3

Planteamiento del problema

3.1 Identificación.

En la actualidad el aluminio ha incrementado su uso en todos los sectores industriales por lo cual existe demasiada demanda en la producción de latas de aluminio que conlleva a una gran oportunidad de aprovechamiento de la regla de las tres erres, debido a que la producción de este tipo de latas de aluminio de la serie 3xxx ha tenido un incremento considerable y que lamentablemente como sociedad no se tiene esa cultura de ver por el daño al medio ambiente al no aprovechar al máximo este tipo de latas como por ejemplo, dándoles una nueva función después de fundirlas, por lo que con este tipo de horno académico lo que se busca es el poder darles no sólo esa segunda ni tercera oportunidad, sino todas las necesarias para poder tener un gran impacto y ayudar en el cuidado del medio ambiente y a su vez, en el tema económico poder ayudar a los demás estudiantes del ITSUR regalándoles la oportunidad de realizar trabajos de fundición y/o tratamientos térmicos dentro de la institución sin ninguna limitante y sin tener que gastar un solo peso debido a que se utilizaría aluminio reciclado y no tendrían la necesidad de tener que estar gastando en comprar material para sus prácticas, además de que los estudiantes pueden aprender y generar nuevos conocimientos y experiencia en el área de función y tratamientos térmicos para futuros trabajos ya que en la industria automotriz es de los materiales más utilizados.

3.2 Justificación.

Como consecuencia de la identificación del problema se llegaron a las grandes oportunidades que nos traería consigo la fabricación de este horno académico en muchos ámbitos, es decir, oportunidades grandiosas para los alumnos, maestros,

Capítulo 3. Planteamiento del problema

institución, como para instituciones que estén alrededor o que quieran realizar convenios con el ITSUR para que puedan trabajar a la par con proyectos de reciclaje de aluminio para su posterior fundición y estar conectados por algún objetivo en común, el cual sería el aprendizaje de los estudiantes de ambas partes involucradas.

La idea del horno académico nace de dos profesores Rogelio Orozco Martínez (profesor de Ing. en Sistemas Automotrices) y Mariano Braulio Sánchez (Coordinador de Ing. en Sistemas Automotrices), quienes de acuerdo a su experiencia dando clases en la carrera y a las limitantes que se les han presentado en los laboratorios a la hora de realizar prácticas con sus estudiantes, llegaron a la conclusión de que se necesitaba un horno tipo mufla para poder potenciar el aprendizaje de los estudiantes, debido a que sólo se cuenta con dos muflas en la institución, pero solamente una de ellas pertenece a la coordinación de Sistemas Automotrices y la otra pertenece a la coordinación de Industrial, lo que hace muy complicado el uso de esta mufla perteneciente a la carrera, porque tanto los procesos de tratamientos térmicos como de fundición suelen ser la mayoría de las veces algo tardados y no todos los estudiantes alcanzan a realizar estas prácticas. Sin embargo, se lograría un gran impacto ambiental debido a que, se podrían realizar campañas de reciclaje de aluminio de la serie 3xxx por parte de los alumnos para que se consiga suficiente material para sus prácticas y siempre puedan estar poniendo en marcha sus conocimientos donde poco a poco vayan adquiriendo más y mejores aprendizajes, y que mejor que sin tener que gastar en comprar material y poder realizar la mayor cantidad de prácticas que muchas veces por la cuestión económica no son posibles de realizar.

3.3 Alcance.

Se diseñará un prototipo de horno académico basado en un controlador PID que sea capaz de regular las temperaturas que el usuario requiera y a su vez los distintos ciclos que se necesiten en el proceso de fundición, siendo capaz de llegar

Capítulo 3. Planteamiento del problema

a altas temperaturas, sirviendo como modelo académico en la institución para luego ver si tiene una buena aceptación por parte de los beneficiarios y a su vez poder implementarlo en diferentes instituciones como material didáctico y generar esa cultura de utilización de la regla de las tres erres en el ITSUR en cuanto a latas de aluminios de serie 3xxx se refiere.

Capítulo 4

Objetivos

4.1 Objetivos generales.

Analizar los fundamentos de un controlador proporcional integral derivativo (PID) y diseñarlo para lograr un control programable de temperatura para su integración a un horno académico.

4.2 Objetivos específicos.

- a) Realizar un circuito capaz de medir la temperatura dentro de un horno eléctrico con ayuda del termopar tipo K y mostrarla en el monitor serial de Arduino IDE.
- b) Implementar una pantalla capaz de mostrar la temperatura que hay dentro de un horno eléctrico para el monitoreo de la misma.
- c) Elaborar el circuito eléctrico con los componentes que conformará el controlador PID.
- d) Crear y diseñar el funcionamiento del controlador PID.
- e) Calcular y obtener el análisis del sistema (horno eléctrico) para la obtención de las constantes del controlador PID que se manejan en la función de transferencia de dicho sistema.
- f) Integrar el controlador PID en el horno académico.

Capítulo 5

Metodología

Este proyecto de residencias tuvo lugar en las instalaciones del Instituto Tecnológico Superior del Sur de Guanajuato, específicamente en salones del edificio B, el centro de cómputo ubicado en el edificio de TIC´S, así como también en el laboratorio de impresos, manufactura y máquinas eléctricas ubicados en el edificio más reciente de la universidad el cual es el E.

El proyecto se trabajó durante un periodo de seis meses, es decir, desde enero hasta junio 2023 como parte de la materia de residencias profesionales, la cual tiene como objeto el formar estudiantes capaces de aplicar los conocimientos obtenidos a lo largo de su carrera y puedan desarrollar esa experiencia que se requiere en un futuro y así poder egresar de la institución.

Es por ello que estos dos profesores Rogelio Orozco Martínez (profesor de Ing. en Sistemas Automotrices) y Mariano Braulio Sánchez (Coordinador de Ing. en Sistemas Automotrices) plantearon este problema como un reto a realizar por 4 estudiantes que estuvieran próximos a realizar residencias profesionales, es decir que el horno académico sirviera como modelo académico y a su vez, en un futuro los institutos tecnológicos e incluso otras instituciones adopten este modelo académico y sus estudiantes también sean capaces de fabricar sus propias herramientas para sus prácticas y no tener que comprarlas a las grandes empresas por precios muy elevados, buscando que los estudiantes vayan obteniendo esas habilidades y sobre todo la experiencia necesaria que en la vida de ingeniero se necesitan.

Como bien se mencionó este proyecto se planteó a 4 estudiantes, es decir, 2 estudiantes se encargarían de la parte de la construcción de la mufla, desde el diseño y estructura hasta el prototipo final y los otros 2 estudiantes se encargarían de la parte del controlador PID, el cual es el principio de funcionamiento del horno académico. Sin embargo, el proyecto se trabajaría en conjunto con los 4 estudiantes

para lograr el buen trabajo en equipo y la excelente lluvia de ideas en cada uno de los procesos que implicara el proyecto, donde los estudiantes encargados de la construcción del horno como tal se implicaran también en el procesos del controlador PID y a su vez, los estudiantes encargados del diseño y funcionamiento del controlador PID también se involucraran en el diseño del horno como tal, así como también en la elección de materiales y cálculos necesarios para las eficiente transferencia de calor en cada una de las paredes interiores de la cámara refractaria.

Semana 1 y 2 (23/Enero/23 – 03/Febrero/23)

- **Planteamiento del proyecto con el asesor externo y coordinador de la carrera.**

En la primera semana del semestre se estuvo trabajando la idea del proyecto con los dos profesores que tuvieron esta iniciativa, en la cual se dio a conocer la identificación del problema, seguido de la justificación del mismo, así como el alcance que pudiera tener el proyecto. Esta idea se convirtió en un proyecto muy interesante donde se vio la gran oportunidad de poder demostrar los conocimientos obtenidos durante la carrera, ya que, si el proyecto se realizara de la mejor manera y consiguiendo los resultados esperados, sería un gran impulso para poder convertir este proyecto de residencias profesionales en un excelente informe y lograr ese título profesional que tanto se esperaba. Sin embargo, los profesores comentaron que en su caso el trabajo pasaría directo a tesis por la complejidad y el impacto que este podría tener para la institución y que, solamente se podría realizar en parejas la tesis o individual, por lo que se optó por dividir el equipo de trabajo de 4 personas en parejas y cada una de las partes dedicarse a una tarea en específico, es decir, una pareja estaría llevando a cabo el proceso de la parte de manufactura como parte de la especialidad que se tomó en la carrera de sistemas automotrices y de igual manera, la otra pareja del equipo se estaría dedicando más al proceso tecnológico en cuanto a programación. Sin embargo, se estaría trabajando a la par con los demás compañeros del equipo en todos los procesos que el proyecto

requiera.

Semana 3 y 4 (06/Febrero/23 – 17/Febrero/23)

- **Estudio del mercado para el presupuesto y participación en convocatoria de proyectos de desarrollo tecnológico e innovación para estudiantes.**

Debido a la gran problemática que se presentó por el presupuesto requerido para poder desarrollar el proyecto, surgieron diferentes alternativas para poder solventarlo. Una de ellas involucraba un presupuesto que se solicita con anticipación de un año en la coordinación de sistemas automotrices para material metálico y no metálico, plástico y cerámico, la cual era en su momento la alternativa más viable si se pensaba que este año se había autorizado dicho presupuesto, pero no fue así en un principio, debido a que, el coordinador comentó que aún no le resolvían si se autorizó o no, por lo que en conjunto con el coordinador y el asesor externo, se pensaron otras maneras de bajar ese recurso para poder comenzar con el proyecto, y la siguiente idea fue la de participar en la convocatoria del Tecnológico Nacional de México que lleva el nombre de “Proyectos de Desarrollo Tecnológico e Innovación para Estudiantes” donde se otorgaría una cantidad considerable para material y suministros y así poder realizar el prototipo del horno académico basado su funcionamiento en un controlador PID. Por lo que en estas dos semanas se trabajó en dicha convocatoria, la cual pedía algunos documentos; introducción del proyecto, marco teórico, planteamiento del problema, antecedentes, objetivos tanto general como específicos y también el estudio del mercado (presupuesto requerido) abarcando los 30mil pesos exactos que se expedían en la convocatoria. Estos documentos se subieron de manera individual uno por uno en cada una de las asignaciones que pedía la plataforma antes de la fecha límite, la cual era el 17 de febrero y claramente se subieron antes de la fecha solicitada, la convocatoria se podía encontrar en la página del Tecnológico Nacional de México, tal como se muestra en la Figura 22.



Figura 22. Convocatoria del Tecnológico Nacional de México.

Fuente: [14].

Semana 5 y 6 (20/Febrero/23 – 03/Marzo/23)

- **Análisis de funcionamiento de la mufla en cuanto a programación y el cómo se comporta a diferentes temperaturas.**

Durante estas dos semanas se trabajó en el análisis de la mufla que se tiene en la coordinación de Sistemas Automotrices para lograr comprender su funcionamiento y para eso no había más que el tener que realizar diversas pruebas de fundición con latas de aluminio serie 3xxx recicladas, ya que para este momento aún no se autorizaba el presupuesto de la convocatoria donde se participó, es decir, aún no nos daban resultados, por lo que se tuvo que aprovechar el tiempo de otra manera y se realizaron 3 tipos de pruebas para ver cómo funcionaba la mufla en cuanto a su programación para lograr detectar que tipos de ciclos tiene.

