

UNIVERSIDAD NACIONAL DEL ALTIPLANO
FACULTAD DE INGENIERIA MECÁNICA ELÉCTRICA,
ELECTRÓNICA Y SISTEMAS
ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA



**“DISEÑO Y SIMULACIÓN DE UNA ESTUFA DE CULTIVO -
MULTIUSO CON CONEXIÓN WIFI PARA MONITOREO
INALÁMBRICO”**

TESIS

PRESENTADO POR:

DARIO ALEJANDRO YEPEZ ARANIBAR

PARA OPTAR EL TÍTULO PROFESIONAL DE:

INGENIERO ELECTRÓNICO

PUNO – PERÚ

2017

**UNIVERSIDAD NACIONAL DEL ALTIPLANO
FACULTAD DE INGENIERÍA MECÁNICA ELÉCTRICA,
ELECTRÓNICA Y SISTEMAS
ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA**

**DISEÑO Y SIMULACION DE UNA ESTUFA DE CULTIVO – MULTIUSO CON
CONEXIÓN WIFI PARA MONITOREO INALAMBRICO**

**TESIS PRESENTADA POR:
DARIO ALEJANDRO YEPEZ ARANIBAR**

Fecha de sustentación: 14 – 12 - 2017

**PARA OPTAR EL TÍTULO DE:
INGENIERO ELECTRÓNICO**

APROBADO POR EL JURADO FIRMANTE CONFORMADO POR:



PRESIDENTE

:.....

Dr. MIDWAR ELÍAS VALENCIA VILCA

PRIMER MIEMBRO

:.....

M.Sc. EDWIN WILBER CHAMBI MAMANI

SEGUNDO MIEMBRO:

.....

Ing. JESUS VIDAL LOPEZ FLORES

DIRECTOR / ASESOR:

.....

Dr. MARCO ANTONIO QUISPE BARRA

PUNO – PERU

2017

Área : Automatización e Instrumentación

Tema : Aplicación de Procesamiento de Señales, Modelado y Simulación de
Sistemas de Control y Automatización.

DEDICATORIA

A mí querida abuela
Sara Ramírez por su
gran ayuda para lograr
hacerme profesional.

A mis queridos padres
José y Edith por su
gran esfuerzo para que
logre ser profesional.

A mí querida hermana
Sara Gabriela.

AGRADECIMIENTOS

Quiero expresar un cálido agradecimiento a todas las personas de la Escuela profesional de Ingeniería Electrónica que me ayudaron con sus sugerencias, información, apoyo, confianza y sinceridad. En especial a Marco Antonio Quispe Barra (Director y asesor de tesis).

Mis sinceros agradecimientos al M.Sc. Edwin Wilber Chambi Mamani por sus consejos y enseñanzas en este trabajo y al Mg. Midwar Elías Valencia Vilca por la pronta corrección y aprobación de este proyecto de tesis.

INDICE GENERAL

RESUMEN	13
ABSTRACT.....	14
CAPITULO I.....	15
INTRODUCCIÓN	15
1.1 El problema de la investigación	15
1.2 Antecedentes de la investigación	17
1.3 Importancia y Utilidad del Estudio	18
1.4 Objetivos de la investigación.....	18
1.4.1 Objetivo general.....	18
1.4.2 Objetivos específicos	18
CAPITULO II	20
REVISIÓN DE LITERATURA	20
2.1 Sistemas de Control Digital.....	20
2.1.1 Diseño de Sistemas de Control Digital	20
2.1.2 Concepto de Estado y Variables de Estado.....	22
2.1.3 Linealización de Procesos.....	23
2.1.4 El Espacio de Estado Discreto	24
2.1.5 Controlabilidad	25
2.1.6 Observabilidad	25
2.1.7 Modelado Matemático	26
2.1.8 Control Óptimo	29
2.1.9 Control Óptimo Cuadrático Estacionario	30
2.1.10 El Regulador Óptimo Proporcional	31

2.1.11 Observador óptimo cuadrático.....	31
2.1.12 El Regulador Óptimo Proporcional Integral.....	33
2.1.13 Parámetros del proceso.....	37
2.2 Comunicación WIFI.....	38
2.2.1 Bandas ISM/UNII.....	38
2.2.2 Tasas de Transmisión.....	39
2.2.3 CSMA vs. TDMA.....	39
2.2.4 Topologías de redes inalámbricas.....	40
2.2.5 Modos de funcionamiento en WiFi.....	41
2.3 El Lenguaje HTML (Hyper Text Markup Language).....	43
2.3.1 Cabecera.....	45
2.3.2 Cuerpo.....	45
2.4 Circuito Optoacoplador.....	46
2.5 HIPOTESIS.....	47
2.5.1 Hipótesis General:.....	47
2.5.2 Hipótesis Específicas:.....	47
CAPITULO III.....	48
MATERIALES Y MÉTODOS.....	48
3.1 Tipo de investigación del problema.....	48
3.2 Población y muestra de la investigación.....	48
3.3 Técnicas de tratamiento de Datos.....	48
3.4 Arduino UNO.....	49
3.4.1 HARDWARE.....	49
3.4.2 SOFTWARE.....	51
3.5 Módulo ESP8266.....	53

3.5.1 Especificaciones de este modulo	53
3.5.2 Características eléctricas	54
3.5.3 Firmware para el ESP8266	54
3.5.4 Módulos ESP-XX	54
3.6 PIC16F84	55
3.6.1 Puertos de entrada/salida	55
3.6.2 Oscilador	56
3.6.3 Reset.....	57
3.7 Driver L298N.....	58
3.7.1 Funcionamiento	58
3.8 Sensor de temperatura por resistencia PT100	59
3.9 MOC3041.....	62
3.10 Optoacoplador 4N25	63
CAPITULO IV	64
RESULTADOS Y DISCUSIÓN	64
4.1 Breve descripción del programa de la página web	77
4.2 Contrastación de Hipótesis	79
CONCLUSIONES	80
RECOMENDACIONES	81
REFERENCIAS.....	82
ANEXOS.....	83

ÍNDICE DE FIGURAS

Figura 1: Estructura de un sistema de control digital	21
Figura 2: Horno de laboratorio o estufa.....	28
Figura 3: Sistema de control óptimo a lazo cerrado	31
Figura 4: Esquema del regulador óptimo proporcional	31
Figura 5: Diagrama de bloques del observador de estados.....	32
Figura 6: El regulador proporcional integral	33
Figura 7: Respuesta al escalón unitario en Matlab resultante de 6000 muestras.....	38
Figura 8: Canales en 802.11 con diferentes frecuencias centrales	40
Figura 9: Diagrama que ilustra los campos de acción en un red Ad-hoc	41
Figura 10: Diagrama que muestra los campos de una res con una estación especializada AP (Punto de acceso).....	41
Figura 11: Grafico que muestra los radios de acción en los distintos modos WiFi.....	43
Figura 12: Diagrama que muestra el circuito Optoacoplador para el manejo de la resistencia calefactora de la estufa de cultivo	47
Figura 13: Figura que nos señala las partes de un Arduino UNO	49
Figura 14: En esta figura se señala las partes de módulo ESP-01	55
Figura 15: En esta figura se indican los nombres y funciones de los pines de PIC16F84A	56
Figura 16: Figura donde se muestran los puertos de este proyecto	58
Figura 17: Figura donde se ilustra las conexiones necesarias para cambiar los voltajes en las salidas	59
Figura 18: En esta grafico se muestra la razón de cambio de la resistencia eléctrica versus la temperatura de un PT100.....	59

Figura 19: Figura donde se muestra la conexión de 2 hilos donde se muestran sus resistencias RC1 y RC2 como la del sensor R(t)	60
Figura 20: Modo de conexión de 3 hilos con sus respectivas conexiones Rc	61
Figura 21: Modo de conexión de 4 hilos con su respectivo voltímetro y amperímetro para calcular la resistencia R(t).....	61
Figura 22: En esta imagen se muestra que una ilustración de MOC3041 de Fairchild ..	62
Figura 23: Figura que ilustra las funciones de los pines de este Optoacoplador	63
Figura 24: Figura donde se muestra la respuesta al escalón unitario en un simulador con 1000 muestras tomadas.	66
Figura 25: Figura donde se muestra las distintas pendientes de cambio de temperatura con razones de 8, 2 y -4 grados por minuto.	67
Figura 26: Figura donde se muestra la pendiente de cambio de temperatura de 8°C/min.	68
Figura 27: Figura donde se muestra la rampa escalonada de la razón de cambio de 2°C/min.....	69
Figura 28: Figura que muestra una estufa de cultivo INCUDIGIT	70
Figura 29: Figura donde se muestra la segunda celda de la página web elaborada.....	70
Figura 30: Figura donde se muestra la tercera celda de la página web elaborada.....	71
Figura 31: Aquí se muestra el comando utilizado para salir de la configuración del ESP.	72
Figura 32: Aquí se muestra la página web elaborada que se mostrara al ingresar con su respectiva IP.....	73
Figura 33: Aquí se muestra el comando usado para el primer cambio de temperatura..	76
Figura 34: Se muestra el comando para cambiar la hora de la estufa multiuso.....	76
Figura 35: Se muestra el comando utilizado para pasar al modo incubadora.....	76

ÍNDICE DE TABLAS

Tabla 1: Tabla de la disminución de temperatura al apagar el Horno de laboratorio.....	28
Tabla 2: Temperaturas del horno en estado estable	28
Tabla 3: Aquí se muestra el programa completo de la simulación del controlador.....	34
Tabla 4: En esta tabla se muestra como escribir algunos caracteres.....	44
Tabla 5: A continuación se muestra el programa HTML de la página web	45
Tabla 6: Aquí se muestran los diferentes valores del sobrepaso máximo respecto a el cambio de proporción entre $Q(1,1)$ y $Q(2,2)$	65
Tabla 7: Aquí se muestran los valores del sobrepaso máximo variando $Q(1,1)$ y R	66
Tabla 8: A continuación se muestra la parte del programa encargado de distinguir estos comandos “CNG”.	73
Tabla 9: A continuación se muestra como el Arduino envía la página web mostrada anteriormente.	77

INCICE DE ANEXOS

ANEXO A 1: En esta figura se muestra un Arduino con el conexionado y en funcionamiento de la configuración de parámetros.....	84
ANEXO A 2: En esta figura se muestra el conexionado y funcionamiento del módulo ESP8266, cuando se encuentra activo se le enciende un LED Rojo como se muestra en la figura.....	84
ANEXO A 3: En esta figura se muestra los principales componentes de este proyecto, sin incluir la estufa y su resistencia calefactora, el motor visto sirve para mover los huevos en la cámara cuando estemos en el modo incubadora.....	85
ANEXO B 1: Aquí se muestra el módulo ESP8266 utilizado para la elaboración de nuestro servidor web para monitoreo inalámbrico.....	86
ANEXO B 2: En esta figura se puede observar una prueba de funcionamiento de la página web elaborada.....	86
ANEXO C 1: En esta fotografía se observa la estufa que adquirí para realizar posibles pruebas, y calibraciones del funcionamiento del controlador elaborado.....	87
ANEXO C 2: En esta imagen se puede observar que el motor DC está a altura de las parrillas de esta estufa, el motivo es porque este motor sirve para realizar el movimiento oscilatorio de las parrillas cuando el equipo está en modo incubadora.....	87

ÍNDICE DE ACRÓNIMOS

BSS	(Basic Service o Infrastructure)
CSMA	(Carrier Sense Multiple Access)
ESS	(Extended Service Set)
HTML	(Hyper Text Markup Language)
IBSS	(Independent Basic Service Set)
IDE	(Entorno de Desarrollo Integrado)
ISM	(Industrial, Scientific and Medical)
LCD	(Liquid Crystal Display)
MCLR	(Mater Clear)
PID	(Proporcional Integrativo Diferencial)
PWM	(Modulation for Wide of Pulse)
TDMA	(Time Division Multiple Access)
UNII	(Unlicensed National Information Infrastructure)
WECA	(Wireless Ethernet Compatibility Alliance)
WIFI	(Wireless Fidelity)
WLAN	(Wireless Local Area Network)

RESUMEN

Las estufas de cultivo en general están diseñadas para un uso en particular, como por ejemplo hay estufas que se especializaron en el cultivo de bacterias, las que se llamaron estufas de cultivo bacteriológicas, aparte de ello existen otras cámaras de control de calor como las incubadoras. Por otro lado también se observa que algunas estufas biomédicas cuentan con una configuración de características algo compleja, que a veces no permite una comprensible visualización ni distinción de los datos por lo que requieren de un manual algo extenso, que por motivos de tiempo no son bien entendidos por el usuario y finalmente la mayoría de estufas de cultivo convencionales no cuentan con un modo de configuración del cambio de razón de temperatura. Por lo descrito anteriormente, se desarrolló en este borrador de tesis el diseño de un controlador de temperatura, procurando obtener una configuración de los parámetros de trabajo más sencilla mediante una visualización de estos por una pequeña página web, también se desarrolló la simulación de las señales de temperatura. Utilice Método Sistémico, el cual usa el modelado de un objeto o proceso determinando sus componentes. Sin más que decir en este informe se mostraran algunas gráficas e imágenes de los resultados obtenidos.

Palabras Clave: eingenvvalor, proceso lineal controlable, proceso lineal observable, jacobiano, escalón unitario, etiqueta HTML.