En la primera prueba se cortaron las latas de aluminio en pedazos muy pequeños para poder ingresar una gran cantidad de material en el crisol de grafito de carburo de silicio que se tiene en la coordinación, sin haber hecho un proceso de lijado de las latas para quitarles la pintura ni nada similar, tal como se puede ver en la Figura 23.



Figura 23. Primera prueba sin lijar las latas y cortadas en pedazos pequeños.

Fuente: Elaboración propia

Posteriormente se realizó otra prueba de fundición de latas de aluminio donde se lijaron las latas para desprenderle la pintura que trae alrededor y de igual manera se cortaron en pedazos pequeños para que también ingresara más material en el crisol, tal como se muestra en la Figura 24.



Figura 24. Segunda prueba lijando las latas y cortadas en pedazos pequeños.

Fuente: Elaboración propia.

Después se realizó una tercer prueba donde se lijaron las latas y se recortó únicamente el cuerpo de la lata, es decir, sin la tapa que tiene en la parte de arriba y sin la tapa que también trae en la parte baja de la lata, tal como se muestra en la Figura 25, para observar cómo se comportaba la mufla en cada una de las pruebas y a su vez, también estudiar al aluminio y poder observar en cuál de las pruebas se obtenía un aluminio mucho más puro, ya que como es parte de la metodología experimental se pretendía tener esa referencia sobre que tanto porcentaje de pureza contienen las latas de aluminio de la serie 3xxx.



Figura 25. Tercera prueba lijando y utilizando solo el cuerpo de las latas sin la tapa.

Fuente: Elaboración propia.

Se analizó por completo la estructura y componentes de la mufla que se tiene en la coordinación de sistemas automotrices. Sin embargo, este análisis costó mucho debido a que, se tuvo que recurrir a la linterna del celular, para poder visualizar de mejor manera en la parte donde está el controlador en la mufla y eran unas rendijas muy pequeñas, por lo que se complicó en gran manera el poder observar bien a detalle que componentes la conformaban y poder entender el funcionamiento de dicha mufla para su posterior creación del controlador PID que se ocuparía en este proyecto.

Semana 7 y 8 (06/Marzo/23 – 17/Marzo/23)

- **Calcular y obtener el análisis del sistema (horno eléctrico) para la obtención de las constantes del controlador PID que se manejan en la función de transferencia de dicho sistema.**

Una vez que se observó el cómo se comporta la mufla que se tiene en la coordinación ante cada una de las 3 pruebas de fundición, se realizó una delicada investigación acerca de los principios básicos de un controlador PID y todo lo que se requeriría después para poder realizar uno e integrarlo en nuestro horno académico, es decir, se buscaron diferentes videos en páginas como YouTube

acerca de un control PID, de los cuales uno que fue de mucha ayuda fue el video con el nombre de “02. control PID explicación sencilla”, el cual se muestra en la Figura 26, este video sirvió sobre todo para recordar la clase que se llevó durante la carrera la cual era la materia de “control” y poder tener la información mucho más fresca sobre, ¿Qué es el control PID?, así como también, el ¿para qué sirve? y el ¿cómo funciona?, por lo que el tema dio para adentrarse más en subtemas que fueron surgiendo y que, para mejor organización y distribución de la información que se iba recabando, se realizó una extensa búsqueda tanto de videos como de páginas de internet para después que se llegaron a ocupar como en el caso de los videos se realizó una recopilación y así tener guardados los videos que ayudarían en caso de quedarse trabados en algunas cuestiones o problemáticas que presentara el proyecto, asimismo, las páginas de internet que más fueron de utilidad se les guardó en la misma pestaña de favoritos en la página de Google para su posterior chequeo si se llegara a dar una duda en algún tema investigado. Sin embargo, no solamente se quedó en videos y páginas de internet, sino que se consideró la investigación en un libro principalmente que fue “Ingeniería de control moderna” de Ogata para obtener información valiosa del mismo controlador PID, desde sus antecedentes, funcionamiento y su diseño.



Figura 26. Video acerca de los fundamentos de un controlador PID.

Fuente: [15].

Semana 9 y 10 (20/Marzo/2023 – 31/Marzo/2023)

- **Determinar los componentes requeridos del controlador PID.**

Se comenzó con la planificación acerca de cómo determinar y cómo obtener los componentes necesarios del controlador PID basándose en videos de YouTube donde se explicaban diversos sistemas que utilizaban algún tipo de controlador PID y sobre ello, se fueron deduciendo los diversos componentes que serían de gran ayuda para poder realizar dicho controlador con base a la experiencia en microcontroladores como Arduino e instrumentación digital. Los videos que más fueron de ayuda fueron los llamados “Determinar los componentes requeridos del controlador PID”, “Pr#30 Control PID de temperatura DC” y “Pr#49 Control PID temperatura 220V AC”, siendo el segundo vídeo el que más se asemejaba a lo que se pensaba realizar en este proyecto, el cual se puede ver en la Figura 27, aunque de manera general en estos tres videos se podía encontrar una breve explicación y muy detallada del cómo se podría llevar a cabo un controlador PID en cualquier tipo de sistema que se le pueda ocurrir a cualquier persona por lo que, sirvió de guía para poder deducir los componentes a necesitar. Sin embargo, una vez determinados los componentes a necesitar, tocó la parte de buscar de manera inteligente ya que se debían encontrar los elementos que cumplieran con todos los aspectos necesarios para poder formar parte del controlador PID, esto debido a que, un solo elemento podría tener cientos de variaciones y modelos, que a pesar de que muchas veces fueran muy mínimas, aun así son capaces de marcar la diferencia y fue por ello que se realizó una minuciosa búsqueda respaldada en una previa investigación tanto en páginas web confiables como Amazon, Mercado Libre como en las páginas oficiales de los proveedores de la marca, así mismo, en grupos de Facebook donde vendieran estos tipos de elementos de instrumentación electrónica en los alrededores de Moroleón, Uriangato y hasta en Yuriria, por lo que, se encontró un local en el municipio de Moroleón, Guanajuato llamado “Al-Bri” donde venden todo este tipo de

componentes.



Figura 27. Video acerca de un controlador PID con una corriente DC.

Fuente: [16].

Después de todo se localizaron estos elementos específicamente en la página de Mercado Libre cumpliendo con todas las especificaciones necesarias por parte de cada uno de los componentes del controlador PID, para cumplir con el esquema que se creó previamente una vez que se determinaron los materiales a utilizar tal como se muestra en la Figura 28.

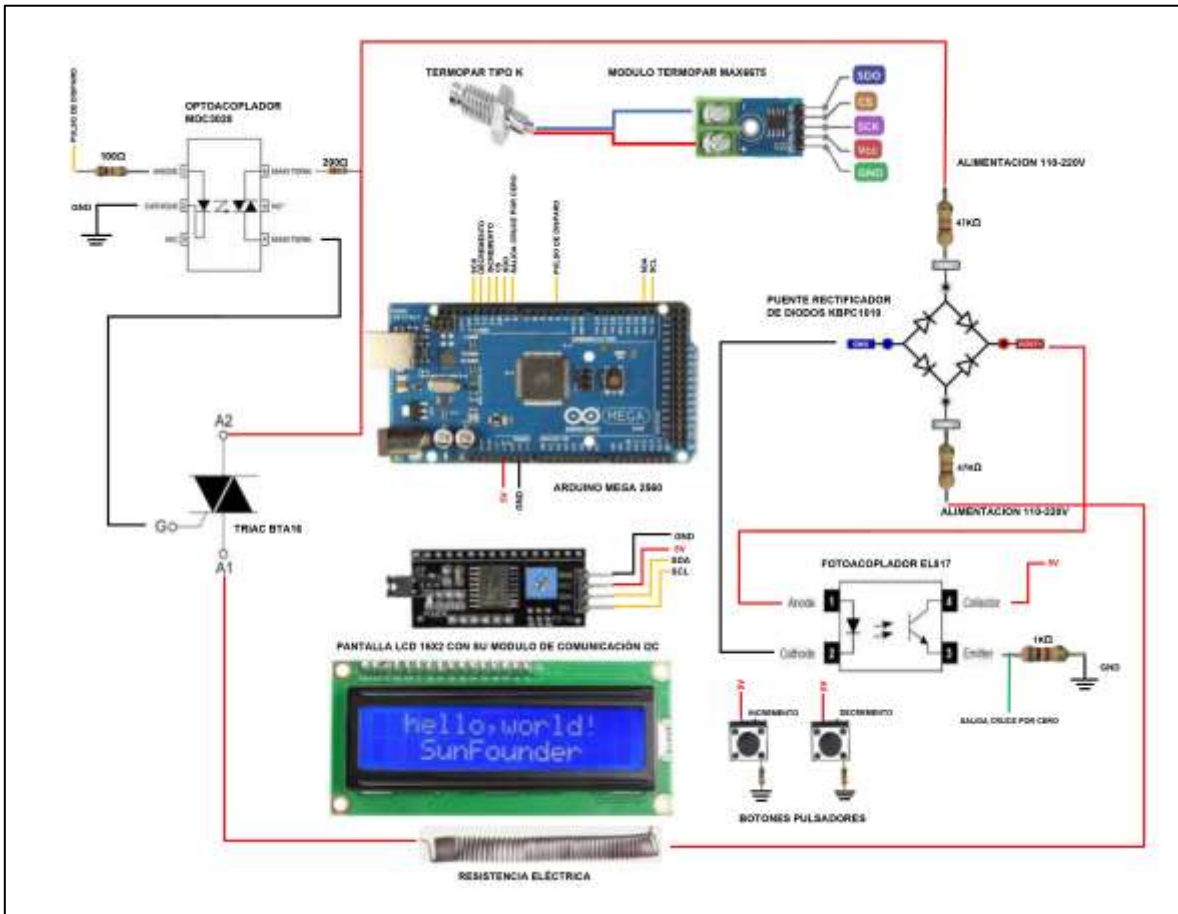


Figura 28. Esquema del circuito eléctrico del controlador PID.

Fuente: Elaboración propia.

Semana 11 y 12 (17/Abril/2023 – 28/Abril/2023)

- Realizar pruebas con un termopar tipo K utilizando el monitor serial de Arduino para el monitoreo de la temperatura.

Durante estas dos semanas se trabajó en pequeñas pruebas de conexión entre la placa de Arduino Mega 2560 y el sensor de temperatura LM35 para poder realizar un pequeño y muy sencillo código que nos pudiera mostrar la temperatura real de la habitación donde se estaba poniendo a prueba para posteriormente poner en práctica esta misma prueba, pero en lugar de realizarlo con este sensor LM35, se realizaría con un termopar tipo K capaz de medir temperaturas máximas de hasta los 1300°C.

Capítulo 5. Metodología.

Para poder llegar a esas pruebas primero se investigó si en el Arduino IDE se requería de la instalación de alguna librería para poder realizar la conexión del termopar tipo K y así pudiera mandar los valores de la temperatura que este midiendo sin problemas, por lo que se realizó una previa investigación de las especificaciones de los componentes del controlador PID antes de su compra como ya se mencionó anteriormente y fue en la misma investigación que se encontró que el propio termopar tipo K necesitaba un módulo de interfaz llamado “MAXX6675”, el cual era el encargado de lograr esa conexión y comunicación del Arduino con el termopar, por lo tanto, se partió de ahí y se investigó a detalle sobre cómo hacer la conexión entre el módulo de interfaz y se encontró que si se necesitaba instalar una pequeña librería llamada “max6675.h”, la cual por su propio nombre y por lógica se sabe que era para poder lograr esa conexión del Arduino con el módulo de interfaz MAX6675 y este a su vez reciba los valores que está midiendo el termopar de tipo K y pueda mostrarlos en el monitor serial como primer opción, tal como se muestra en la Figura 29 , debido a que en las siguientes semanas se pensaba realizar esta misma prueba pero que los valores medidos por el termopar fueran mostrados en un LCD 16x2 modelo 1602A para una mejor visualización y así ir logrando un mejor circuito.



Figura 29. Prueba de monitoreo de temperatura en monitor serial con un termopar tipo K.

Fuente: Elaboración propia

Semana 13 y 14 (01/Mayo/2023 – 12/Mayo/2023)

- **Realizar pruebas con un termopar tipo K utilizando una pantalla LCD 16x2 para el monitoreo de la temperatura con Arduino.**

En estas dos semanas se realizó, la modificación del código utilizado en las semanas 11 y 12, ya que, en esas semanas, el programa de Arduino solamente era capaz de mostrar el monitoreo de la temperatura censada por el termopar tipo K en el monitor serial que viene dentro del mismo Arduino IDE, y en este caso se buscó que el programa fuera capaz de censar la temperatura con el mismo termopar tipo K, pero con diferencia de que no solamente se imprimieran los valores de la temperatura en el monitor serial, sino que, también se imprimieran en la pantalla LCD 16x2, por lo que se investigó la manera de realizar este enlace de la pantalla LCD con el Arduino y pudieran funcionar en conjunto y en tiempo real. Es por ello que, después de una buena investigación, se observó que para conectar dicha pantalla se necesitaba de 14 pines y aun así se realizó la conexión de esa manera, pero era un desorden en cuanto al cableado del circuito y al momento de estar moviendo el protoboard se estaban saliendo de su lugar los cables y provocaban

que la pantalla LCD 16x2 no funcionara de manera correcta, sino que, mandaba símbolos aleatorios como muestra de que no le estaba llegando la señal correcta por estar mal conectados, tal como se muestra en la Figura 30.

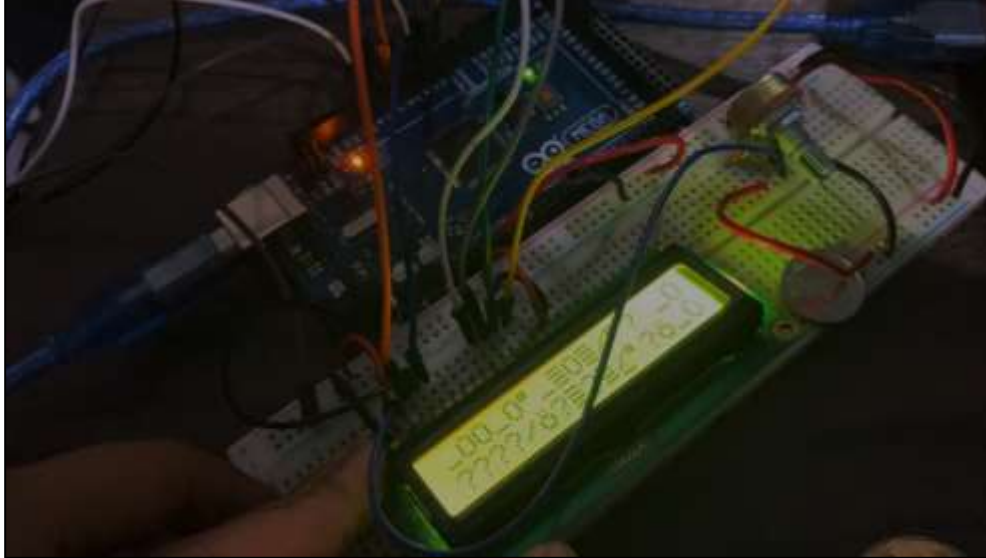


Figura 30. Pantalla LCD 16x2 mal conectada con sus 14 pines.

Fuente: Elaboración propia.

Por lo que se tuvo que tener más cuidado en dichas conexiones hasta que se logró que la pantalla LCD 16x2 estuviera arrojando de manera correcta los valores de la temperatura 1000ms, pero solamente se mostraban en Celsius, tal como muestra en la Figura 31.

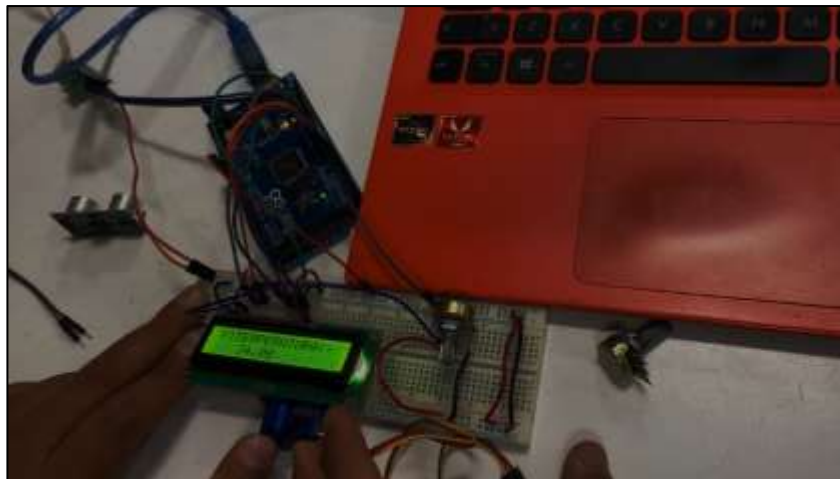


Figura 31. Pantalla LCD 16x2 bien conectada con sus 14 pines.

Fuente: Elaboración propia.

A pesar de que ya había funcionado ese monitoreo de temperatura en la pantalla LCD 16x2 con sus 14 pines, se investigó y encontró que había una manera evitar todo ese desorden de tantos cables conectados, la cual era la de conectarle un módulo de interfaz con comunicación I2C para una pantalla LCD 16x2, ya que, este módulo solamente ocupó conectarle 4 pines hembra-macho hasta el protoboard, ya que este módulo venía con una header de 14 pines que solamente se conectaba a los 14 agujeros de la pantalla LCD 16x2 y con esto simplemente se cargó el programa previamente realizado con la dirección correcta para que pudiera lograrse el enlace entre el Arduino con la pantalla y se puede ver en la Figura 32 como quedó mostrando la temperatura en Centígrados y Fahrenheit.



Figura 32. Monitoreo de la temperatura en una pantalla LCD 16x2 con modulo I2C.

Fuente: Elaboración propia.

Semana 15 y 16 (15/Mayo/2023 – 26/Mayo/2023)

- **Realizar pruebas de temperatura del controlador PID integrado a un prototipo de horno.**

En estas últimas dos semanas, se estuvo trabajando en la implementación del controlador PID, por lo que, primero se construyó el circuito eléctrico conectando todo a un solo protoboard de manera muy minuciosa ya que, cualquier elemento

Capítulo 5. Metodología.

que no esté bien conectado o que simplemente se conectara de una manera incorrecta, se podría causar algún problema y justamente eso es lo que no se quería, debido a que, el controlador se conectaría la corriente alterna del ITSUR que nos arroja 127V, pero al no funcionar con ese voltaje, se tuvo que buscar alguna otra alternativa y fue la de conectar el controlador a la corriente de 220VAC, tal como se muestra en la

Figura 33, por lo que era mucho más riesgoso pero se hizo con todos los cuidados posibles y con un SetPoint no tan alto, es decir, de 190°C que fue el que se acordó junto a maestros que estuvieron checando la prueba, ya que, si se llegara a establecer un SetPoint mucho más alto o en este caso el de 800°C, el cual sería el ideal para lograr una fundición de aluminio de la serie 3xxx, se hubiera corrido el riesgo de pasar algún accidente y para priorizar la seguridad se estableció aquel SetPoint de 190°C, simplemente para ver si el controlador PID era capaz de regular la corriente de 220VAC para alcanzar a estabilizar esa temperatura dentro del horno, por lo que se trabajó en las constantes de la función de transferencia de manera experimental para que cuando oscilara de una manera pasable, es decir, que la temperatura no bajara más de 10°C o subiera también más de 10°C del SetPoint, es por eso que cuando tenía oscilaciones de ese tipo se movían las constantes una por una hasta llegar a un ajuste mucho más limpio, en el cual no existieran oscilaciones muy marcadas y que mejor, que fueran más naturales casi nulas para lograr ese buen funcionamiento de un controlador PID.

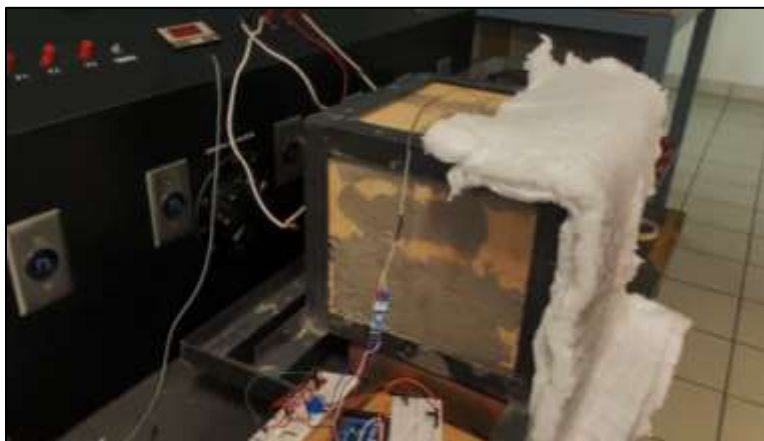


Figura 33. Controlador PID conectado a 220V y al horno académico.

Fuente: Elaboración propia.

Capítulo 6

Resultados

En esta sección se mostrarán los resultados obtenidos de manera enlistada con un orden cronológico desde que se comenzó con el proyecto, hasta los resultados que se consiguieron al final del mismo.

1. Primero se logró la creación de un programa en el software de Arduino IDE capaz de censar la temperatura dentro de un prototipo de horno eléctrico con ayuda de un termopar tipo K y su módulo de comunicación con Arduino, el cual es el MAX6675. Este programa se encargó de censar dicha temperatura y de la manera más sencilla posible se mostró dicho valor en el monitor serial del mismo programa, en el cual se estuvo actualizando cada 1000ms la temperatura. Así como se puede ver en la Figura 34, acerca de cómo se mantiene conectado el termopar tipo K dentro del prototipo del horno y con la conexión con el Arduino.

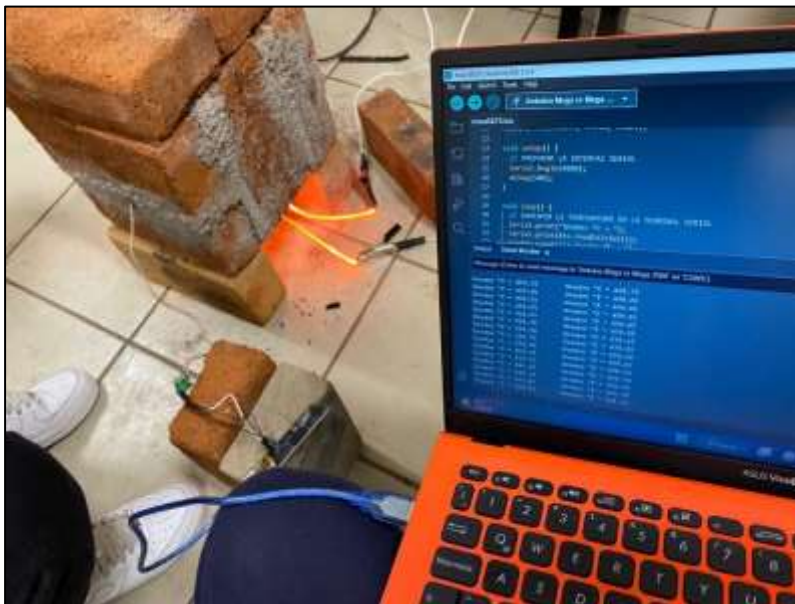


Figura 34. Programa con termopar tipo K que muestra la temperatura en el monitor serial.