ABSTRACT

The cultivation stoves in general are designed in particular for a use, as for example there are stoves that specialized in the cultivation of bacterial, those that called themselves bacteriological cultivation stoves, apart from they exist it other cameras of control of heat like the incubators. On the other hand also it is observed that some biomedical stoves have a configuration of characteristic something complex, that in some cases it doesn't allow a comprehensible visualization neither distinction of the data for what they require of a manual something extensive, that for reasons of time they are not very understood by the user and finally most of conventional cultivation stoves don't have a way of configuration of the change of reason of temperature. For that described previously, it was developed in this thesis draft the design of a controller of temperature, trying to obtain a configuration of the simplest work parameters by means of a visualization of these for a small page web, it was also developed the simulation of the signs of temperature. Use Systemic Method, which uses the modeling of an object or process determining their components. Without more than to say in this report some graphs and images of the obtained results was shown.

Key Words: ingenvalor, controllable lineal process, observable lineal process, jacobiano, unitary step, HTML labels.

CAPITULO I

INTRODUCCIÓN

1.1 El problema de la investigación

En la actualidad encontramos diferentes cámaras de calor eléctricas, como los hornos eléctricos, los esterilizadores, las estufas de cultivo, las incubadoras, etc., que tienen la principal similitud de controlar la temperatura. Por otro lado las estufas de cultivo convencionales que encontramos en el mercado por lo general están diseñadas para un uso en particular, como por ejemplo hay estufas que se especializaron en el cultivo de bacterias llamándose estufas bacteriológicas, mientras tanto por otro lado encontramos otras cámaras de calor como las incubadoras de huevos y las incubadoras neonatales, cada una de ellas tiene características que las distinguen de las otras, por ejemplo las estufas de cultivo de bacterias pueden cambiar por lo general dentro de un rango de temperatura ambiente a °C 80, por otro lado las incubadoras de huevos trabajan dentro del rango de °C 37 a 39 y tienen un movimiento periódico de la bandeja de puesta de huevos; todo ello me dio la idea de por qué no se podría elaborar una cámara de control de calor que pueda hacer ambas cosas ya que son diferencias poco significativas; por ello uno de los aspectos desarrollados en este proyecto fue incluir en una sola cámara estas dos diferencias con la ayuda de un motor de corriente continua; lo que permitiría a

los diferentes usuarios (biólogos, médicos, etc.) adquirir un solo equipo que nos brinde ambos beneficios.

También observe que hay diferentes modelos de estufas que tienen diferentes tipos de configuración, las más simples utilizan una lógica muy sencilla utilizando un reloj de configuración manual y un termostato, otras utilizan displays 7 segmentos que también son de sencilla configuración y algunas utilizan un LCD (Liquid Crystal Display) que nos da la posibilidad de configurar más parámetros de operación, pero su costo es más elevado que las anteriores. Dentro del grupo de estufas que usan displays, hay algunas que tienen displays que son de difícil comprensión, que requieren de un manual algo extenso, que por motivos de tiempo no son bien entendidos por los usuarios; por este motivo y por el costo elevado de las estufas que utilizan LCD es que en este proyecto se elaboró una página web que nos brinda una visualización completa de todos los parámetros de funcionamiento y una configuración mediante el uso de unos sencillos comandos.

Por otra parte también la gran mayoría de estufas de cultivo convencionales no cuentan con un modo de configuración del cambio de temperatura-tiempo (razón de cambio de temperatura versus tiempo), por lo que ello las limita para el cultivo de algunos seres vivos que tienen estándares de temperatura rígidos o sino también las limita para el uso en procesos de sumo cuidado, por estos motivos es que es que tuve la iniciativa de elaboración de una estufa que pueda controlar la razón de cambio de temperatura, el cual es uno de los temas que se desarrolló en este proyecto con el uso de un controlador óptimo y una secuencia de incremento de temperatura.

A parte de lo anterior es preferible tener un monitoreo constante del funcionamiento de nuestro equipo, que en algunos casos es interrumpido por desperfectos en la línea de

alimentación u otras fallas, por lo que tuve la idea realizar un monitoreo inalámbrico que se puede dar mientras la persona puede estar haciendo otras actividades.

1.2 Antecedentes de la investigación

- Artículo de: Control de Temperatura Usando Lógica Difusa. Por: C. Palacio-Gómez y J. García Sucerquia (Escuela de Física, Universidad Nacional de Colombia Sede Medellín. 2010)

En este trabajo se presentó el diseño e implementación de un control de temperatura con lógica difusa como elemento de toma de decisiones, la temperatura fue fijada a un valor de entre 30 hasta 100 grados centígrados, el controlador utilizado fue un controlador PIC 16F877A el cual fue conectado a un PC para su monitoreo y lograr establecer en tiempo real la temperatura del sistema controlado. En sus conclusiones aclaro que no se desarrolló la parte del dimensionamiento del ambiente a controlar, ya que este pertenece a la parte de potencia del mismo.

- Artículo de: Diseño de simulación de un controlador difuso de temperatura empleando el congresor basado en relaciones booleanas. Por: Helbert Eduardo Espitia C. y José Jairo Soriano Mendez (Universidad Distrital Francisco José de Caldas, Bogotá, Colombia. 2011)

En la primera parte de este trabajo se presentó el diseño del controlador Booleano, posteriormente se efectuó el diseño del controlador difuso partiendo del controlador Booleano realizado anteriormente. Las superficies de control obtenidas con y sin simplificación de términos evidenciaron las características que tuvo la metodología propuesta. Adicionalmente se analizaron los efectos que tuvieron las simplificaciones realizadas en las reglas de control. En este trabajo se concluyó diciendo que se mejoró el comportamiento dinámico del sistema empleando conjuntos Booleanos adyacentes.

- Tesis de: Sistema de control de temperatura para incubadora avícola con control difuso y la plataforma Labview. Por: Adán Cortes de la Cruz y Oscar Arenas Rosales (Universidad Autónoma de Zacatecas. 2011)

En esta tesis se utilizó la plataforma LabVIEW para el sistema de control Fuzzy, este sistema conto con un control digital de encendido y apagado para el encendido de humidificadores; en este proyecto se utilizó la tarjeta de adquisición de datos NI USB DAQ 6009) la cual era de la misma familia (National Instruments). Además de ello en este proyecto se mencionó que hubo problemas con el tiempo en la que la humedad tarda en llenar el recinto de la incubadora, provocando un retardo en la detección del sensor de temperatura o humedad del sistema.

1.3 Importancia y Utilidad del Estudio

Este proyecto se permitiría ahorros económicos para las personas dedicadas al campo biológico, biomédico, porque solo tendrían que adquirir una estufa como esta para múltiples cultivos, y ya no comprar varias para distintos tipos de cultivos. Otra importante contribución sería el mejor monitoreo del equipo, ya que este contaría con comunicación inalámbrica. Finalmente brindaría una configuración más sencilla y entendible al usuario.

1.4 Objetivos de la investigación

1.4.1 Objetivo general

Diseñar e implementar una estufa que pueda controlar cambios de temperaturas dentro de un rango y pueda tener un monitoreo inalámbrico.

1.4.2 Objetivos específicos

- a) Obtener una configuración más sencilla para los parámetros de funcionamiento de trabajo de una estufa.
- b) Controlar la razón de cambio de temperatura vs tiempo en una estufa de cultivo.

c) Elaborar una página web en donde se observe los parámetros de funcionamiento de la estufa de cultivo.

CAPITULO II

REVISIÓN DE LITERATURA

Para la elaboración de esta investigación, fue necesario el uso y análisis de sistemas discretos, modelado matemático, controlador óptimo, comparación entre una estufa de cultivo y una incubadora, comunicación WiFi y otros temas más; los cuales serán detallados a continuación.

2.1 Sistemas de Control Digital

2.1.1 Diseño de Sistemas de Control Digital

La Figura 1 muestra la estructura de un sistema de control digital, cuyo objetivo de control es lograr que la señal de salida Y del proceso (de carácter continuo, como en la mayoría de los casos) siga a la señal de referencia r cumpliendo ciertas especificaciones de diseño, como: mínimo sobre impulso, mínimo tiempo de estabilización y/o error en estado estacionario nulo en Y . En otras palabras, el algoritmo de control implementado, en este caso digitalmente, debe ser capaz de crear una señal de control U (la variable

manipulada), que actuando sobre el proceso a mediante un actuador, minimice la señal de error e . El bloque de filtraje es útil para eliminar las componentes contaminantes de alta frecuencia de la señal de ruido de medición n . El filtro de ruido puede ser analógico o digital y su inclusión depende de la magnitud del ruido y su relevancia dentro del funcionamiento del sistema de control.

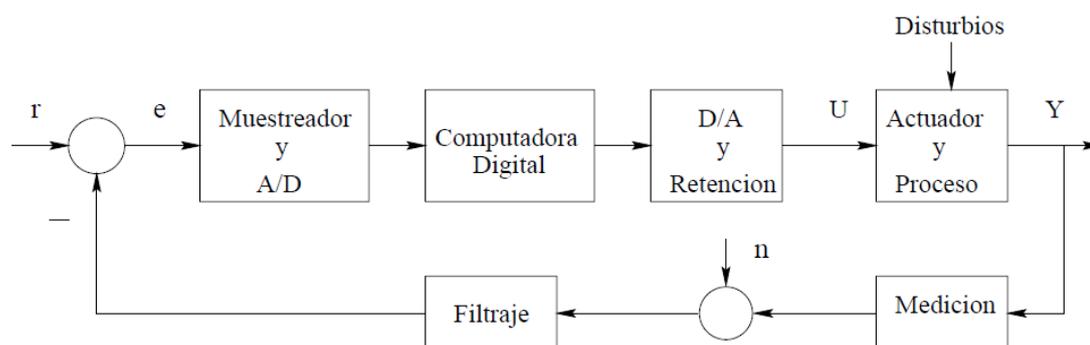


Figura 1: Estructura de un sistema de control digital

Fuente: Moreno, 2001, pág. 30

Esta estructura de control puede ser implementada en un dispositivo digital, como por ejemplo una computadora, un micro controlador como PIC; o como en el caso de este proyecto en un Arduino UNO. El procesamiento digital de un algoritmo de control requiere el uso de conversores A/D y D/A; en este proyecto utilizamos el puerto analógico digital con el que cuenta el Arduino, también necesitamos de retenedores de señal.

La estructura de configuración mostrada en la figura anterior no es única, por ejemplo en el presente proyecto la señal de referencia y de comparación serian originadas en un micro controlador como el Arduino. Por lo que tenemos que seleccionar un adecuado algoritmo de control que sea capaz de cumplir los objetivos plantados con el menor gasto y el mayor rendimiento posible.

2.1.2 Concepto de Estado y Variables de Estado

La dinámica de un proceso multivariable puede ser representada en el espacio de estado, mediante dos conjuntos de ecuaciones diferenciales ordinarias de primer orden, denominadas ecuaciones de estado y de salida, como sigue:

$$\begin{aligned} \dot{X} &= f(X, U, v, t) \\ \dot{Y} &= h(X, U, w, t) \dots\dots\dots (Ec.1) \end{aligned}$$

Donde f y h son funciones vectoriales de variable vectorial de orden n y r respectivamente, X es el vector de estado de orden n , U es el vector de control de orden m , Y es el vector de salida de orden r , v es el vector de disturbios (de orden n) en los estados, y w es el vector de disturbios (de orden r) en las salidas. El tiempo t se incluye como parte de las ecuaciones para indicar que pueden existir parámetros variantes con el tiempo. Po lo que sería bueno definir el concepto de estado:

“El vector de estado $X = [X_1 \dots X_n]^T$ de un proceso (donde el superíndice T indica transpuesta) es el mínimo conjunto de variables, las variables de estado $X_1 \dots X_n$, las cuales contienen información suficiente acerca de la historia pasada del proceso. Esta información permite computar todos los futuros estados del proceso, asumiendo por supuesto, que todas las futuras entradas U son también conocidas, como del mismo modo lo son las ecuaciones dinámicas que describen dicho sistema. El número n de variables de estado define el orden o la dimensión del sistema.” (Moreno, 2001, pág. 31)

Cuando el sistema es de orden 2, el espacio de estado es conocido como plano de fase, los puntos de equilibrio se determinan cuando el estado del proceso no cambia gracias a la acción de una fuerza de entrada \bar{U} . Por lo que los puntos de equilibrio de \bar{X} se determinan de:

$$\dot{\bar{X}} = f(\bar{X}, \bar{U}) = 0 \dots\dots\dots (Ec.2)$$

2.1.3 Linealización de Procesos

La representación linealizada de un proceso en el espacio de estado sin la presencia de parámetros variantes en el tiempo es como sigue:

$$\begin{aligned} \dot{x} &= Ax + Bu + Ev \\ y &= Cx + Du + Fw \dots\dots\dots(Ec.3) \end{aligned}$$

Donde A es la matriz de estado, B es la matriz de control, E es la matriz de disturbios en los estados, C es la matriz de salida de los estados, D es la matriz de salida de las entradas y F es la matriz de disturbios en las salidas. Las dimensiones de dichas matrices son: A_{nn} , B_{nm} , E_{nn} , C_{rn} , D_{rm} y E_{rr} , donde el primer subíndice es el número de filas de la matriz y el segundo, el número de columnas.