Fuente: Elaboración Propia.

Capítulo 6. Resultados.

Cabe resaltar que el horno se encendió hasta después de haber conectado el Arduino a la computadora para cargarle el programa y pudiera existir esa comunicación con el termopar tipo K, el cual ya estaba colocado dentro del prototipo del horno eléctrico como se puede ver en la Figura 35.



Figura 35. Prototipo de horno eléctrico en funcionamiento.

Fuente: Elaboración propia

Después de haber compilado y subido el programa creado previamente para la lectura de la temperatura dentro del prototipo del horno eléctrico con el termopar tipo K, tal como se puede observar en la Figura 36, se procedió a conectar las dos resistencias colocadas dentro del horno a la corriente alterna de 127V, para poder comenzar la prueba de elevación y monitoreo de temperatura.

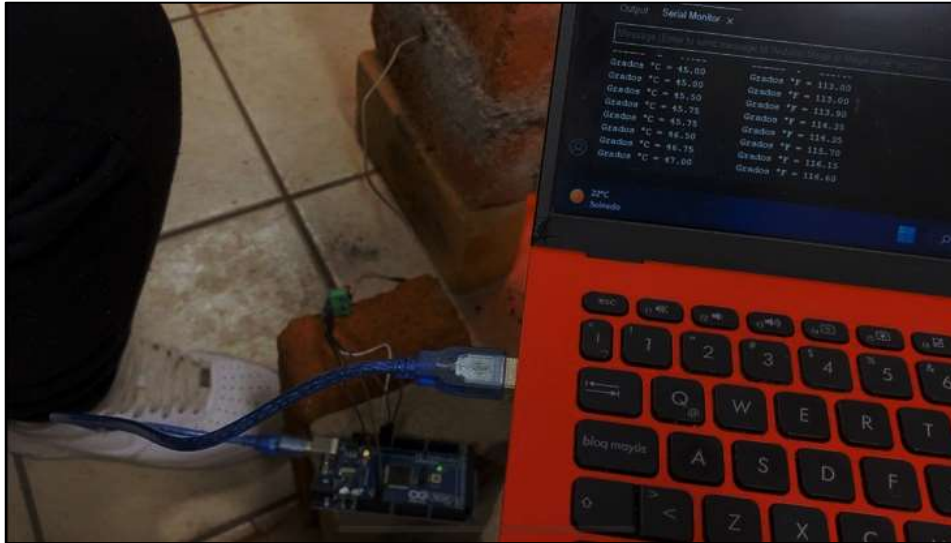


Figura 36. Conexión del Arduino con el horno eléctrico con el termopar tipo K.

Fuente: Elaboración propia.

Finalmente, se obtuvo una temperatura máxima de 538°C como se puede ver en la Figura 37, después de media hora de funcionamiento y con tan sólo 2 resistencias conectadas dentro del horno.

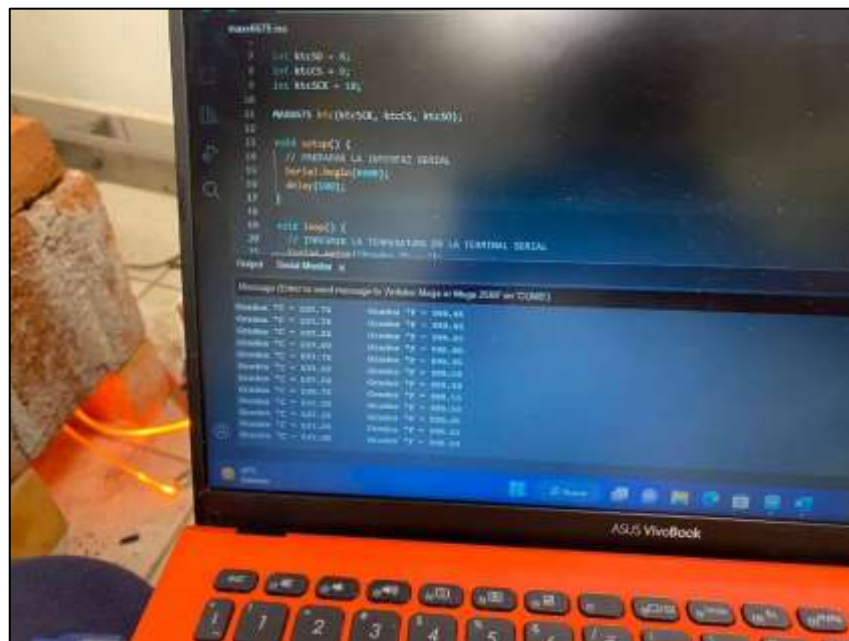


Figura 37. Temperatura máxima censada (538°C) dentro del prototipo del horno eléctrico.

Fuente: Elaboración propia.

2. Después de realizar el primer programa se consiguió la creación y el funcionamiento de un circuito eléctrico basado en medir la temperatura dentro de un horno eléctrico con ayuda del microcontrolador de Arduino Mega2560 capaz de mostrar el valor censado por un termopar tipo K con ayuda del módulo de MAX6675 tanto en grados centígrados como en Fahrenheit en una pantalla LCD 16x2 con un módulo de comunicación i2C. En este programa se colocó el mensaje de “TERMOPAR TIPO K MAXX6675 ARDUINO” en la LCD al inicio cuando va comenzar a censar la temperatura como se puede ver en la Figura 38.



Figura 38. Mensaje de inicio al comenzar a censar la temperatura.

Fuente: Elaboración propia.

Después de haber pasado 3000ms desde que se subió el programa a la placa de Arduino se comenzó a censar la temperatura como se muestra en la Figura 39.



Figura 39. Pantalla LCD mostrando el aumento de temperatura.

Fuente: Elaboración propia.

Finalmente, se puede ver en la Figura 40 que, mientras más siga subiendo la temperatura dentro del prototipo del horno eléctrico, se mostrará ese aumento en la pantalla LCD 16x2 de manera automática tanto en grados Centígrados como en Fahrenheit.



Figura 40. Pantalla LCD evidenciando el correcto funcionamiento del termopar tipo K.

Fuente: Elaboración propia.

3. Se estableció el esquema a realizar como prototipo final en cuanto a los componentes del circuito eléctrico se refiere en el controlador PID, tomando en cuenta que se tiene que trabajar con la corriente alterna de 220V y con el microcontrolador Arduino Mega2560 a la par, por lo que se tuvo que investigar cómo proteger ese paso de corriente de 220V a los 12V que es el voltaje al que trabaja el Arduino. Por lo tanto, se realizó la conexión con los componentes electrónicos necesarios para poder regular esta corriente de 220VAC tal como se muestra en la Figura 41.

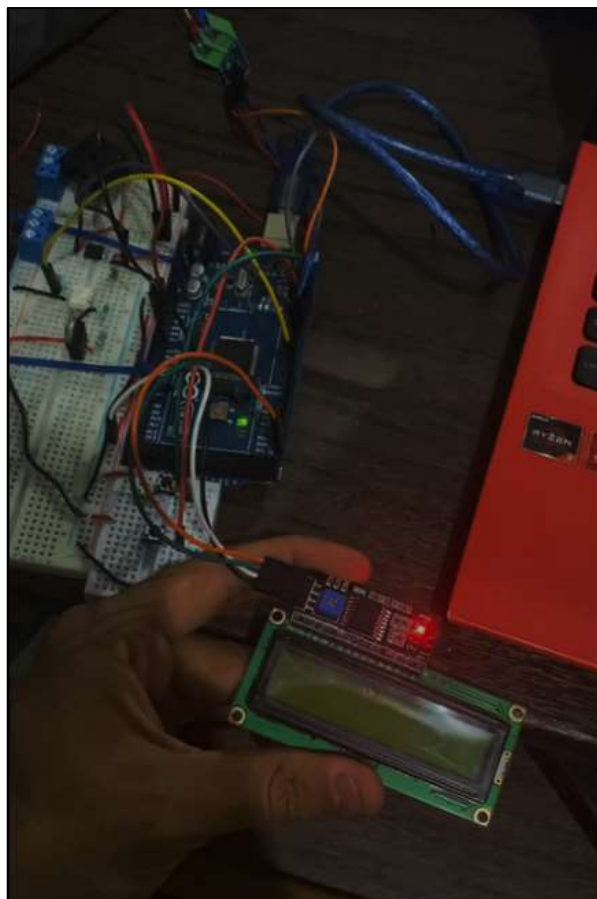


Figura 41. Componentes del controlador PID.

Fuente: Elaboración propia.

Capítulo 6. Resultados.

Una vez conectado cada componente en la placa de pruebas, que en este caso fue un protoboard, se conectó la placa de Arduino Mega2560 a la computadora para poder cargar el programa previamente realizado y así poder probar el funcionamiento en cuanto a la regulación de la señal y que se pudiera trabajar a la par con Arduino como se ve en la Figura 42.



Figura 42. Funcionamiento de comunicación de Arduino con el controlador PID.

Fuente: Elaboración propia.

Después de haber probado de que su funcionamiento fuera correcto y ver que no se tuvo ningún obstáculo como el quemarse algún componente por conectarlo a una corriente de 220VAC, se tomó la decisión de conectarse el Arduino a todo el circuito eléctrico con el programa ya cargado como se puede ver en la Figura 43.

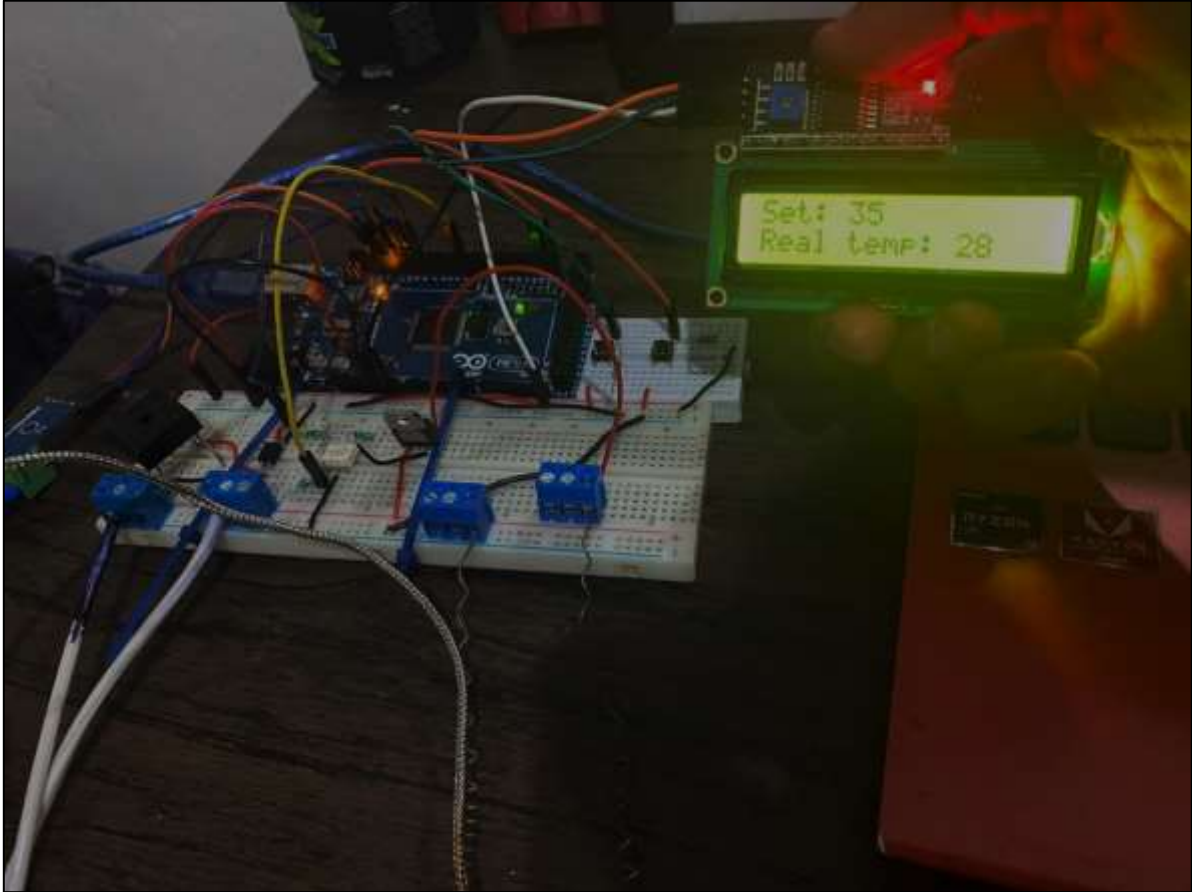


Figura 43. Controlador PID en funcionamiento con la placa de Arduino Mega2560.

Fuente: Elaboración propia.

4. Por último, se logró obtener un controlador PID con ayuda del microcontrolador Arduino Mega2560, quien sirvió como la acción de control, esto debido a que, el propio Arduino es quien mandaría la señal al transistor TRIAC, el cual es el encargado de dispararse de acuerdo a cuanto ángulo de la onda dejaría pasar para regular el voltaje de 220VAC, que con ayuda del puente de diodos que fue quien rectificaba la señal senoidal de la corriente AC, así como también con la ayuda del optoacoplador y el MOC3020 para poder trabajar con esa cantidad de voltaje, y poder bajarla a una frecuencia más baja de 5V para así poder hacer uso del Arduino sin correr el riesgo de dañarlo por suministrarle más voltaje del necesario y con ello poderlo utilizar como ya se mencionó, donde el Arduino es el encargado de ejecutar la acción de control gracias al programa previamente

Capítulo 6. Resultados.

creado. Para esta prueba era necesario montar el controlador PID en el prototipo final de un horno académico realizado por dos estudiantes de la misma carrera de Ingeniería en Sistemas Automotrices, quienes nos facilitaron el uso de uso para poder realizar esta prueba del controlador PID como se muestra en la Figura 44, y que a su vez también se puede observar cómo está conectado el controlador PID a las resistencias que fueron la carga en el circuito eléctrico en forma de estrella a 220VAC.

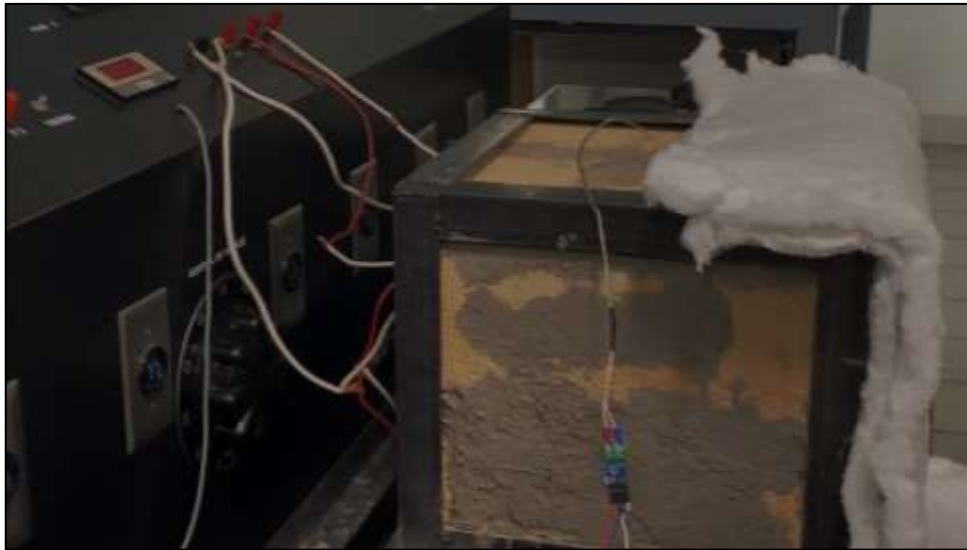


Figura 44. Implementación del controlador PID con un prototipo de horno a 220VAC.

Fuente: Elaboración propia.

Una vez que se hicieron las conexiones necesarias de las resistencias en forma de estrella a 220VAC en conjunto con el controlador PID, se verificó su correcta conexión tal como se muestra en la Figura 45.

Capítulo 6. Resultados.

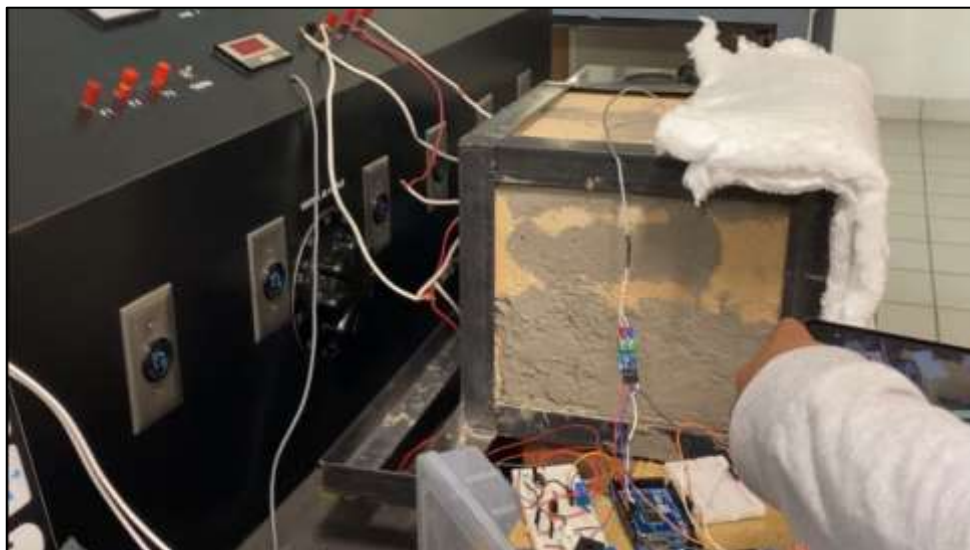


Figura 45. Conexión del controlador PID con el horno académico.

Fuente: Elaboración propia.

Después de verificar su correcta conexión, se compiló el programa de Arduino en la computadora y posteriormente se subió a la placa de Arduino Mega2560, con el SetPoint deseado para una simple prueba, el cual fue de 190° C, tal como se muestra en la Figura 46.



Figura 46. Ajuste del SetPoint en el programa de Arduino previo a su carga a la placa.

Fuente: Elaboración propia.

Capítulo 6. Resultados.

Luego se puede observar en la Figura 47, cómo en la pantalla LCD 16x2 junto al módulo de comunicación i2C, se muestra el SetPoint establecido en un inicio en el Arduino IDE era de 190 y a su vez, se muestra también la temperatura real que está leyendo el termopar tipo K, el cual está incrustado en la parte superior del horno.

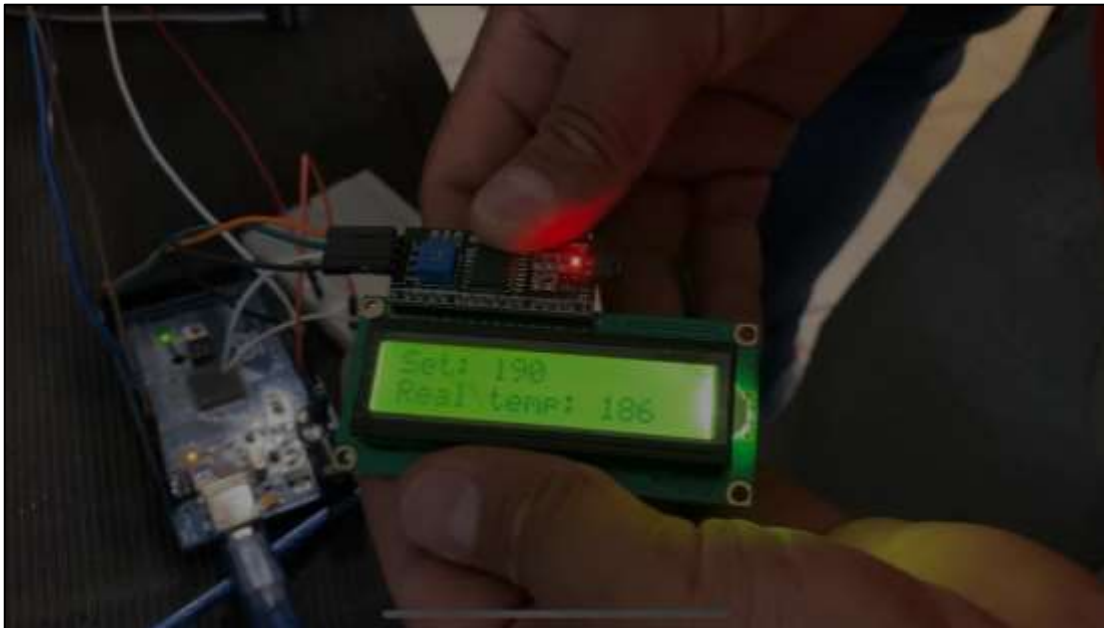


Figura 47. SetPoint y temperatura real dentro del prototipo del horno académico.

Fuente: Elaboración propia.

Como se puede observar en la Figura 48, la temperatura real que está siendo censada por el termopar tipo K dentro del prototipo del horno académico, está a nada de llegar a la temperatura del SetPoint establecido al inicio antes de ejecutar el programa en el Arduino IDE, el cual es de 190°C y la temperatura dentro del horno es de 189°C, por lo que en esta parte debería de comenzar a oscilar la señal por parte de la acción de control para tratar de ajustar esta temperatura real a la del SetPoint y mantenerla así hasta que el usuario decida apagar el controlador PID.