Si no consideráramos la presencia de disturbios y el proceso alrededor de un estado de equilibrio (\bar{X}, \bar{U}) , las matrices A , B , C y D pueden ser determinadas evaluando las siguientes matrices jacobianas:

$$\begin{aligned} A &= \begin{bmatrix} \frac{\partial f_1}{\partial X_1} & \dots & \frac{\partial f_1}{\partial X_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial X_1} & \dots & \frac{\partial f_n}{\partial X_n} \end{bmatrix} \\ B &= \begin{bmatrix} \frac{\partial f_1}{\partial U_1} & \dots & \frac{\partial f_1}{\partial U_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial U_1} & \dots & \frac{\partial f_n}{\partial U_m} \end{bmatrix} \\ C &= \begin{bmatrix} \frac{\partial h_1}{\partial X_1} & \dots & \frac{\partial h_1}{\partial X_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial h_r}{\partial X_1} & \dots & \frac{\partial h_r}{\partial X_n} \end{bmatrix} \end{aligned}$$

$$D = \begin{bmatrix} \frac{\partial h_1}{\partial U_1} & \dots & \frac{\partial h_1}{\partial U_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial h_r}{\partial U_1} & \dots & \frac{\partial h_r}{\partial U_m} \end{bmatrix}$$

(Ec.4)

Por lo que las ecuaciones (2 y 3) quedarían de la siguiente forma:

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

$$y = Cx + Du \dots \dots \dots (Ec.5)$$

Donde la matriz de transferencia de procesos multivariables se determina empleando la ecuación matricial:

$$Y(s) = [C(sI - A)^{-1}B + D]U(s) \dots \dots \dots (Ec.6)$$

Con la siguiente ecuación característica:

$$\det(sI - A) = 0 \dots \dots \dots (Ec.7)$$

Esas son las ecuaciones matriciales más usadas en el diseño de un controlador óptimo, cuando trabajamos en tiempo discreto, nos daremos cuenta que *A* cambiara por *G* y *B* por *H* respectivamente; y al agregar el efecto del tiempo muerto, solo varia un poco al eliminar el efecto de la matriz *D*.

El proceso descrito por la ecuación (6) es estable, siempre que todos los eigenvalores de la ecuación característica (7) posean parte real negativa. Cuando al menos uno de tales eigenvalores posea parte real positiva o cero, entonces el proceso es inestable.

2.1.4 El Espacio de Estado Discreto

Discretizando la ecuación (5) para un tiempo discreto $k = t/k$ y asumiendo que entre muestras la señal de control *u* es constante, entonces para la señal de entrada $u(t) = u(kt)$, $kT \leq (k + 1)T$ y para un estado inicial $x(kT) = u(kT)$, la ecuación de estado se convierte en:

$$x(t) = \phi(t - kT)x(t_0) + u(kt) \int_{t_0}^t \phi(t - \tau)Bd\tau \dots \dots \dots (Ec.8)$$

Si sólo interesa la solución para $t = (k + 1)T$ y empleando el cambio de variables que sigue $t = (k + 1)T - \tau, d\lambda = -d\tau, d\lambda = -d\tau$ y después de resolverlo recursivamente se obtiene:

$$x(k + 1) = Gx(k) + Hu(k) \qquad y(k) = Cx(k) + Du(k) \qquad (Ec.9)$$

Si las resolvemos recursivamente y las pasamos posteriormente al dominio de z la matriz de transferencia de pulso se convierte en:

$$y(z) = G_p(z)u(z) = [C(zI - G)^{-1}H + D]u(z) \dots \dots \dots (Ec.10)$$

Con la siguiente ecuación característica:

$$\det(zI - G) = 0 \dots \dots \dots (Ec.11)$$

2.1.5 Controlabilidad

“Un proceso dinámico lineal se dice que es controlable, si es que existe un vector $u(k)$ capaz de trasladar el estado del proceso desde un estado inicial $x(0)$ hacia un estado final $x(N)$ en un tiempo finito N . “ (Moreno, 2001, pág. 64)

Para que un proceso sea completamente controlable, se debe cumplir:

$$M = [H \ GH \ \dots \ G^{N-1}H] \dots \dots \dots (Ec.12)$$

$$\text{rango}(M) = n \dots \dots \dots (Ec.13)$$

Donde n es el orden del proceso.

2.1.6 Observabilidad

Un proceso lineal dinámico con salida $y(k)$ se llama observable si algún estado $x(k)$ puede ser obtenido a partir de un número finito de salidas $y(k), y(k - 1) \dots y(k - n)$.

Considerando las siguientes ecuaciones:

$$\begin{aligned}
 x(k + 1) &= Gx(k) + Hu(k) & y(k) &= Cx(k) \\
 & & & \text{(Ec.14)}
 \end{aligned}$$

Luego podemos obtener Y_n de forma recursiva de las 2 ecuaciones anteriores:

$$Y_n = Nx(k) + SU_n \dots \dots \dots \text{(Ec.15)}$$

Dónde:

$$Y_n = [y(k) \ y(k + 1) \ \dots \ y(k + n - 1)]^T \dots \dots \dots \text{(Ec.16)}$$

$$N = [C \ CG \ \dots \ CG^{n-1}]^T = [C^T \ G^T C^T \ \dots \ (G^T)^{n-1} C^T] \dots \dots \dots \text{(Ec.17)}$$

$$S = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & CH \\ \vdots & \ddots & & \vdots & \\ 0 & CH & CGH & \dots & CG^{n-2}H \end{bmatrix} \dots \dots \dots \text{(Ec.18)}$$

Despejamos el vector incógnita:

$$x(k) = N^{-1}[Y_n - SU_n] \dots \dots \dots \text{(Ec.19)}$$

De lo cual concluimos diciendo que un proceso dinámico es observable si la matriz de observabilidad N cumple:

$$\text{rango}[N] = n$$

2.1.7 Modelado Matemático

En esta parte del informe se verá el modelo dinámico de una estufa de cultivo, el cual es muy similar al de una estufa de cultivo con su correspondiente discretización, también se verá su respuesta a la entrada del escalón, determinando su controlabilidad y obserbabilidad empleando como herramienta un simulador como el Matlab.

Las ecuaciones que gobiernan el funcionamiento de un horno muy próximo a la realidad sin la presencia de aire forzado y otros parámetros como el tiempo de difusión del calor dentro de la cámara, son las siguientes:

$$c \frac{d}{dt}(\theta_i - \theta_o) = Q_c - Q_e \dots\dots\dots(\text{Ec.20})$$

$$Q_c = \frac{V_c^2}{R} \dots\dots\dots(\text{Ec.21})$$

$$Q_e = \frac{\theta_i - \theta_o}{r} \dots\dots\dots(\text{Ec.22})$$

Dónde:

- θ_i es la temperatura de la cámara en grados Kelvin o Celsius.
- θ_o es la temperatura ambiente en K o °C.
- Q_c es el flujo de calefacción o calor entregado de por la resistencia calefactora al interior de la estufa, la cual está en J/s o watts.
- Q_e es el flujo de perdida en J/s o watts.
- c es la capacidad térmica en J/K.
- r es la resistencia térmica en K/watt.
- R es la resistencia eléctrica en ohm.
- V_c es el voltaje de calefacción.

Para este modelado se consideró constante la temperatura ambiente θ_o , ya que en la realidad nos daremos cuenta que las estufas de cultivo se encuentran en un ambiente en el cual no cambia mucho la temperatura; porque la dinámica del horno se describiría como:

$$\frac{d\theta_i}{dt} + \frac{1}{rc} \theta_i = \frac{1}{rc} \theta_o + \frac{1}{Rc} V_c^2 \dots\dots\dots(\text{Ec.23})$$

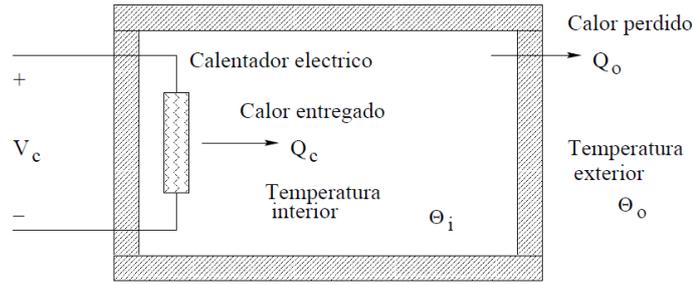


Figura 2: Horno de laboratorio o estufa

Fuente: Moreno, 2001, pág. 102

y cuando θ_i alcanza su estado estacionario $\bar{\theta}_i$:

$$\bar{\theta}_i = \bar{\theta}_o + \frac{r}{R} V_c^2$$

.....(Ec.24)

Tabla 1: Tabla de la disminución de temperatura al apagar el Horno de laboratorio

Tiempo	Temperatura (°C)
14:23:10	120.0
14:31:00	108.0
14:39:30	92.0
14:48:35	80.5
15:05:00	63.0
15:34:00	40.5

Fuente: Moreno, 2001, pág. 103

Tabla 2: Temperaturas del horno en estado estable

Voltaje	Temperatura (°C)
9.8	34.1
20.0	50.5
29.5	74.0
40.0	110.0

Fuente: Moreno, 2001, pág. 103

La Tabla 1 contiene datos experimentales de la caída de temperatura cuando se apaga una estufa estándar. La Tabla 2 tabula el voltaje de calentamiento versus la temperatura

interior en estado estable. Para ambas tablas la temperatura exterior es de 30 °C. Por lo que la dinámica del proceso resulta:

$$\frac{d\theta_i}{dt} = -\frac{1}{3000}\theta_i + \frac{1}{3000}\theta_o + \frac{1}{60000}V_c^2 = f(\theta_i, \theta_o, V_c) \dots\dots\dots(\text{Ec.25})$$

Para efectos de linealización podemos definir las siguientes variables:

$$\theta_i = \theta_i - \bar{\theta}_i; \qquad \theta_o = \theta_o - \bar{\theta}_o; \qquad v_c = V_c - \bar{V}_c \dots\dots\dots(\text{Ec.26})$$

Entonces la ecuación linealizada empleando el jacobiano para el punto de operación, resulta:

$$\frac{d\theta_i}{dt} = -\frac{1}{3000}\theta_i + \frac{1}{3000}\theta_o + \frac{1}{1500}v_c^2 \dots\dots\dots(\text{Ec.27})$$

Esta ecuación será la que utilizaremos para diseñar nuestro controlador.

2.1.8 Control Óptimo

El proceso de diseño del controlador óptimo requiere de la formulación de una función de costo, en la cual trataremos de encontrar un extremo de la misma. La cual posee la forma:

$$J = \sum_{k=0}^N L(x(k), r(k), u(k)) \dots\dots\dots(\text{Ec.28})$$

Dónde:

- k es el tiempo discreto.
- N es el tiempo discreto final
- $x(k)$ es el tiempo discreto final

- $r(k)$ es el vector de referencia
- $x(k)$ es el vector de control optimo

A continuación desarrollare el diseño de un control óptimo cuadrado discreto, el resultado del diseño proporciona una matriz de ganancia que multiplicada por el vector de estado del proceso, resulta en una ley de control optima cuadrática. Esta se determina utilizando un observador óptimo.

El controlador cuadrático no estacionario evoluciona en tiempo finito y su matriz de ganancias de realimentación varía en el tiempo, mientras que el controlador cuadrático estacionario mantiene su matriz de ganancia constante. Para el diseño del controlador en este borrador nosotros usaremos la última ya que esta nos da más simplicidad en el diseño y también trabaja para regímenes en estado permanente.

2.1.9 Control Óptimo Cuadrático Estacionario

Para un tiempo N infinito la solución del control optimo cuadrático pasa a ser una solución de estado estacionario. Por consiguiente, la función de costo para el estado estacionario toma la forma:

$$J = \frac{1}{2} \sum_{k=0}^{\infty} [x^T(k)Qx(k) + u^T(k)Rx(k)] \dots\dots\dots(\text{Ec.29})$$

Por otra parte, en el estado estacionario la matriz $P(k)$ es una matriz constante P . Por lo que la ecuación de Riccati en estado estacionario toma la forma:

$$P = Q + G^T P G - G^T P H [R + H^T P H]^{-1} H^T P G \dots\dots\dots(\text{Ec.30})$$

Y la matriz de ganancia resulta:

$$K = [R + H^T P H]^{-1} H^T P G \dots\dots\dots(\text{Ec.31})$$

Podemos hallar P de forma recursiva, luego podemos obtener la matriz de ganancia K usando la ecuación (31). Estas últimas ecuaciones solo variaran un poco para los demás

casos, manteniendo su estructura; a continuación una figura de un sistema de control lazo cerrado para una mejor comprensión del mismo:

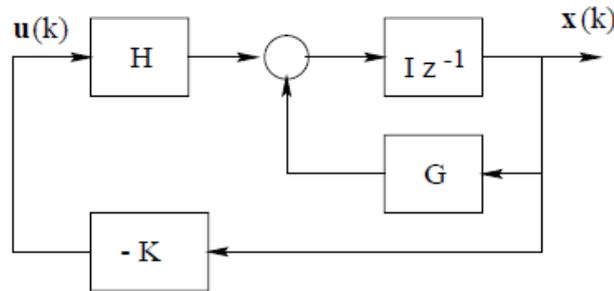


Figura 3: Sistema de control óptimo a lazo cerrado
Fuente: Moreno, 2001, pág. 128

2.1.10 El Regulador Óptimo Proporcional

En donde la salida de un sistema sigue a la señal de referencia $r(k) = \mu(k)$, donde $\mu(k)$ es la señal escalón. A continuación, damos un esquema de este regulador:

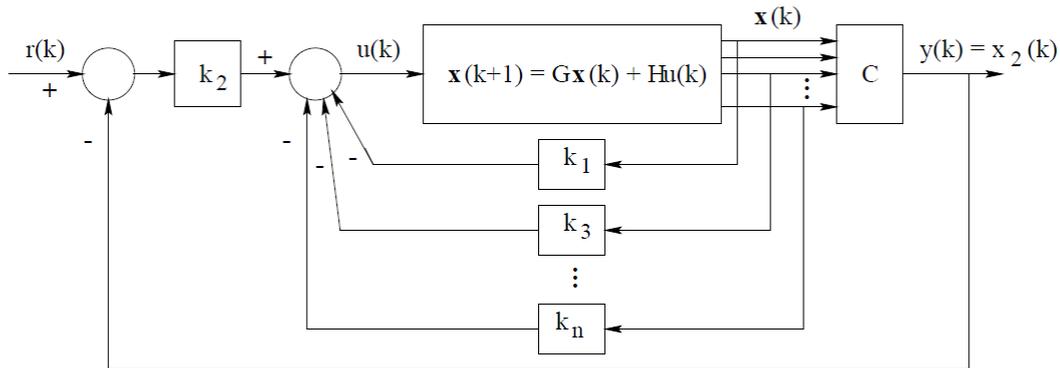


Figura 4: Esquema del regulador óptimo proporcional
Fuente: Moreno, 2001, pág. 129

Obteniendo las siguientes ecuaciones:

$$u(k) = -Kx(k) + k_2r(k) \dots \dots \dots (Ec.32)$$

$$x(k + 1) = (G - HK)x(k) + Hk_2r(k) \dots \dots \dots (Ec.33)$$

2.1.11 Observador óptimo cuadrático

Se utiliza cuando haya variables del vector de estado que no puedan ser medidas de manera directa, el cual tiene su propia matriz de ganancia K_e . Al cual lo representamos en la Figura 5:

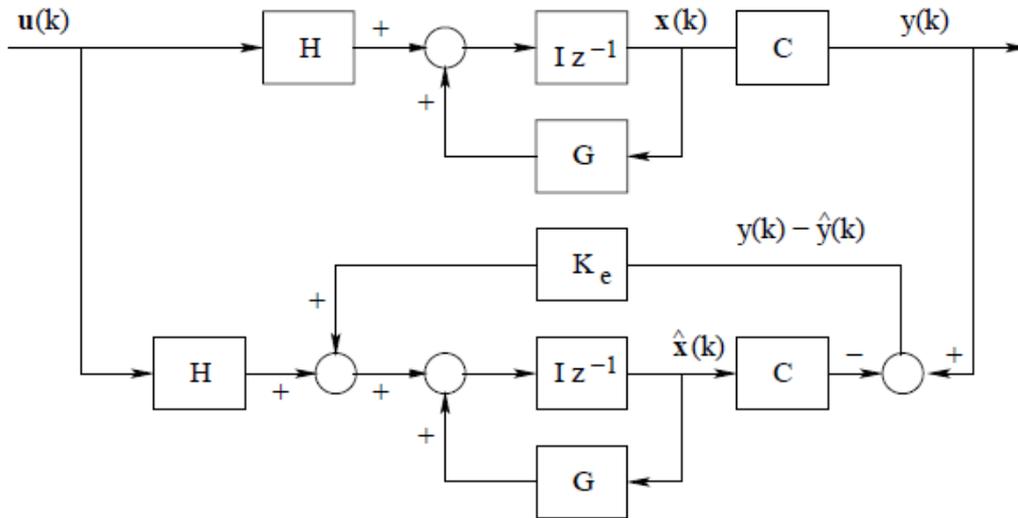


Figura 5: Diagrama de bloques del observador de estados

Fuente: Moreno, 2001, pág. 137

De la cual podemos formular que:

$$x(k + 1) = Gx(k) + Hu(k) \dots \dots \dots (Ec.34)$$

$$y(k) = Cx(k) \dots \dots \dots (Ec.35)$$

Y que la ecuación del observador toma la forma:

$$\hat{x}(k + 1) = G\hat{x}(k) + Hu(k) + Ke[y(k) + C\hat{x}(k)] \dots \dots \dots (Ec.36)$$

Reemplazando la ecuación (35) en (34) y restando la ecuación resultante de (36), podemos obtener la ecuación del error del observador:

$$e(k + 1) = [G - KeC]e(k) \dots \dots \dots (Ec.37)$$

$$e(k) = x(k) - \hat{x}(k) \dots \dots \dots (Ec.38)$$

Que después de varios remplazos y transformaciones nos da la siguiente ecuación de Riccati:

$$Pe = Qe + GPeGT - GPeCT[Re + CPeCT]^{-1}CPeGT \dots \dots \dots (Ec.39)$$

$$Ke = [Re + CPeCT]^{-1}CPeGT \dots \dots \dots (Ec.40)$$

2.1.12 El Regulador Óptimo Proporcional Integral

Es necesario agregar un regulador integral al proporcional anteriormente nombrado, para reducir su error en estado estacionario. A continuación, se muestra la figura de un regulador óptimo con una acción integral.

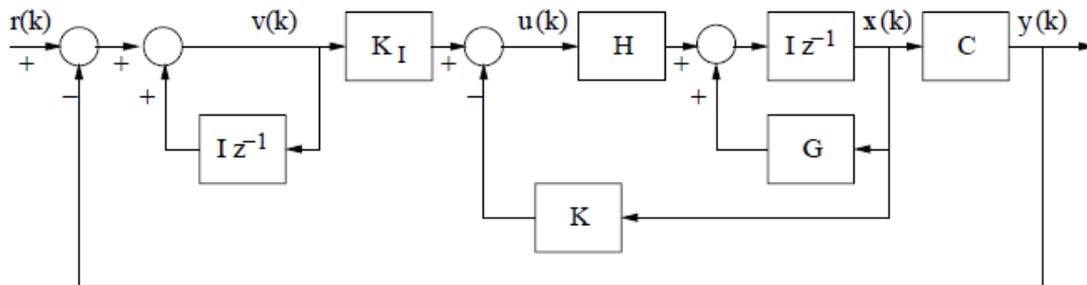


Figura 6: El regulador proporcional integral

Fuente: Moreno, 2001, pág. 140

Siendo la señal de referencia $r(k)$ un escalón unitario, de la figura anterior podemos deducir que las ecuaciones de estado y de salida son:

$$\begin{aligned} x(k + 1) &= Gx(k) + Hu(k) \\ y(k) &= Cx(k) \dots\dots\dots(\text{Ec.41}) \end{aligned}$$

La señal de control $u(k)$ viene dada por:

$$u(k) = -Kx(k) + K_I v(k) \dots\dots\dots(\text{Ec.42})$$

Y la ecuación para el integrador resulta:

$$v(k) = v(k - 1) + r(k) - y(k) \dots\dots\dots(\text{Ec.43})$$

Para obtener las siguientes matrices:

$$\begin{aligned} \begin{bmatrix} x(k + 1) \\ v(k + 1) \end{bmatrix} &= \begin{bmatrix} G - HK & HK_I \\ -CG + CHK & 1 - CHK_I \end{bmatrix} \begin{bmatrix} x(k) \\ v(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r(k + 1) \\ y(k) &= [C \quad 0] \begin{bmatrix} x(k) \\ v(k) \end{bmatrix} \dots\dots\dots(\text{Ec.44}) \end{aligned}$$

En estado de equilibrio (cuando $k \rightarrow \infty$) estas ecuaciones se convierten:

$$\begin{bmatrix} X(\infty) \\ v(\infty) \end{bmatrix} = \begin{bmatrix} G - HK & HK_I \\ -CG + CHK & 1 - CHK_I \end{bmatrix} \begin{bmatrix} x(\infty) \\ v(\infty) \end{bmatrix} + \begin{bmatrix} 0 \\ r \end{bmatrix} \dots\dots\dots(\text{Ec.45})$$

Si definimos:

$$x(k) - x(\infty) = x_e(k)$$

$$v(k) - v(\infty) = v_e(k) \dots \dots \dots (Ec.46)$$

Y restamos los 2 grupos de matrices anteriores obtendremos:

$$\begin{bmatrix} X_e(k+1) \\ v_e(k+1) \end{bmatrix} = \begin{bmatrix} G - HK & HK_I \\ -CG + CHK & 1 - CHK_I \end{bmatrix} \begin{bmatrix} X_e(k) \\ v_e(k) \end{bmatrix}$$

$$= \begin{bmatrix} G & 0 \\ -CG & I \end{bmatrix} \begin{bmatrix} X_e(k) \\ v_e(k) \end{bmatrix} + \begin{bmatrix} H \\ -CH \end{bmatrix} \begin{bmatrix} -K & K_I \end{bmatrix} \begin{bmatrix} X_e(k) \\ v_e(k) \end{bmatrix} \dots \dots \dots (Ec.47)$$

Esta última ecuación puede ser escrita como:

$$\xi(k+1) = \tilde{G}\xi(k) + \tilde{H}\omega(k) \dots \dots \dots (Ec.48)$$

$$\omega(k) = -\tilde{K}\xi(k) \dots \dots \dots (Ec.49)$$

Dónde:

$$\xi(k) = \begin{bmatrix} x_e(k) \\ v_e(k) \end{bmatrix} \quad \tilde{G} = \begin{bmatrix} G & 0 \\ -CG & I \end{bmatrix}$$

$$\tilde{H} = \begin{bmatrix} H \\ -CH \end{bmatrix} \quad \tilde{K} = [K \quad -K_I]$$

En el presente proyecto también utilicé un control óptimo proporcional integral, el cual fue diseñado usando el software Matlab.

Tabla 3: Aquí se muestra el programa completo de la simulación del controlador

```
%Parametros del proceso
A=[-1/3000];
B=[1/3000]; E=[1/1500];
C=[1];
D=[0];
T=0.06;%Tiempo en segundos
%Convercion al estado discreto
[G,H,C,D]=c2dm(A,B,C,D,T,'zoh');
[G,F,C,D]=c2dm(A,E,C,D,T,'zoh');
G1=[ G, 0;
    -C*G, 1];
H1=[ H;
    -C*H];
M=[H1 G1*H1];
rM=rank(M);%Probando controlabilidad
%Cálculo de la ganancia del controlador
Q=[1000, 0;
    0, 0.1];
```

```

R=[0.001];
% P=zeros(2,2);
% for i=1:1000
%   P=Q+G1'*P*G1-G1'*P*H1*inv(R+H1'*P*H1)*H1'*P*G1;
% end
% K1=inv(R+H1'*P*H1)*H1'*P*G1
[KK,PP]=dlqr(G1,H1,Q,R);
%Cálculo de la ganancia del observador
Qe=[1000];
Re=[10];
% Pe=[0];
% for i=1:1000
%   Pe=Qe+G*Pe*G'-G*Pe*C'*inv(Re+C*Pe*C')*C*Pe*G';
% end
% Ke1=inv(Re+C*Pe*C')*C*Pe*G';
[Ke,PPe]=dlqr(G',C',Qe,Re);

%Simulacion del control de la estufa-multiuso
%Entrada de variables
Camb=[5,25,45];
Tem=[40,50,30];
Raz=[8,2,4];
r_z=[1,1,1];
RZ=1000./Raz;
Cam=1000*Camb;
if Tem(1)<0
    r_z(1)=-1;
else
    r_z(1)=1;
end
r_z(2)=Tem(2)-Tem(1);
if r_z(2)<0
    r_z(2)=-1;
else
    r_z(2)=1;
end
r_z(3)=Tem(3)-Tem(2);
if r_z(3)<0
    r_z(3)=-1;
else
    r_z(3)=1;
end
%Condiciones iniciales
x1=0; x2=0; xb=0;
v=0; r=0;
rep=60000; u1=20;%1hora rep=56250;
%Cuerpo del simulador
k_z=1;
for k=1:rep

```

```

if (Cam(1)<=k)&&(Cam(2)>k)
    if r~=Tem(1)
        if k_z<RZ(1)
            k_z=k_z+1;
        else
            k_z=1;
            r=r+r_z(1);
        end
    end
end
if (Cam(2)<=k)&&(Cam(3)>k)
    if r~=Tem(2)
        if k_z<RZ(2)
            k_z=k_z+1;
        else
            k_z=1;
            r=r+r_z(2);
        end
    end
end
if k>=Cam(3)
    if r~=Tem(3)
        if k_z<RZ(3)
            k_z=k_z+1;
        else
            k_z=1;
            r=r+r_z(3);
        end
    end
end

u = -KK(1)*xb - KK(2)*v;
xb = G*xb + H*u + Ke*(x1 - C*xb);
x1 = G*x1 + H*u + F*u1;
v = v + r - x1;
y(k) = x1; U(k) = u;
end
t=linspace(0,T*rep,rep);
figure(1)
plot(t,y(1:rep)); grid
ylabel('Temperatura de la estufa')
xlabel('Tiempo en (s)')
print -deps -f spoz
    
```

Elaboración: Propia

A continuación se mostrara el procedimiento realizado para el diseño del controlador de este proyecto:

2.1.13 Parámetros del proceso

$$A=[-1/3000];$$

$$B=[1/3000]; E=[1/1500];$$

$$C=[1];$$

$$D=[0];$$

$$T=0.06;$$

Y después de pasar el proceso a un estado discreto, obtendremos las matrices G y H , seguidamente se obtiene las nuevas matrices del controlador con el efecto producido por el observador, las cuales aumentaran en un grado.