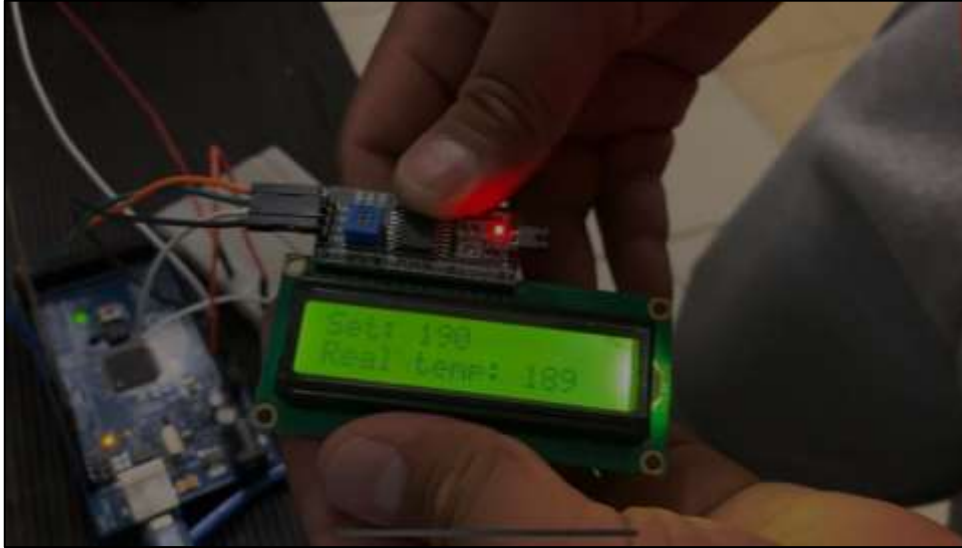


Figura 48. Controlador PID oscilando por debajo de la temperatura del SetPoint.

Fuente: Elaboración Propia

En este momento comenzó a oscilar la temperatura, logrando que subiera por mucho a 195°C, es decir, que la temperatura real se pasó por 5°C, pero llegó un momento donde la temperatura medida con el termopar empezó a bajar y llegó a estar a una diferencia de un 1°C, es decir, bajó a la temperatura de 191°C, por lo tanto el controlador PID estaba funcionando de manera correcta, tal como se muestra en la Figura 49.

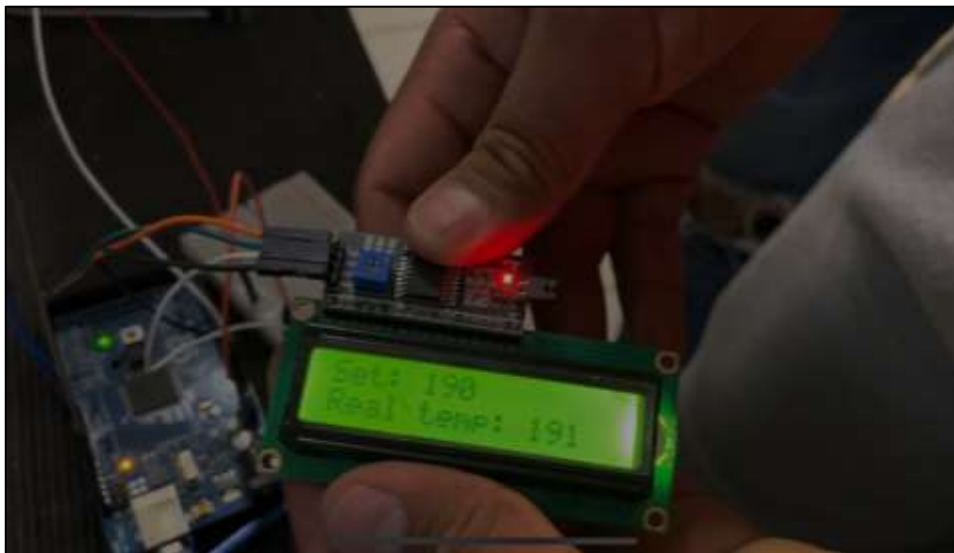


Figura 49. Controlador PID oscilando con una falla de 1°C por arriba del SetPoint.

Fuente: Elaboración propia.

Capítulo 6. Resultados.

Después de algunos minutos la temperatura se comenzó a estabilizar y mantener en el SetPoint que se estableció desde un inicio como se muestra en la Figura 50, es decir, que las oscilaciones ya no eran muy elevadas como estaban siendo de 5°C de diferencia como al inicio, sino que ya eran de $\pm 3^{\circ}\text{C}$, por lo que el controlador estaba funcionando con muy buenos resultados.



Figura 50. Controlador PID estabilizando la temperatura real a temperatura del SetPoint.

Fuente: Elaboración propia

Capítulo 7

Análisis de Resultados

Para lograr una mejor estructura en cuanto a los análisis de cada uno de los resultados obtenidos, se enlistarán de la misma manera con su respectivo orden que la sección anterior de los resultados, es decir, cada análisis tendrá el mismo número que el resultado que se vaya a estar explicando con respecto a la sección anterior.

1. Los resultados obtenidos en cuanto al primer código creado en Arduino, tal como se muestra en la , el cual fue la primera prueba realizada tal como se muestra en la Figura 34, fue un éxito por el motivo que nos permitió monitorear la temperatura del primer prototipo de un horno eléctrico utilizando el termopar tipo K incrustado en la parte posterior del mismo horno, por lo que estaba censando de manera correcta la temperatura dentro del horno y a su vez, lo estaba mostrando de manera fluida cada 500ms en la interface del Arduino IDE, precisamente en el monitor serial, tal como se muestra en la Figura 51.

```
21 void loop() {
22   // Instrucciones para poder imprimir la temperatura en el monitor serial tanto en
23   Serial.print("Grados °C = ");
24   Serial.print(ktc.readCelsius());
25   Serial.print("\t Grados °F = ");
26   Serial.print(ktc.readFahrenheit());
27
28   // Se espera medio segundo entre cada lectura de temperatura
29   delay(500);
30 }
```

Figura 51. Impresión del monitoreo de temperatura con el monitor serial.

Fuente: Elaboración propia

2. Esta prueba permitió avanzar a la siguiente fase, debido a que, se cumplió con el primer objetivo específico que se había trazado desde el inicio del proyecto, así que, se llegó a que el programa tenía la necesidad de una mejor interfaz de visualización con el usuario que operara el horno, por lo que, se le integró al circuito eléctrico una pantalla LCD 16x2, la cual tendría un mejor monitoreo de la temperatura sin la necesidad de tener que estar viendo el monitor serial del

Arduino IDE. Sin embargo, la pantalla que se utilizó no contaba con la comunicación i2C por lo que se tenía que soldar y cablearla con 14 cables Dupont macho-macho en un protoboard, lo que provocaba un circuito mucho más desordenado, así que se consiguió un módulo con comunicación i2C para una pantalla LCD 16x2 para poder cablearla con tan solo 4 cables Dupont hembra-macho y así mejor la estética, el orden y el ruido que se pudiera causar en el circuito eléctrico por tener tantos cables conectados para un solo componente, así que por eso se optó por ese módulo. Al terminar las conexiones en donde se realizó esa prueba de monitoreo de la temperatura y se imprimiera el valor en la pantalla LCD 16x2, tal como se muestra en la Figura 38, se cargó el nuevo programa realizado y no se mostraban los valores en la pantalla debido a que, existe una propiedad cuando se utilizan este tipo de pantallas, los cuales podrían ser 3 posibles direcciones a la hora de usar la librería de “LiquidCrystal_I2C.h”; (0x3f,16,2), (0x27,16,2) y (0x20,16,2). Sin embargo, la dirección que se había puesto en el programa no era la correcta por lo que al cambiarla a la dirección de (0x27,16,2) como se muestra en la Figura 52, fue la solución ya que, con esa dirección comenzó a mostrar los valores de la temperatura de una excelente manera, por lo que se concluyó de manera exitosa esta prueba cumpliendo así el segundo objetivo específico que se había trazado en un comienzo.

```
1  /*
2  Este programa es un ejemplo para medir la temperatura con un termopar tipo K
3  con ayuda de su módulo MAX6675 y mostrarla en una pantalla LCD 16x2
4  Autores del programa: RAMIRO GARCÍA SIXTOS Y FRANCISCO JAVIER MENDOZA DÍAZ
5  */
6  #include "max6675.h"
7  #include <LiquidCrystal_I2C.h> //Librería para poder utilizar la pantalla LCD con i2C
8  LiquidCrystal_I2C lcd(0x27,16,2); /* Aquí es importante verificar la dirección
9  que funcione en tu pantalla LCD, ya que, si no te sale con esta dirección puedes usar
10 (0x3f,16,2) || (0x27,16,2) ||(0x20,16,2)*/
11
12 //Se definen los pines que se van a usar para conectar el modulo MAX6675
13 int ktcS0 = 8;
14 int ktcCS = 9;
15 int ktcSCK = 10;
16
17 MAX6675 ktc(ktcSCK, ktcCS, ktcS0);
```

Figura 52. Dirección del LiquidCrystal para que funcionara la pantalla LCD.

Fuente: Elaboración propia.

3. En la tercera prueba realizada donde se elaboró el circuito eléctrico completo del controlador PID, se presentaron muchos obstáculos en su momento debido a que, se realizó una muy detallada investigación de cuales elementos nos podrían ayudar a hacer posible el funcionamiento en conjunto entre la corriente alterna de 220V y la placa de Arduino Mega2560, ya que como bien se mencionó en las secciones anteriores, el Arduino trabaja con 12V como máximo, así que si se le conectara la corriente de 220VAC de manera directa, lógicamente provocaríamos que se quemara la placa de Arduino Mega2560 y las siguientes pruebas con el prototipo final del horno académico no sería posible, así que mediante dicha investigación se determinó el conjunto de componentes a utilizar tomando en cuenta la función que pudiera realizar cada uno pero, el problema fue a la hora de querer conectar y cablear todo el controlador PID, ya que, cualquier cable que estuviera mal conectado al protoboard o no estuviera bien ajustado, podría ser de grave peligro porque la corriente de 220VAC sería capaz de dar un salto entre los demás componentes y no solo los podría quemar sino que, en el peor de los casos podría causarnos graves lesiones e incluso hasta mortales. Es por ello que se realizó la conexión del circuito de una manera meticulosa para poder llegar al circuito completo del controlador PID tal como se muestra en la Figura 43, y a su vez, se le cargó el programa creado en el interfaz de Arduino IDE para el funcionamiento de dicho controlador PID dejando ver así, el cumplimiento de otros dos objetivos para poder lograr el objetivo general, el cual se propuso que era el poder crear un controlador PID que fuera capaz de regular la temperatura en un horno académico.
4. La última de las pruebas fue la de implementar el controlador PID en un horno académico, para terminar de cumplir el objetivo general que se trazó al comienzo del proyecto, y para ello, obviamente se debía de contar con un horno para después hacerle su estudio al sistema o planta y a partir de ahí tratar de

Capítulo 7. Análisis de Resultados.

encontrar la función de transferencia para poder encontrar las constantes proporcional, integral y la derivativa, así como la razón de cambio en una curva de escalón unitario, el cual sería el método utilizado para conocer esas constantes en este sistema del horno académico, tal como se muestra en la Figura 44, el cual fue facilitado por dos compañeros de la misma carrera que realizaron sus residencias haciendo dicho horno y fue de gran ayuda, ya que, en pocas palabras la combinación de ambos trabajos abriría paso a un gran trabajo en conjunto para obtener lo que se conoce como horno tipo mufla y que, por si fuera poco así se lograría un impulso en la carrera debido a que, ya no solamente se contaría con una mufla en la división de sistemas automotrices, sino que, serían ya dos muflas para que los estudiantes de dicha carrera puedan seguir fundiendo con mayor frecuencia. Sin embargo en esta prueba no se elevó a una temperatura considerable por el hecho de que en el ITSUR no cuentan con una muy buena instalación eléctrica, ya que las resistencias por más que se conectaron en 127VAC, aun así demandaban 15 amperes cada una de las resistencias y se colocaron 3 resistencias en paralelo y las pastillas del ITSUR son de 20 amperes por lo que solamente aguantaba de una resistencia por contacto ya que, se subía de más la corriente demandando 45 amperes y esto provocaba lógicamente que se votara la pastilla, hasta que se conectó en la corriente de 220VAC fue cuando se pudo realizar esta pequeña prueba y para no arriesgar de más tanto el controlador como el prototipo final del horno académico, se decidió solamente realizarlo con una temperatura (SetPoint) de 190°C, en la cual se puede asegurar que si cumplió y se mantuvo en ahí hasta que el usuario decidiera apagar el controlador desconectando la extensión que venía de la corriente de 220VAC.