Después encontré la ganancia del controlador con las siguientes ecuaciones de ponderación:

$$Q = \begin{bmatrix} 1000 & 0 \\ 0 & 0.1 \end{bmatrix} \quad R = [0.001]$$

También obtuve la ganancia del observador con estas ecuaciones de ponderación:

$$Q_e = [1000] \quad R_e = [10]$$

Finalmente lo elaboré en Matlab obteniendo la siguiente grafica de respuesta a un escalón unitario:

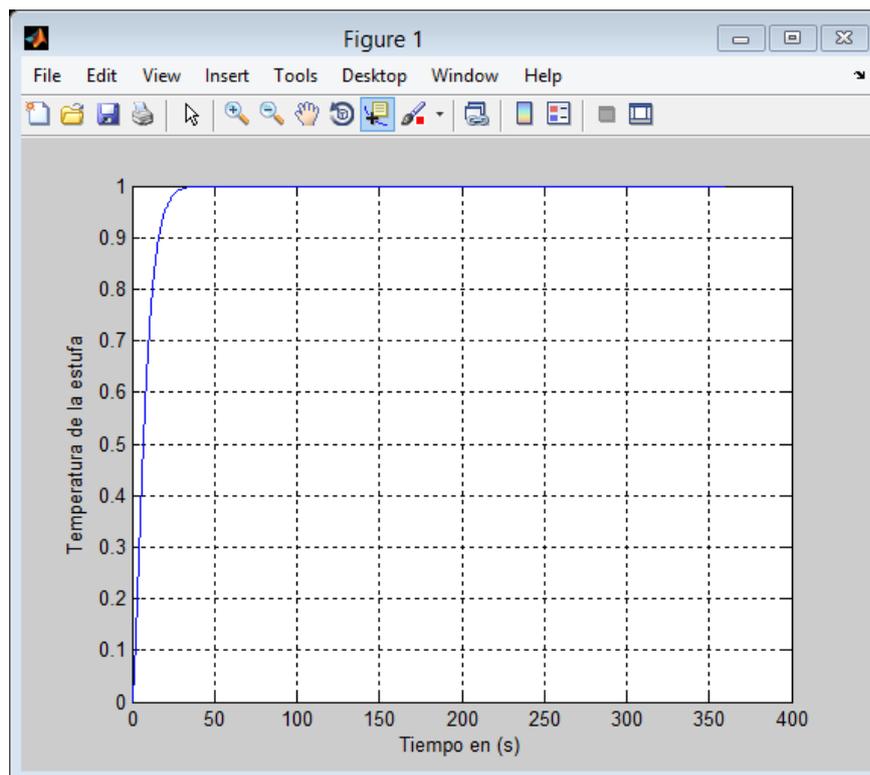


Figura 7: Respuesta al escalón unitario en Matlab resultante de 6000 muestras
Elaboración: Propia

2.2 Comunicación WIFI

WIFI (contracción de Wireless Fidelity, en español Fidelidad inalámbrica), es el nombre dado por WECA (Wireless Ethernet Compatibility Alliance) a los equipos que cumplan con la norma 802.11. Debido al abuso de lenguaje y por razones de marketing una red con norma 802.11 se dice Wi-Fi.

Gracias al Wi-Fi es posible crear WLAN (Wireless Local Area Network). Por lo que es posible la comunicación entre equipos portátiles en un radio de entre 20 y 50 metros.

2.2.1 Bandas ISM/UNII

La mayoría de dispositivos inalámbricos utilizan frecuencias de radio adjudicadas mediante una licencia. Wi-Fi utiliza anchos de banda que no requieren licencia.

- Las bandas ISM (Industrial, Scientific and Medical) admiten el uso de las porciones 2,4 – 2,5 GHz, 5,8 GHz y muchas otras frecuencias no utilizadas en WiFi.

- Las bandas UNII (Unlicensed National Information Infrastructure) permiten el uso sin licencia en otras porciones del espectro de 5 GHz.

Algunos protocolos de redes inalámbricas:

- 802.11a en las bandas no licenciadas de 5 GHz permite hasta 54Mbps
- 802.11b en la banda no licenciada de 2,4 GHz permite hasta 11 Mbps
- 802.11g en la banda no licenciada de 2,4 GHz permite hasta 54 Mbps
- 802.11n permite hasta 600 Mbps en las bandas licenciadas a 2.4 GHz y 5 GHz
- 802.16 (WiMAX) no es Wi-Fi; es una tecnología que utiliza tanto bandas licenciadas y no licenciadas.

2.2.2 Tasas de Transmisión

Las tasas de transmisión anteriores se refieren a la velocidad de transmisión de caracteres por segundo, no al verdadero caudal de transmisión de información a nivel TCP/IP. La diferencia es la que se conoce como protocolo overhead y es utilizada por el protocolo WiFi para manejar retransmisiones, colisiones y en general la gestión de datos del enlace. Una regla general es que el caudal máximo a nivel TCP/IP es la mitad de la tasa de velocidad de envío de símbolos.

Por ejemplo un enlace 802.11b tiene un rendimiento máximo de transmisión de 5Mbps, un enlace 802.11g tiene un rendimiento máximo práctico de unos 25Mbps.

2.2.3 CSMA vs. TDMA

El 802.11 utiliza CSMA (Carrier Sense Multiple Access) para evitar las colisiones de transmisión. Antes de que un nodo pueda transmitir debe escuchar en el canal por las posibles transmisiones de otros radios. El nodo sólo puede transmitir cuando el canal está desocupado.

Otras tecnologías como WiMAX y AirMAX usan TDMA (Time Division Multiple Access). TDMA particiona el acceso al canal en múltiples divisiones de tiempo, y

asigna divisiones de tiempo a cada nodo de la red. Cada nodo transmite solo en su ranura de tiempo y de esta manera se evitan las colisiones.

WiFi solo provee funcionamiento en la capa 1 y 2 del protocolo TCP/IP (Transmission Control Protocol). En la capa uno de configurarse adecuadamente los siguientes parámetros para poder establecer conectividad:

- Canal de radio
- Modo de operación de radio
- Nombre de la red
- Tipo de seguridad

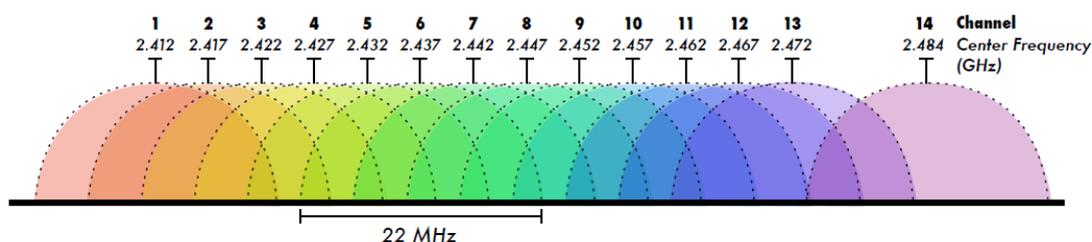


Figura 8: Canales en 802.11 con diferentes frecuencias centrales

Fuente: Ermanno, 2010, pág. 9

Los dispositivos WiFi deben usar el mismo canal para poder comunicarse. Ellos envían y reciben en el mismo canal, por lo que solo un dispositivo puede transmitir en un instante determinado. Esta modalidad de transmisión se llama halfduplex (Ermanno, 2010, pág. 9)

2.2.4 Topologías de redes inalámbricas

- IBSS (Independent Basic Service Set) o Ad Hoc BSS
- BSS (Basic Service Set) o Infraestructure BSS
- ESS (Extended Service Sets) un Service Set es una agrupación lógica de dispositivos.

IBSS: Independent Basic Service Set o Ad-hoc network, grupo de estaciones 802.11 comunicándose directamente entre ellas que es generalmente pequeña y duran poco tiempo. No hay límite al número de miembros, BSSID (ID del BSS) es elegido al azar.

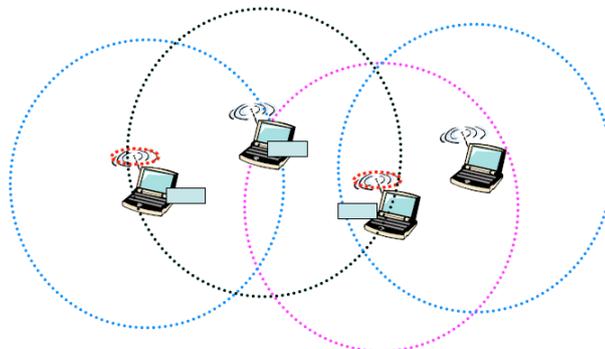


Figura 9: Diagrama que ilustra los campos de acción en un red Ad-hoc

Fuente: upna, pág. 11

BSS: Basic Service o Infraestructure, incluye un punto de acceso (AP) a través del cual los clientes se comunican. El AP puede contener un uplink que conecta a la red cableada, BSSID es la MAC Wi-Fi del AP.

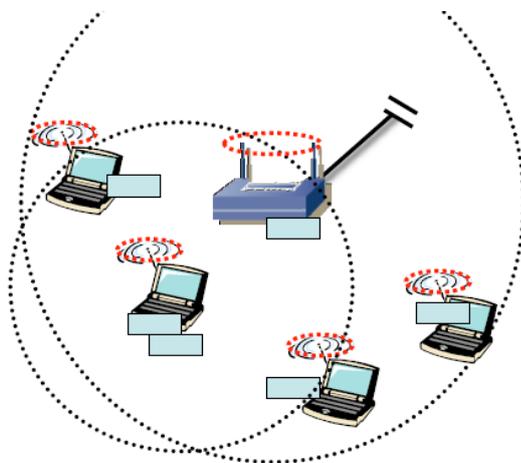


Figura 10: Diagrama que muestra los campos de una red con una estación especializada AP (Punto de acceso)

Fuente: upna, pág. 12

ESS: Extended Service Set, varias BSS conectadas por sus interfaces de uplink, todas emplean el mismo ESSID, se comunican a través del DS (Distribution System) el cual está creado en base a Ethernet.

2.2.5 Modos de funcionamiento en WiFi

Los dispositivos WiFi pueden operar en alguno de los siguientes modos:

Modo Master: También llamado modo AP, se usa para instalar una red con un AP que conecta a diferentes clientes.

El AP crea una red con un nombre específico (denominado SSID o ESSID) y un canal sobre el cual se ofrecen los servicios de la red. Los dispositivos en modo master solo pueden comunicarse con aquellos dispositivos conectados a ellos que estén en modo managed.

Modo Managed: Es llamado modo cliente. Los dispositivos inalámbricos en modo managed se unirán en una red creada por el master y automáticamente cambiarán el canal para ajustarse al del master.

De los clientes que usan un determinado AP se dice que están asociados con él.

Los radios en modo managed no pueden comunicarse directamente entre sí y sólo se pueden comunicar con el master al cual están asociados.

Modo ad_hoc: El modo Ad-hoc se usa para crear redes en malla, en donde:

- No hay dispositivos en modo AP
- La comunicación se realiza entre los nodos de forma directa.

Los dispositivos deben de estar dentro de un rango de cobertura para poder comunicarse y deben elegir un nombre de red y canal común.

Modo Monitor: Este modo se usa para monitorear pasivamente todo el tráfico de un canal dado. Y este es útil para:

- Analizar los problemas en un enlace inalámbrico.
- Observar en una zona el uso de un espectro.
- Realizar tareas de seguridad y mantenimiento.

En el siguiente diagrama observaremos los modos de funcionamiento del WiFi:

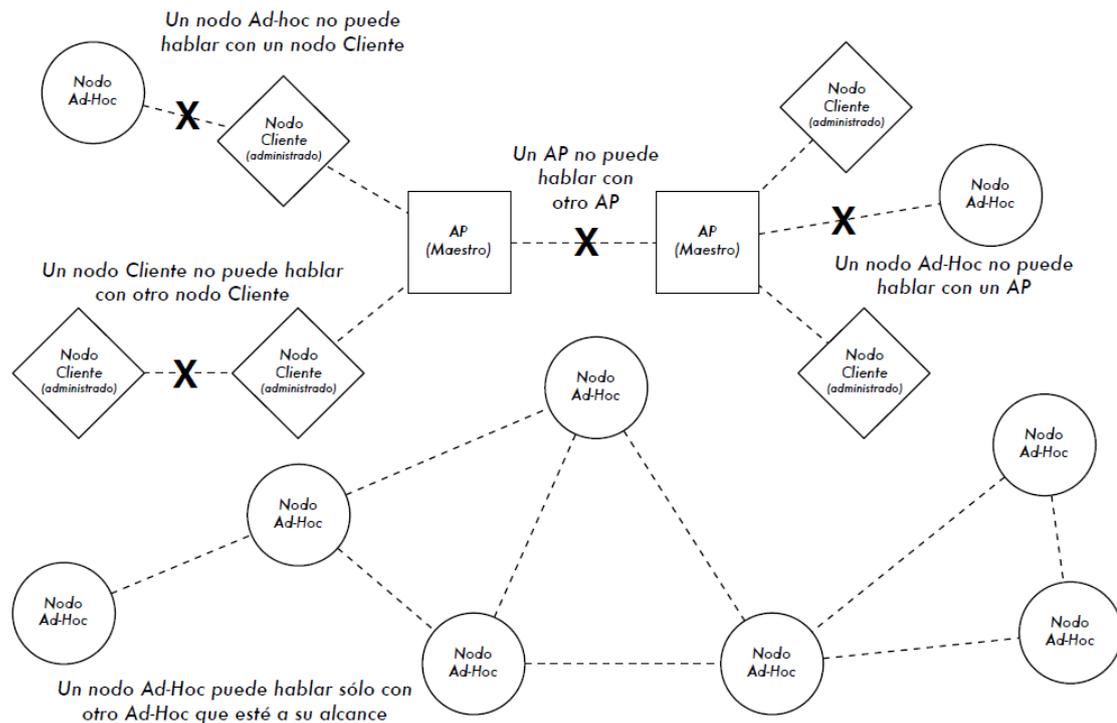


Figura 11: Grafico que muestra los radios de acción en los distintos modos WiFi

Fuente: Ermanno, 2010, pág. 21

2.3 El Lenguaje HTML (Hyper Text Markup Language)

Lenguaje de Marcas de Hipertexto, es un lenguaje que se utiliza para la creación de páginas en la WWW (Word Wide Web).

Las páginas web pueden ser vistas por el usuario mediante un tipo de aplicación llamada navegador. Podemos decir por lo tanto que el HTML es el lenguaje usado por los navegadores para mostrar las páginas webs al usuario.

Este lenguaje nos permite unir textos, sonidos e imágenes y combinarlos a nuestro gusto. Además el HTML nos permite la introducción de referencias a otras páginas por medio de los enlaces hipertexto.

El HTML se creó en un principio con objetivos divulgativos. El HTML se creó sin dar respuesta a todos los posibles usos que se le iba a dar y a todos los usuarios que lo utilizarían en un futuro. Sin embargo se han ido incorporando modificaciones con el tiempo, a estas se les llamo estándares del HTML.

Un documento HTML es un archivo de texto simple, por lo que podemos elaborarlo con editor de texto sencillo, como un bloc de notas de Windows. HTML no distingue entre mayúsculas y minúsculas en la especificación de marcas y sus atributos.

Los visualizadores de texto no toman en cuenta las tabulaciones, los espacios en blanco ni los saltos en línea; por lo que su principal ventaja es que nos permite obtener resultados uniformes y con buena presentación de manera bastante fácil.

En HTML hay caracteres especiales por que juegan un papel importante en al funcionamiento del mismo, como los símbolos mayor y menor que (<, >).

Sin embargo existen otros símbolos que no pueden escribirse directamente, sino que deben remplazarse por una cadena de caracteres que el visualizador interpretara de manera correcta; esta cadena de caracteres comienza con el símbolo (&) seguido de una combinación de caracteres alfanuméricos que tienen un significado especial en HTML, en la siguiente tabla se muestra algunos de estos caracteres:

Tabla 4: En esta tabla se muestra como escribir algunos caracteres

Carácter especial	Transcripción HTML	Comentario
Los acentos	&<vocal a acentuar> acute;	En la actualidad la mayoría de los browsers los soportan.
Símbolo menor que (<)	<	
Símbolo mayor que (>)	>	
La eñe	ñ	En la actualidad la mayoría de los browsers la soportan
Espacios en blanco	 (non breaking space)	HTML solo reconoce un espacio en blanco entre palabras.

Fuente: ATICA, pág. 13

Un documento HTML, es un texto definido entre las marcas:

```
<html></html>
```

Un documento HTML siempre se compone de 2 partes siguientes:

```

<html>

    <head> Cabecera del documento

</head>

    <body> Contenido del documento

</body>

</html>

```

2.3.1 Cabecera

Dentro de la cabecera hay información del documento, que no se ve en la pantalla principal, y que precisa las características del documento, principalmente el título del documento.

2.3.2 Cuerpo

Es comando acepta numerosos modificadores. Dentro del cuerpo se incluye cualquier etiqueta imprimible.

Hay muchas otras etiquetas pero su descripción no es necesaria en este informe de proyecto de tesis; por lo que daremos a continuación el código HTML de la página web para la visualización de las características de funcionamiento de la estufa de cultivo multi-uso.

Tabla 5: A continuación se muestra el programa HTML de la página web

```

<html>
<head></head>
<body text="blue">
<font face="Arial">
<table width="70%" height="25%" bgcolor="50cf00" border="10" bordercolor="009f00"
cellspacing="10" cellpadding="10">
<tr align="left" valign="top" bordercolor="50cf00">
<td width="30%">
<font face="bauhaus 93" size="6" color="007f00">
<p>ESTUFA</p>
<p>MULTIUSO</p>
</font>
</td>
<td width="30%" bgcolor="A8f000">

```

```

<h2>
<p>Temperatura:</p>
<p>&deg;C 00.0</p>
<p>Hora: 00:00</p>
<p>Modo: Estufa</p>
</td>
</h2>
<td width="40%" bgcolor="f0f800">
<h3>
<p>1CAMBIO: 01:00 &deg;C 40 10</p>
<p>2CAMBIO: 02:00 &deg;C 50 10</p>
<p>3CAMBIO: 03:00 &deg;C 60 10</p>
<p>FINAL: 04:00</p>
</h3>
</td>
</tr>
</table>
</font>
</body>
</html>

```

Elaboración: Propia

2.4 Circuito Optoacoplador

Para ello primero definiremos lo que es un optoacoplador, el cual es un componente electrónico que se utiliza como transmisor y receptor óptico (de luz), es decir puede transmitir de un punto a otro una señal eléctrica sin necesidad de conexión física ni cables (por el aire), mediante una señal luminosa. Por eso también se llaman OptoInterruptor.

Activamos una luz y esta luz llega a un detector que genera una tensión de salida, interruptor cerrado. Si no se activa la luz o no le llega la luz al detector, este no genera ninguna tensión de salida, es decir interruptor abierto.

A continuación se mostrara un circuito electrónico que se llama interface de potencia, basado en un optoacoplador DIODO - TRIAC capaz de manejar cargas del orden de 10

amperios o más dependiendo la capacidad del triac a utilizar a la salida.

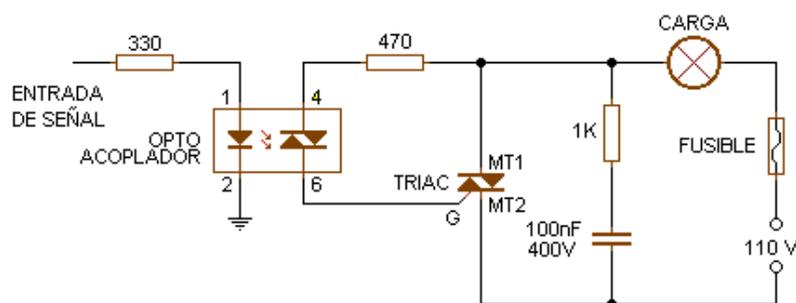


Figura 12: Diagrama que muestra el circuito Optoacoplador para el manejo de la resistencia calefactora de la estufa de cultivo

Fuente: TECNOLOGIA

2.5 HIPOTESIS

2.5.1 Hipótesis General:

Elaborar una estufa multiuso que pueda servir para el cultivo de diferentes organismos.

2.5.2 Hipótesis Específicas:

- Lograr una configuración más sencilla de las características de trabajo de una estufa.
- Mejorar el desempeño de una estufa aumentándole la capacidad de controlar la razón de cambio de temperatura versus tiempo; que le permitirá el manejo de cultivos que tienen parámetros más estrictos.
- Esperar superar las alarmas usadas para fallas eléctricas, mediante el uso de la comunicación inalámbrica WiFi y una página Web.

CAPITULO III

MATERIALES Y MÉTODOS

3.1 Tipo de investigación del problema

El método a utilizar es el Método Sistémico, que consiste en modelar el objeto mediante la determinación de sus componentes, así como las relaciones entre ellos. Esas relaciones determinan por un lado la estructura del objeto y por otro su dinámica.

3.2 Población y muestra de la investigación

La población de los elementos de esta investigación son las estufas de cultivo, o de crianza (incubadoras). También podríamos incluir los distintos tipos de controladores de temperatura para estufas.

La muestra tomada sería los controladores de temperatura óptimos.

3.3 Técnicas de tratamiento de Datos

Para esta investigación se utilizó programas con buena capacidad de simulación y toma de datos, como es el Matlab, también se utilizó lenguajes como el C para el uso de

Microcontroladores u otros tipos de lenguajes de bajo nivel, los cuales servirán para la toma de datos.

3.4 Arduino UNO

3.4.1 HARDWARE

El Arduino UNO es una placa electrónica basada en el ATmega328 (ficha técnica). cuenta con 14 entradas/salidas digitales de las cuales 6 pines se pueden usar como salidas pwm, 6 entradas analógicas, el oscilador de cristal es de 16MHz, una conexión USB, un conector de alimentación, una cabecera de ICSP, y un botón de reinicio.

El hardware del arduino Uno tiene las siguientes partes:

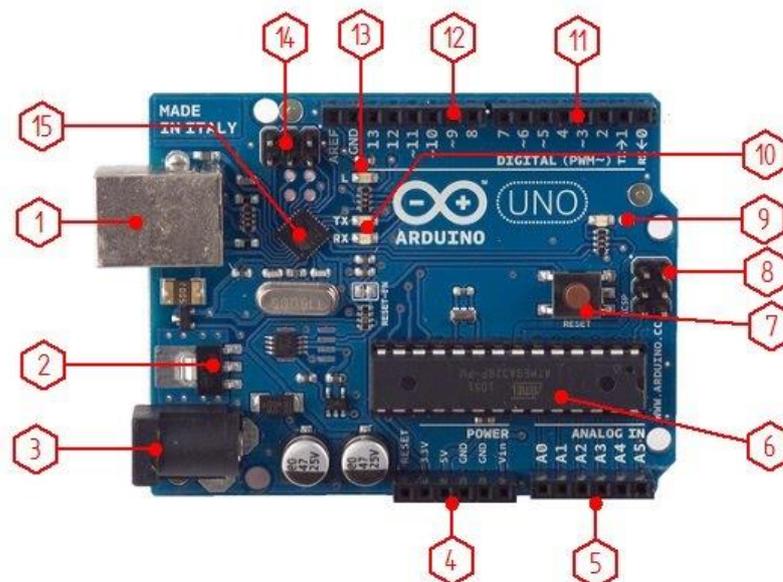


Figura 13: Figura que nos señala las partes de un Arduino UNO

Fuente: Profeyony

1. Conector USB: que puede ser tipo B o mini, este permite la comunicación para la programación y la toma de datos, también da una fuente de 5V para alimentar al arduino, pero de baja corriente por lo que no sirve para alimentar motores grandes por ejemplo.
2. Regulador de voltaje de 5V: se encarga de convertir el voltaje ingresado por el plug, en un voltaje de 5V regulado. Necesario para el funcionamiento de la placa y para alimentar circuitos externos.

3. Plug de conexión para fuente de alimentación externa: el voltaje que se suministra por aquí debe ser directo y estar entre 7V y 12V, generalmente se usa un adaptador, pero debe tener cuidado de que el terminal del centro del plug quede conectado a positivo ya que algunos adaptadores traen la opción de intercambiar la polaridad de los cables.
4. Puerto de conexiones: constituido por 6 pines de conexión con las siguientes funciones: RESET, permite resetear el microcontrolador al enviarle un cero lógico. Pin 3.3V, este pin provee una fuente de 3.3VDC para conectar dispositivos externos como en la protoboard por ejemplo. Pin 5V, es una fuente de 5VDC para conectar dispositivos externos. Dos pines GND, que proveen la salida de cero voltios para dispositivos externos. Pin Vin, este pin está conectado con el positivo del plug 3 por lo que se usa para conectar la alimentación de la placa con una fuente externa de entre 9V y 12V.
5. Puerto de entradas análogas, aquí se conectan las salidas de los sensores analógicos. Estos pines solo funcionan como entradas recibiendo voltajes entre cero y cinco voltios directos.
6. Microcontrolador Atmega 328, es el microcontrolador implementado en los Arduino Uno y sobre el cual vamos a programar, en la versión SMD del arduino uno R2, se usa el mismo microcontrolador pero en montaje superficial, en este caso las únicas ventajas que se me ocurren son la reducción del peso y ganar un poco de espacio.
7. Botón de RESET: Nos sirve para resetear el modulo.
8. Pines de programación ICSP, son usados para programar microcontroladores en protoboard o sobre circuitos impresos sin tener que retirarlos de su sitio.
9. LED de encendido: Se prende cuando el Arduino está encendido.

10. LEDs de recepción y transmisión, estos se encienden cuando hay comunicación serial con la PC. El Tx indica transmisión de datos y el Rx recepción.
11. Encontramos 8 pines digitales: El pin 0(RX) y 1(TX) se utiliza para recibir (RX) y transmitir (TX) datos serie TTL, estos se encuentran conectados al puerto USB. Los pines 2 y 3 pueden ser utilizados para activar una interrupción. Los pines 3, 5, 6 proporcionan salida PWM.
12. Encontramos 8 pines digitales: Los pines 3, 5, 6 proporcionan salida PWM (Modulation for Wide of Pulse). Los pines 10(SS), 11(MOSE), 12(MISO), 13(SCK) nos pueden servir para comunicación SPI.
13. LED 13: Hay un LED conectado al pin 13 para comprobar si este está en estado alto o bajo.
14. De función similar descrita en el punto 8.
15. Chip de comunicación que permite la conversión de serial a USB.

3.4.2 SOFTWARE

Programación: La placa Arduino se puede programar a través del IDE (Entorno de Desarrollo Integrado) de Arduino. Primero hay que decirle al IDE nuestro el tipo de placa. Para ello, selecciona bien la targeta (Arduino UNO en este caso). Después hay que decirle al IDE en qué puerto USB hemos conectado la placa, esto lo haremos a través del menú Herramientas seleccionando Puerto Serial.

Entorno de desarrollo para el Arduino: Este entorno permite la conexión del ordenador con la placa para cargar los programas y comunicarse con ella.

El programa o “sketch” se escribe en el editor de texto (se puede cortar, copiar, pegar, etc., como en los editores de texto habituales). La consola inferior muestra el texto de salida para el entorno de Arduino incluyendo los mensajes de error completos y otras informaciones.

Existen cinco menús; Archivo, Editar, Sketch, Herramientas y Ayuda. Los menús son sensibles al contexto, lo que significa que estarán disponibles sólo los elementos relevantes para la tarea que estemos realizando.

Entorno de programación de Arduino: Contiene las siguientes partes principales.