Al final los resultados obtenidos lograron cumplir los objetivos específicos planeados, y estos a su vez ayudaron al cumplimiento del objetivo general de lograr crear un controlador PID usando el entorno de programación del Arduino IDE, así como también utilizando circuitos electrónicos integrados, el cual permitió el monitoreo de

Capítulo 7. Análisis de Resultados.

temperatura y control de regulación de voltaje en las resistencias del horno. Por lo tanto, así se logró tener un controlador que sea capaz de monitorear la temperatura y esta a su vez se mantenga en una temperatura deseada (SetPoint) por el usuario que lo programe a la hora de querer fundir.

Capítulo 8

Conclusiones y trabajo a futuro

En este trabajo, se logró cumplir con todos los objetivos trazados en un principio, por lo cual nos llevaron al cumplimiento del objetivo general el cual fue “Analizar los fundamentos de un controlador proporcional integral derivativo (PID) y diseñarlo para lograr un control programable de temperatura para su integración a un horno académico.”. Esto se logró ya que, el controlador elaborado es capaz de regular la temperatura deseada, la cual se le conoce como el SetPoint y es el punto de referencia al cual debía llegar el controlador PID con ayuda del programa creado en Arduino como tal, por lo que el programa se compiló de manera correcta y a su vez, se subió de manera exitosa a la placa de Arduino Mega2560, lo que nos dice que el programa estaba bien en cuanto a su estructura del código, sin embargo, eso no es todo lo que necesita el controlador PID para poder funcionar sino que, también era necesario tener mucho cuidado en la conexión del circuito eléctrico, el cual sería la parte física del controlador como tal, ya que, como bien se mostró en los resultados del proyecto y en el propio análisis de resultados que se tuvo que conectar a la corriente alterna de 220V, así que con esto ya se comenzaría a poner más serio el tema, lógicamente por lo peligroso que pudo haber sido si estuviera conectado algo mal. Es por ello que se está satisfecho con el resultado a pesar de que no se elevó tanto la temperatura en el SetPoint, la cual se estableció de 190°C por la sencilla razón de que estaba contemplado para usarlo a 127V como máximo pero por los cambios de que las resistencias no alcanzaban un óptimo funcionamiento por las pastillas que se tienen en la instalación eléctrica de la escuela, ya que, solamente tenían de 20 amperes y en nuestro caso cada resistencia demandaba 15 amperes, por lo que se tuvo que buscar la alternativa de conectarlas en estrella y por supuesto a 220V para que así solo nos mandara por mucho 30 amperes entre las 3 resistencias, esto debido a que, como ya estaba mandando más voltaje, ya no sería necesario mandarles tanta corriente a las 3 resistencias por lo que así si funcionó y

Capítulo 8. Conclusiones y trabajo a futuro.

se logró el correcto funcionamiento del controlador PID con el enlace de comunicación con Arduino y junto al programa creado, permitiendo así, regular la temperatura de 190°C donde su oscilación máxima fue de $\pm 5^{\circ}\text{C}$, lo que significa un buen resultado.

Este controlador dejó buenas sensaciones en sus creadores y los dejó satisfechos después de que se tuvieron demasiados problemas en cuanto a su creación y a la hora de realizar las pruebas de funcionamiento con la corriente de 127V y demás problemas presentados durante todo el semestre, sobre todo por problemas del recurso económico para comprar los elementos que lo componían y así poder construir el controlador como tal. Sin embargo, con el controlador PID creado, se tienen muy buenas expectativas en cuanto a sus mejoras que se le pueden realizar como trabajo a futuro, pensando en un trabajo de titulación por tesis, en el cual se tendría planeado realizar un cambio en el código en el que ya no solamente permita estabilizar una temperatura, sino que también nos deje estabilizar más temperaturas en un solo ciclo, y con ello también lograr que el código tenga la capacidad de que puedas utilizar varios ciclos en un solo proceso de fundición e incluso se pueda controlar los tiempos de subida de temperatura, es decir, que exista la posibilidad de realizar un proceso de fundición en escalera para que vallas controlando el tiempo que tenga que durar el horno en alguna temperatura específica y terminando ese tiempo, el mismo programa sea capaz de que vuelva a subir la temperatura y así sucesivamente para lograr un mejor controlador PID capaz de controlar la temperatura de un horno académico basado en una programación de escalera como la tiene en su caso la propia mufla disponible en la carrera de sistemas automotrices.

De la misma manera también podría tener una mejora en cuanto a la manera de controlar o programar la temperatura, agregando un control como tal afuera del horno donde ya no se tenga que ajustar desde el Arduino sino que, simplemente tengas una pantalla con botones y ahí puedas manipular el interfaz del horno y a su vez, se podría implementar placa ESP8266 para lograr una conexión del horno a internet y

Capítulo 8. Conclusiones y trabajo a futuro.

se pueda controlar y monitorear tanto su temperatura como su proceso en general desde una página web.

Referencias bibliográficas

- [1] B. C. Kuo, Sistemas de control automático, Séptima ed., L. G. C. Plascencia, Ed., Estado de México: PRENTICE-HALL HISPANOAMERICANA. S.A., 1996.
- [2] K. Ogata, Ingeniería de control moderna, 5 ed., Madrid: PEARSON EDUCACIÓN, S.A., 2010, p. 904.
- [3] J. Medina, «Monografías,» 2018. [En línea]. Available: <https://www.monografias.com/trabajos109/sistemas-regulacion-y-control/sistemas-regulacion-y-control2>.
- [4] S. Castaño, «Control Automatico Educación,» [En línea]. Available: <https://controlautomaticoeducacion.com/control-realimentado/lazo-abierto-y-lazo-cerrado/>.
- [5] T. H. Karl J. Åström, Control PID Avanzado, Madrid: PEARSON EDUCACIÓN S. A., 2009.
- [6] C. Cinjordiz, «<https://www.infootec.net/arduino-ide/>,» 12 Febrero 2018. [En línea]. Available: <https://www.infootec.net/arduino-ide/>.
- [7] P. HET, «HETPRO,» 2015. [En línea]. Available: <https://hetpro-store.com/el817/>.
- [8] UNIT, «UNIT ELECTRONICS,» 2016. [En línea]. Available: <https://uelectronics.com/producto/kbpc1010-puente-de-diodos-10a-1000v/>.
- [9] A. A. COMPANY, «Newark,» 2019. [En línea]. Available: <https://mexico.newark.com/onsemi/moc3020-m/optocoupler-triac-4200vrms/dp/72K8895>.
- [10] TEII, «TEII.MX,» 4 Junio 2019. [En línea]. Available: <https://teii.mx/alambre-kanthal/>.
- [11] TEC, «TECNEU,» 2016. [En línea]. Available: <https://www.tecneu.com/products/termopar-tipo-k-50cm-sensor-de-temperatura>.
- [12] Geek, «Geek factory,» 2013. [En línea]. Available: <https://www.geekfactory.mx/tienda/modulos/max6675-modulo-interfaz-para-termopar-tipo-k/>.
- [13] 330Company, «330ohms,» 2023. [En línea]. Available: <https://www.330ohms.com/products/pantalla-lcd-16x2-con-interfaz-i2c-fondo-azul>.
- [14] TecNM, «Tecnológico Nacional de México,» 2023. [En línea]. Available: https://www.tecnm.mx/?vista=Convocatorias_Tecnm.
- [15] J. Lorenzo, «Youtube,» 2017. [En línea]. Available: <https://www.youtube.com/watch?v=ulBble7lcW4>.
- [16] ELECTRONOBS, «Youtube,» 8 Noviembre 2018. [En línea]. Available: <https://www.youtube.com/watch?v=NVwsOzVxZak&t=85s>.

Anexos

```
1  /*
2  Este programa es un ejemplo para medir la temperatura de un
3  termopar tipo K con ayuda de su módulo MAX6675 y mostrarlo en el monitor serial de Arduino
4  Autores del programa: RAMIRO GARCÍA SIXTOS Y FRANCISCO JAVIER MENDOZA DÍAZ
5  */
6
7  #include "max6675.h"
8  //Se definen los pines que se van a usar para conectar el modulo MAX6675
9  int ktcS0 = 8;
10 int ktcCS = 9;
11 int ktcCLK = 10;
12
13 MAX6675 ktc(ktcCLK, ktcCS, ktcS0);
14
15 void setup() {
16     // Se activa la comunicación serial
17     Serial.begin(9600);
18     delay(500);
19 }
20
21 void loop() {
22     // Instrucciones para poder imprimir la temperatura en el monitor serial tanto en
23     Serial.print("Grados °C = ");
24     Serial.print(ktc.readCelsius());
25     Serial.print("\t Grados °F = ");
26     Serial.print(ktc.readFahrenheit());
27
28     // Se espera medio segundo entre cada lectura de temperatura
29     delay(500);
30 }
```

Figura 53. Primer programa para medir temperatura y mostrarlo en el monitor serial.

Fuente: Elaboración propia

```
1  /* |
2  Este programa es un ejemplo para medir la temperatura con un termopar tipo K
3  con ayuda de su módulo MAX6675 y mostrarla en una pantalla LCD 16x2
4  Autores del programa: RAMIRO GARCÍA SIXTOS Y FRANCISCO JAVIER MENDOZA DÍAZ
5  */
6  #include "max6675.h"
7  #include <LiquidCrystal_I2C.h> //Librería para poder utilizar la pantalla LCD con i2C
8  LiquidCrystal_I2C lcd(0x27,16,2); /* Aquí es importante verificar la dirección
9  que funcione en tu pantalla LCD, ya que, si no te sale con esta dirección puedes usar
10 (0x3f,16,2) || (0x27,16,2) ||(0x20,16,2)*/
11
12 //Se definen los pines que se van a usar para conectar el modulo MAX6675
13 int ktcSO = 8;
14 int ktcCS = 9;
15 int ktcSCK = 10;
16
17 MAX6675 ktc(ktcSCK, ktcCS, ktcSO);
18
19 void setup() {
20     // Se activa la comunicación serial
21     Serial.begin(9600);
22     delay(500);
23 }
```

Figura 54. Segundo programa para monitorear temperatura desde un LCD 16x2 (1/3).

Fuente: Elaboración propia.

```
24     // Iniciamos la pantalla LCD 16x2
25     // Se imprime el mensaje que llevará al inicio la pantalla durante 3 segundos
26     lcd.init();
27     lcd.backlight();
28     lcd.clear();
29     lcd.setCursor(0,0);
30     lcd.print(" TERMOPAR TIPO K ");
31     lcd.setCursor (0,1);
32     lcd.print("MAX6675 ARDUINO");
33
34     // Esperamos 3 segundos para que comience a mostrar las temperatura
35     delay(3000);
36 }
37 }
```

Figura 55. Segundo programa para monitorear temperatura desde un LCD 16x2 (2/3).