Comentarios: “Las primeras líneas del sketch, las cuales se encuentran entre /* y */ , son comentarios y son ignorados por el compilador de Arduino. Aparecen en gris” (Catedu, 2014, pág. 7)

Otra forma de incluir comentarios cortos es mediante el //. Todo lo que sigue a la doble barra hasta el final de línea se considera comentario.

Variables: Como todas las variables en C posee un nombre, un tipo y un valor. Por ejemplo:

```
boolean ledPin = 1;
```

La cual está declarando un variable con nombre ledPin, de tipo boolean (lógica, 1 o 0) a la cual se le ha asignado el valor 1.

Funciones: Una función es una porción de código que puede ser usado desde cualquier otro punto del sketch. La estructura básica del lenguaje de programación del Arduino está organizada al menos de 2 funciones, que encierran bloques de declaraciones o sentencias: void setup() y void loop(); ambas funciones deben aparecer en todo programa.

La función setup() se usa para inicializar variables, asignar modos (INPUT u OUTPUT) de los pines, iniciación de librerías, etc. Esta función solo se ejecutara una vez después que se alimente el arduino o se resetee el modulo.

La función loop() ejecuta un bloque de código de forma cíclica continuamente.

Por lo descrito anteriormente y por los motivos se enunciaran a continuación se escogió el Arduino Uno como el microcontrolador para este proyecto.

- Una ventaja a comparación de un microcontrolador como el PIC es su programación más sencilla, ya que no hay que configurar sus registros.
- Si lo comparamos con arduinos superiores u otro microcontrolador como el Rasperypi, no eran necesarios ya que las entradas analógicas y digitales del Arduino UNO son suficientes y seria costoso comprar los anteriores módulos nombrados.

3.5 Módulo ESP8266

EL ESP8266 es un chip que gracias a sus características y un gran precio ha logrado convertirse en uno de los adaptadores Wifi más populares. Antes de continuar con más descripción cabe mencionar que debemos distinguir entre los módulos y los microcontroladores, el ESP8266 es un microcontrolador con el cual podemos trabajar de manera independiente, mientras que el módulo WiFi de este microcontrolador nos permite un punto de acceso inalámbrico.

3.5.1 Especificaciones de este modulo

- Soporta el protocolo 802.11
- Capacidad mediante software para trabajar como cliente o servidor.
- Soporta IPv4 y los protocolos TCP/IP Y UDP
- Tiene un núcleo Diamond Standard Core (LX3) hecho por Tensilica
- Su núcleo de arquitectura RISC 32bits corre a 80Mhz
- 64KBytes de RAM de instrucciones
- 96KBytes de RAM de datos
- Tiene puertos GPIO, I2C, ADC, SPI, PWM.
- Los módulos cuentan con una memoria flash SPI Winbond W25Q40BVNIG
- Los ESP8266 son fabricados por Espressif
- Los módulos que usan este chip son de varios fabricantes y vienen en diferentes formatos.

3.5.2 Características eléctricas

- Voltaje de trabajo entre 3V y 3,6V.
- Por información de CEO de Espressif, admite tensiones de entrada de 5V e los puertos GPIO.

3.5.3 Firmware para el ESP8266

“El ESP-01 viene inicialmente con el “firmware AT”, con el cual puedes usarlo como un adaptador WiFi-serial, usando un microcontrolador externo como un Arduino para controlarlo y comunicarse con la red inalámbrica.” (Naylamp)

La placa NodeMCU del ESP8266 nos permite cambiar el firmware del ESP8266 y usar lenguajes como el LUA, Python, Basic o JavaScript.

3.5.4 Módulos ESP-XX

El fabricante AI-Thinker proporciona la serie ESP con diferentes modelos para diferentes usos, de los cuales describiremos los siguientes:

ESP-01: Este módulo tiene disponible dos pines GPIO digitales para controlar sensores y actuadores, también se puede llegar a utilizar para este uso los pines Rx y Tx si no se utilizan para la comunicación a través del puerto serie. Se puede programar a través de un adaptador serie/USB o a través de Arduino.

ESP-05: Es el módulo más simple de los que describiremos. La disposición de los pines nos permite un fácil conexionado con la protoboard, pero no dispone de ningún puerto GPIO accesible.

ESP-12: En este módulo tenemos acceso a 11 puertos GPIO, de los cuales uno es analógico con una resolución de 10-bit (1024 valores posibles). Si le soldamos los pines o compramos un adaptador podremos usarlo con un protoboard, por los que quizá esta sea la mejor opción si queremos hacerlo funcionar de forma autónoma.

NodeMCU: Este módulo incluye un adaptador serie/USB y se alimenta a través del microusb. Y lo más novedoso de este módulo es que puedes descargar un firmware que te permite programar en lenguajes como LUA, Python, Basic o JavaScript.

En este proyecto se utilizó el ESP-01 ya que en este proyecto solo requerimos su uso como un dispositivo de complemento para la comunicación inalámbrica, los otros módulos citados anteriormente como el ESP-05 es muy simple y poco comercial, el ESP-12 tiene más cualidades ya mencionadas anteriormente, las cuales no son necesarias para este proyecto. Por lo que a continuación señalaremos los pines de este módulo:

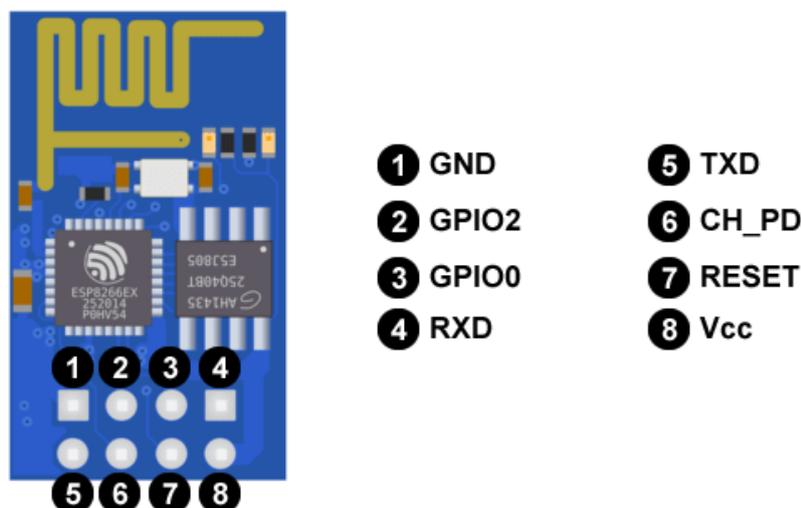


Figura 14: En esta figura se señala las partes de módulo ESP-01

Fuente: del Valle Hernández

3.6 PIC16F84

Es un microcontrolador con memoria de programa tipo FLASH, lo que permite programarlo sin borrado previo.

3.6.1 Puertos de entrada/salida

Estos están compuestos por líneas digitales de entrada/salida que trabajan entre 0 y 5V.

Los puertos se pueden configurar como entradas para recibir datos o como salidas para manipular dispositivos externos.

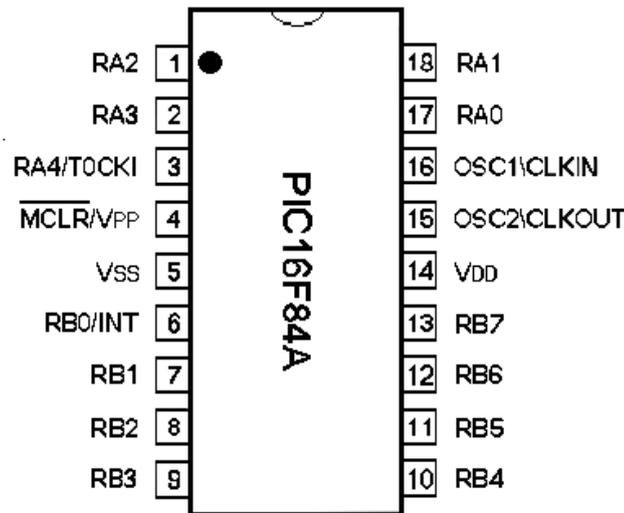


Figura 15: En esta figura se indican los nombres y funciones de los pines de PIC16F84A

Fuente: Palacios, Remiro, & López, 2004, pág. 2

Este PIC tiene 2 puertos, tal como se ilustra en la Figura14:

- El Puerto A con 5 líneas, pines RA0 a RA4.
- El Puerto B con 8 líneas, pines RB0 a RB7.

Cada línea puede ser configurada de manera independiente unas de otras. Las líneas son capaces de entregar niveles TTL cuando la tensión de alimentación aplicada es VDD es 5V. La máxima corriente depende cuál de los siguientes estados se encuentra:

Cuando el pin está a nivel bajo: Consume una corriente de 25 mA. Sin embargo las 5 líneas del Puerto A no puede exceder de 50 mA, ni la suma de corriente de las 8 líneas del Puerto B puede exceder de 100 mA.

3.6.2 Oscilador

“Todo microcontrolador requiere de un circuito que le indique la velocidad de trabajo, es el llamado oscilador o reloj. Este genera una onda cuadrada de alta frecuencia que se utiliza como señal para sincronizar todas las operaciones del sistema. Este circuito es muy simple pero de vital importancia para el buen funcionamiento del sistema. Generalmente todos los componentes del reloj se encuentran integrados en el propio microcontrolador y tan solo se requieren unos pocos componentes externos, como un

cristal de cuarzo una red RC, para definir la frecuencia de trabajo.” (Palacios, Remiro, & López, 2004, pág. 27)

En este PIC los pines OSC/CLKIN y OSC/CLKOUT son las líneas utilizadas para este fin, el cual permite cinco tipos de osciladores para definir la frecuencia de funcionamiento:

- XT: Cristal de cuarzo.
- RC: Oscilador con resistencia y condensador.
- HS: Cristal de alta velocidad.
- LP: Cristal para baja frecuencia y bajo consumo de potencia.
- Externa: Cuando se aplica una señal de reloj externa.

3.6.3 Reset

El reset es un microcontrolador que provoca la reiniciación de funcionamiento. En este estado, la mayoría de componentes internos del microcontrolador toman un estado inicial.

Los microcontroladores como el PIC16F84 tienen un pin para este fin, el cual se llama MCLR (Master Clear). El fabricante recomienda conectar en serie con el pulsador una resistencia de 50 a 100 ohm.

Este PIC también permite el llamado Power-On Reset, que proporciona un reset al microcontrolador en el momento de conectar la fuente de alimentación. El PIC tiene un temporizador llamado Reset PWRT, que proporciona un retardo de 72 ms desde el instante de la conexión a la alimentación, un reset se mantiene durante ese tiempo, garantizando que VCC alcance un nivel aceptable de tensión para un arranque correcto del sistema. Para utilizar este tipo de reset hay que conectar el pin MCLR al positivo de alimentación. Más información de este microcontrolador podemos encontrarlo en su manual datasheet.

Para este proyecto fue necesario tener una señal de reloj constante, la cual no puede ser proporcionada por el arduino, ya que este tiene una librería milis que falla cuando utilizamos interrupciones, por lo que se requirió de una interrupción externa, la cual fue proporcionada por el PIC16F84 descrito anteriormente, el cual es uno de los más sencillos de su clase.

3.7 Driver L298N

Este módulo basado en el chip L298N te permite controlar dos motores de corriente continua o un motor paso a paso bipolar de hasta 2 amperios. Este módulo cuenta con diodos de protección y un LM7805 que suministra 5V a la parte lógica del integrado L298N, cuenta con jumpers para habilitar cada una de sus salidas (A y B); la salida A está conformada por OUT1 y OUT2, y la salida B por OUT3 y OUT4.

Los pines de habilitación son ENA y ENB respectivamente, En la parte inferior están los pines de control del módulo, marcados como IN1, IN2, IN3 e IN4.

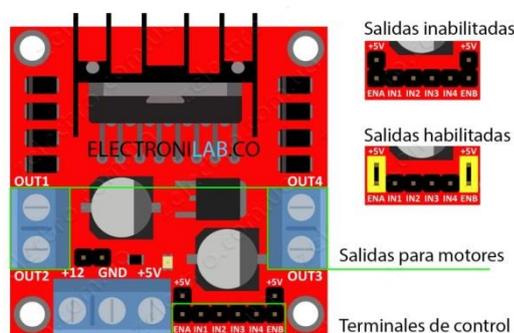


Figura 16: Figura donde se muestran los puertos de este proyecto

Fuente: Zuffli, 2014

3.7.1 Funcionamiento

Este módulo puede alimentarse de 2 maneras gracias al regulador de voltaje LM7805, estas dos formas las describimos a continuación:

Cuando el jumper de selección de 5V se encuentra activo: El modulo nos da una alimentación de 6 a 12V DC, como el regulador se encuentra activo el pin marcado como +5V tendrá un voltaje de 5V.

Cuando el jumper de selección de 5V se encuentra inactivo: El modulo nos da una alimentación de 12 a 35V DC, como el regulador no está activo, se tiene que conectar el pin marcado como +5V a VCC (5V DC) para alimentar el L298N.