Fuente: Elaboración propia.

```
38 void loop() {
39     // El termopar lee el valor en centigrados y fahrenheit y lo guarda en una variable
40     double tC = ktc.readCelsius();
41     double tF = ktc.readFahrenheit();
42     // Primero se limpia la pantalla LCD para después imprimir la temperatura
43     lcd.clear();
44     lcd.setCursor(0,0);
45     lcd.print("TEMPERATURA");
46     lcd.setCursor( 0, 1 );
47     lcd.print("=");
48     lcd.setCursor( 1, 1);
49     lcd.print(tC);
50     lcd.setCursor( 6, 1 );
51     lcd.print("C");
52     lcd.setCursor( 8, 1 );
53     lcd.print("=");
54     lcd.setCursor( 9, 1);
55     lcd.print(tF);
56     lcd.setCursor( 15, 1 );
57     lcd.print("F");
58
59     // Imprimimos la temperatura al mismo tiempo en el monitor serial
60     Serial.print("C = ");
61     Serial.print(tC);
62     Serial.print(" F = ");
63     Serial.println(tF);
64     // Esperamos 1/3 de segundo en cada lectura de temperatura
65     delay(333);
66 }
```

Figura 56. Segundo programa para monitorear temperatura desde un LCD 16x2 (3/3).

Fuente: Elaboración propia.

```

1  /*
2  Este programa es un ejemplo para regular un voltaje de entrada 127VAC y regularlo
3  para su regulación de la temperatura utilizando resistencias.
4  Autores del programa: RAMIRO GARCÍA SIXTOS Y FRANCISCO JAVIER MENDOZA DÍAZ
5  */
6
7  #include "max6675.h"
8  #include <Wire.h>
9  #include <LiquidCrystal_I2C.h>
10 LiquidCrystal_I2C lcd(0x27,16,2); /* Aquí es importante verificar la dirección
11 que funcione en tu pantalla LCD, ya que, si no te sale con esta dirección puedes usar
12 (0x3f,16,2) || (0x27,16,2) ||(0x20,16,2)*/
13
14 //aquí se declaran las entradas y salidas
15 int cruce_cero = 8; //el pin de la detección del cruce por cero será el 8
16 int angulo_disparo = 3; //el pin del angulo de disparo será el 3
17 int ktcSO = 9; //el pin utilizado por la variable ktcSO del modulo MAX6675 con el termopar será el 9
18 int ktcCS = 10; //el pin utilizado por la variable ktcCS del modulo MAX6675 con el termopar será el 10
19 int ktcCLK = 13; //el pin utilizado por la variable ktcCLK del modulo MAX6675 con el termopar será el 13
20 int incremento = 11; //el pin para el incremento del set point con el push boton será el 11
21 int decremento = 12; //el pin para el decremento del set point con el push boton será el 12
22
23 MAX6675 thermocouple(ktcSO, ktcCS, ktcCLK); //comando de inicialización de termopar con su modulo MAX6675
24
25 //en esta sección se declaran las Variables
26 int ultimo_estado = 0;
27 int delay_angulo_disparo = 7400;
28 bool cruce_cero_detectado = false; //variable booleana para decir si fue verdadero o falso que hubo cruce por el cero
29 int delay_maximo_angulo_disparo = 7400;
30 /* son 7400ms de retraso máximo después de la detección de cero debido

```

Figura 57. Programa final del controlador PID (1/6).

Fuente: Elaboración propia.

```

31 al medio periodo de una onda de frecuencia en Mexico de 60Hz*/
32 unsigned long previousMillis = 0;
33 unsigned long currentMillis = 0;
34 int setpoint = 100;
35 bool pushboton1_presionado = false;
36 bool pushboton2_presionado = false;
37 int delay_lectura_temperatura = 500; //declaramos la variable del tiempo de lectura por cada temperatura en el termopar
38 int temperatura_real = 0; //declaramos la variable donde se guardará el valor de temperatura medida por el termopar al momento
39
40
41 //declaramos las variables del controlador PID como tal
42 float error_PID = 0;
43 float error_previo = 0;
44 float tiempo_transcurrido, tiempo_actual, tiempo_previo;
45 int valor_PID = 0;
46 //ahora declaramos las constantes PID de acuerdo a la planta donde se este trabajando
47 int kp = 12.57;
48 int ki = 9.36;
49 int kd = 2.94;
50 int PID_p = 0;
51 int PID_i = 0;
52 int PID_d = 0;
53
54 void setup() {
55 //Se definen los pines a utilizar, así como su uso
56 pinMode (cruce_cero,INPUT);
57 pinMode (angulo_disparo,OUTPUT);
58 pinMode (incremento,INPUT);
59 pinMode (decremento,INPUT);
60

```

Figura 58. Programa final del controlador PID (2/6).

Fuente: Elaboración propia.

```

81 //Las siguientes líneas del código anterior son particularmente importantes. La primera línea habilita la interrupción del
82 banco 0 de cambio de pín (bit PCIE0 o banco 0 de habilitación de interrupción de cambio de pín) en el registro de control de interrupción
83 de cambio de pín (PCICR). Posteriormente después de las siguientes tres líneas específicas entonces qué pín (x) en el banco 0 se utilizará
84 (Pin Change Interrupt pin 0,3 y 4 o PCINT0,PCINT3 y PCINT4) en el registro Pin Change Mask para el banco 0 (PCMSK0).
85 */
86 PCICR |= (1 << PCIE0); //se habilita evento PCMSK0
87 PCMSK0 |= (1 << PCINT0); //se establece el pín 00 (entrada de cruce por cero) activa una interrupción en el cambio de estado.
88 PCMSK0 |= (1 << PCINT3); //se establece el pín D11 (botón de asiento) activa una interrupción en el cambio de estado.
89 PCMSK0 |= (1 << PCINT4); //se establece el pín D12 (botón de eliminación) activa una interrupción en el cambio de estado.
90
91 lcd.init(); //se inicia la comunicación ICD
92 lcd.backlight(); //se enciende la luz de fondo para LCD
93 }
94
95 void loop() {
96   currentMillis = millis(); //en esta variable anteriormente declarada se guarda el valor de tiempo en milisegundos antes de ingresar al void loop.
97
98   /* se crea esta condición para leer la temperatura y cambiar los valores guardados en la variable de "delay lectura temperatura"
99   Cambiando ese valor por encima del valor que desee. Sin embargo la lectura del módulo DS18B20 es lenta por lo que afectaría al
100   control PID al intentar leer la temperatura cada 200 ms pero, es por ello que se cambia a 500ms.
101   */
102
103   if((currentMillis - previousMillis >= delay_lectura_temperatura){
104     previousMillis += delay_lectura_temperatura; // se aumenta el tiempo anterior para el siguiente ciclo
105     temperatura_real = ktr.readCelsius(); //se obtiene la temperatura real en grados Celsius
106
107     error_PID = setpoint - temperatura_real; //se calcula el error que existe entre el setpoint con la temperatura real medida por el sensor
108
109     if(error_PID > 30) //constante integral solo afectará a los errores por debajo de 30°C
110
111
112
113
114
115
116
117
118
119
120

```

Figura 59. Programa final del controlador PID (3/6).

Fuente: Elaboración propia.

```

91   PID_i = 0;
92 }
93
94 PID_p = kp * error_PID; //se calcula el valor de la constante P
95 PID_i = PID_i + (ki * error_PID); //se calcula el valor de la constante I
96 tiempo_previo = tiempo_actual; // el tiempo anterior se guarda antes de la lectura del tiempo real
97 tiempo_actual = millis(); // lectura en tiempo real
98 tiempo_transcurrido = (tiempo actual - tiempo_previo) / 1000;
99 PID_d = kd * ((error_PID - error_previo)/tiempo_transcurrido); //se calcula el valor D
100 valor_PID = PID_p + PID_i + PID_d; //se calcula el valor PID total
101
102 // Definimos el rango de retardo de disparo entre 0 y 7400.
103 if(valor_PID < 0)
104 {   valor_PID = 0;   }
105 if(valor_PID > 7400)
106 {   valor_PID = 7400;   }
107 //se imprimen los valores en la pantalla LCD
108 lcd.clear();
109 lcd.setCursor(0,0);
110 lcd.print("Set: ");
111 lcd.setCursor(5,0);
112 lcd.print(setpoint);
113 lcd.setCursor(0,1);
114 lcd.print("Real temp: ");
115 lcd.setCursor(11,1);
116 lcd.print(temperatura_real);
117 error_previo = error_PID; //Guarda el error anterior.
118 }
119
120 //Si se detectó la interrupción del cruce por cero, creamos el pulso de disparo a 100ms

```

Figura 60. Programa final del controlador PID (4/6).

Fuente: Elaboración propia.

```
121 if (cruce_cero_detectado)
122 {
123     delayMicroseconds(delay_maximo_angulo_disparo - valor_PID); //Este retardo controla la potencia
124     digitalWrite(angulo_disparo,HIGH);
125     delayMicroseconds(100);
126     digitalWrite(angulo_disparo,LOW);
127     cruce_cero_detectado = false;
128 }
129 }
130
131 /*Una vez habilitado, ahora necesitamos escribir el propio ISR. Como se mencionó,
132 esta rutina ya tiene un nombre (PCINT0_vect) y todo lo que necesita hacer es incrementar la variable de conteo global.
133 */
134 ISR(PCINT0_vect){
135     //entrada desde el optoacoplador
136     if(PINB & 00000001){ //Se crea un AND con el registro de estado, Verificamos si el pin D0 esta ALTO
137         if(ultimo_estado == 0){ //Si el último estado fue 0, entonces tenemos un cambio de estado
138             cruce_cero_detectado = true; //se realiza detectado un cambio de estado
139         }
140     } else if(ultimo_estado == 1){ //Si el pin 8 es BAJO y el último estado fue ALTO, entonces tenemos un cambio de estado
```

Figura 61. Programa final del controlador PID (5/6).

Fuente: Elaboración propia.

```
141     cruce_cero_detectado = true; //se detecta un cambio de estado
142     ultimo_estado = 0; // se almacena el estado actual en el último estado para el siguiente ciclo
143 }
144 if(PINB & 000001000){ //se crea un AND con el registro de estado, Verificamos si el pin D11 esta ALTO
145     if (!pushboton1_presionado)
146     {
147         setpoint = setpoint + 5; //se incrementa la temperatura en 5. Cambie esto con su valor si lo desea.
148         delay(20);
149         pushboton1_presionado = true;
150     }
151 }
152 else if (pushboton1_presionado)
153 {
154     pushboton1_presionado = false;
155 }
156 if(PINB & 000010000){ //se crea un AND con el registro de estado, Verificamos si el pin D13 esta ALTO
157     if (pushboton2_presionado)
158     {
159         setpoint = setpoint - 5; //se disminuye la temperatura en 5. Cambie esto con su valor si lo desea.
160         delay(20);
161         pushboton2_presionado = true;
162     }
163 }
164 else if (pushboton2_presionado)
165 {
166     pushboton2_presionado = false;
167 }
168 }
169 }
```

Figura 62. Programa final del controlador PID (6/6).

Fuente: Elaboración propia.