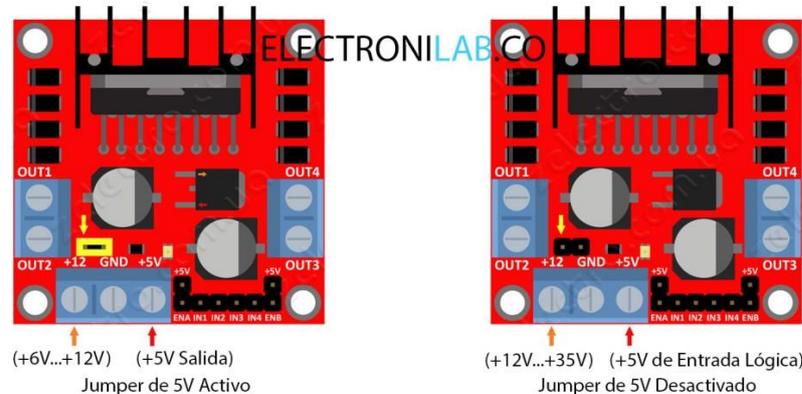


Figura 17: Figura donde se ilustra las conexiones necesarias para cambiar los voltajes en las salidas

Fuente: Zuffli, 2014

En este proyecto se utilizó este módulo por cuestiones de tiempo y simplificación del trabajo, ya que elaborar nuestro propio circuito actuador para el control de un motor de hubiese sido algo tedioso, por los que decidí elegir este módulo por sus buenas características mencionadas anteriormente.

3.8 Sensor de temperatura por resistencia PT100

Este sensor consiste en un alambre de platino que a 0 °C tiene 100 ohm y que al aumentar aumenta su resistencia eléctrica, el incremento de la temperatura no es lineal pero si creciente, como se muestra en la siguiente tabla:

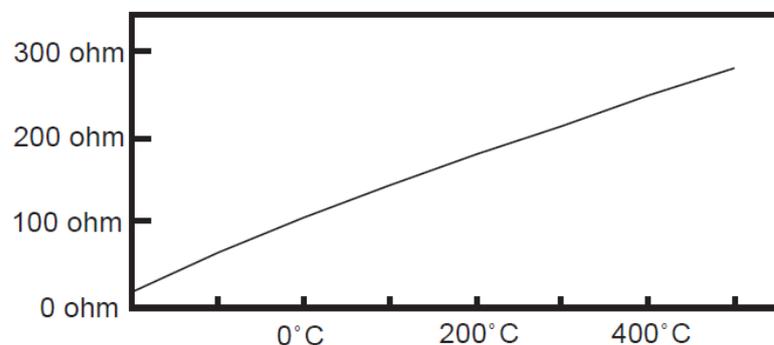


Figura 18: En esta grafico se muestra la razón de cambio de la resistencia eléctrica versus la temperatura de un PT100

Fuente: Arian, pág. 1

Normalmente los PT100 industriales se consiguen encapsulados en la misma forma que las termocuplas, es decir dentro de un tubo de acero inoxidable u otro material (vainas), en un extremo está el elemento sensible (alambre de platino) y en el otro está el terminal eléctrico de los cables protegido dentro de una caja redonda de aluminio (cabezal) (Arian, pág. 1)

Este sensor puede fácilmente entregar precisiones de una décima de grado, tiene la cualidad de abrirse antes de deteriorarse, por lo que el dispositivo medidor detecta inmediatamente la falla del sensor.

Cuando lo conectamos con 2 hilos se originará un error previsible, dado por la lectura de los cables R_{c1} y R_{c2} que unen el PT100 al instrumento, por lo que se podría minimizar con el uso de cables gruesos que den menor resistencia. En la siguiente grafica se muestra lo explicado:

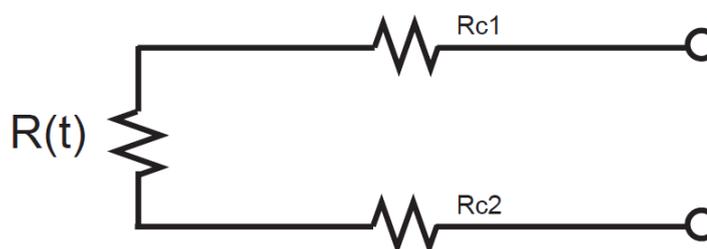


Figura 19: Figura donde se muestra la conexión de 2 hilos donde se muestran sus resistencias R_{c1} y R_{c2} como la del sensor $R(t)$

Fuente: Arian, pág. 2

El modo de conexión de 3 hilos resuelve bastante bien el problema de error generado por los cables, el modo consiste en pasar una corriente conocida por los cables azul y verde con el cual el instrumento mide $2R_c$. Luego medimos la resistencia por los cables café y azul para finalmente restarle $2R_c$ y obtener la resistencia $R(t)$. Lo explicado se ilustra a continuación:

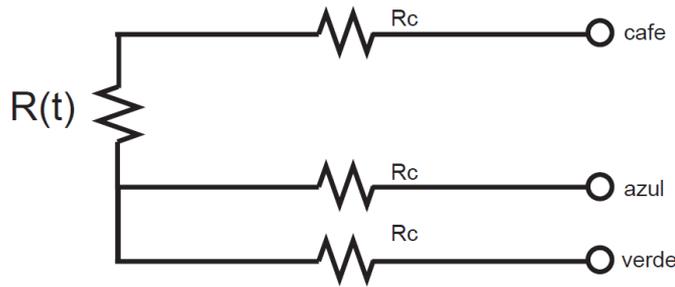


Figura 20: Modo de conexión de 3 hilos con sus respectivas conexiones Rc

Fuente: Arian, pág. 3

El modo de los cuatro hilos es el más preciso de todos, pero es más costoso. Por los cables 1 y 4 de la figura siguiente se hace circular una corriente conocida I , provocando una resistencia conocida en los extremos de $R(t)$.

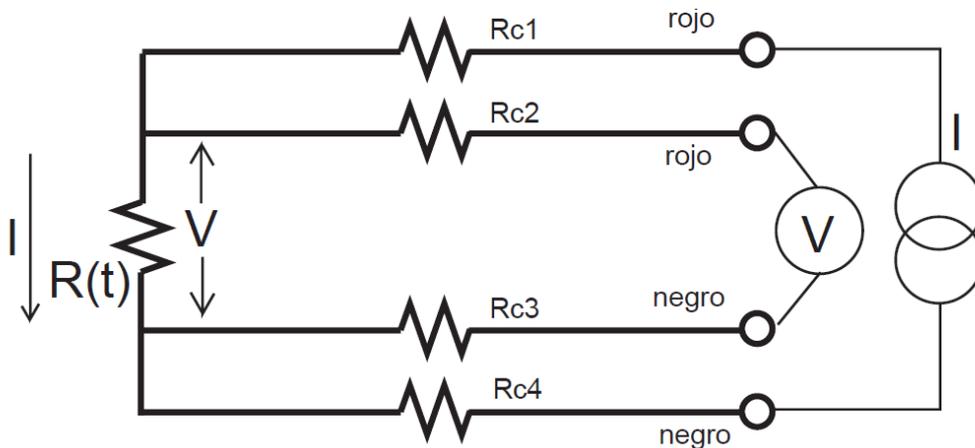


Figura 21: Modo de conexión de 4 hilos con su respectivo voltímetro y amperímetro para calcular la resistencia $R(t)$

Fuente: Arian, pág. 3

Los cables 2 y 3 están conectados a la entrada de un voltímetro de alta impedancia; como R_{c2} y R_{c3} están sin corriente, el voltímetro medirá exactamente el voltaje de los extremos de $R(t)$, por lo que se obtendrá el valor de $R(t)$ al dividir V/I .

Un problema que se le podría originar a este método es por el efecto Joule, el cual puede generar un calentamiento y producir un error en su lectura. Este problema depende del tamaño del sensor y de la conducción de calor del medio.

Escogí este sensor para este trabajo por que tolera un rango más amplio de temperaturas a medir que otros sensores y además es el más utilizado por sus cualidades en lo que se refiere a estufas de cultivo o esterilizadores.

3.9 MOC3041

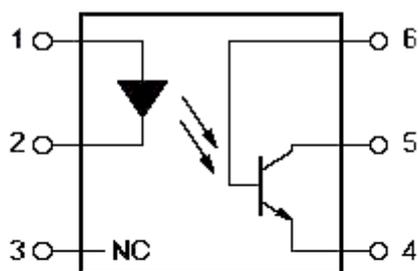


Figura 22: En esta imagen se muestra que una ilustración de MOC3041 de Fairchild
Fuente: COMPIC

Este es un optoacoplador, estos también son llamados optoaislador o aislador acoplado ópticamente, es un dispositivo de emisión y recepción que funciona como un interruptor activado mediante la luz emitida por un diodo LED que satura un componente optoelectrónico, normalmente en forma de fototransistor o fototriac. De este modo se combinan en un solo dispositivo semiconductor, un fotoemisor y un fotoreceptor cuya conexión entre ambos es óptica. Estos elementos se encuentran dentro de un encapsulado que por lo general es del tipo DIP. Se suelen utilizar para aislar eléctricamente a dispositivos muy sensibles.

El MOC3041 de Fairchild es un optoacoplador de salida de conductor triac de paso a través en un paquete DIP de 6 patillas. Este dispositivo contiene un diodo emisor de infrarrojos de GaAs acoplado ópticamente a un detector de silicio monolítico que realiza la función de un conductor triac bilateral de cruce de voltaje cero. Este dispositivo está diseñado para conectar el sistema de lógica a equipos alimentados a partir de líneas de 115VAC, teletipos, CRTs, relés de estado sólido, controles industriales, impresoras, motores, solenoide y electrodomésticos.

3.10 Optoacoplador 4N25



4N25 Opto-acoplador

- PIN 1. ANODE
- 2. CATHODE
- 3. NO CONNECTION
- 4. EMITTER
- 5. COLLECTOR
- 6. BASE

Figura 23: Figura que ilustra las funciones de los pines de este Optoacoplador

Fuente: TBEM

El 4N25 es un diodo emisor de infrarrojos de arseniuro de galio, junto con un fototransistor de silicio. Un opto acoplador, también llamado optoaislador o aislador acoplado ópticamente, es un dispositivo de emisión y recepción que funciona como un interruptor activado mediante la luz emitida por un diodo LED que satura un componente opto electrónico, normalmente en forma de fototransistor o fototriac. De este modo se combinan en un solo dispositivo semiconductor, un fotoemisor y un fotoreceptor cuya conexión entre ambos es óptica. Estos elementos se encuentran dentro de un encapsulado que por lo general es del tipo DIP. Se suelen utilizar para aislar eléctricamente a dispositivos muy sensibles.

CAPITULO IV

RESULTADOS Y DISCUSIÓN

Como resultado del objetivo general, que era el diseño de un control de temperatura de una estufa, en esta parte realicé el diseño de un controlador Optimo Cuadrático Estacionario, el cual se obtuvo del modelado matemático descrito en la sección de Revisión de literatura, teniéndose las siguientes constantes de estado y control:

$$A = -1/3000$$

$$B = 1/3000$$

$$E = 1/1500$$

$$C = 1$$

$$D = 0$$

Por otro lado tenemos la interferencia producida por el efecto de la temperatura ambiente, el cual tiene una matriz en tiempo continuo que denomine E y que en tiempo discreto se llama F , en la simulación se considera la temperatura ambiente como un

valor constante, como el proceso es univariable las matrices G y H tienen los siguientes valores:

$$G = 1 \quad H = 2 \times 10^{-5} \quad F = 4 \times 10^{-5} \quad C = 1 \quad D = 0$$

Después le añadí el efecto producido por el regulador Optimo Proporcional Integral para tener un error nulo en estado estacionario, por lo que se tiene las siguientes matrices:

$$G1 = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}$$

$$H1 = \begin{bmatrix} 2 \times 10^{-5} \\ -2 \times 10^{-5} \end{bmatrix}$$

$$M = [H1 \quad G1 * H1]$$

A estas matrices se les comprobó su controlabilidad, la cual consiste en hallar el rango de la matriz M , la cual coincidió con el orden del proceso, que en este caso es 2.

Después de esto halle la matriz de ganancia del controlador, la cual se obtiene estimando las matrices de ponderación R (para la señal de control $u(k)$) y Q que pondera el vector de estado $x(k)$. Los valores obtenidos fueron los siguientes:

Tabla 6: Aquí se muestran los diferentes valores del sobrepaso máximo respecto a el cambio de proporción entre $Q(1,1)$ y $Q(2,2)$.

Q(1,1)	Q(2,2)	R	Tiempo Pico en segundos	Sobrepaso Máximo
200	30	0.3	18.98	1.043
200	3	0.3	33.07	1.043
200	0.3	0.3	33.07	1.043
200	0.03	0.3	105.7	1.040
200	0.003	0.3	193.1	1.035
200	0.0003	0.3	369.6	1.020

Elaboración: Propia

Nos damos cuenta que a menor de $Q(2,2)$ va creciendo el valor del tiempo pico, pero disminuye muy poco el sobrepaso máximo, por lo que este cuadro solo nos muestra un comportamiento del proceso. Se escogió como razón de incremento y decremento el factor 10 para hacer más sencilla la busca de un valor óptimo. Nos damos cuenta que no

se obtiene una disminución significativa del sobrepaso máximo por lo que también probé con disminuir el valor de R a partir de un valor $Q(2,2)=0.003$, obteniendo los siguientes resultados:

Tabla 7: Aquí se muestran los valores del sobrepaso máximo variando $Q(1,1)$ y R .

$Q(1,1)$	$Q(2,2)$	R	Tiempo Pico en segundos	Sobrepaso Máximo
200	0.003	0.03	119.4	1.020
200	0.003	0.003	---	0
300	0.03	0.003	40.63	1.013
300	0.3	0.003	19.8	1.031
3000	0.3	0.003	---	0
1000	0.1	0.001	---	0

Elaboración: Propia

Estos resultados nos muestran que a $Q(1,1) \gg Q(2,2)$ nos encontramos con un sistema sin tiempo Pico, los que nos da un control más estable, por lo que el valor elegido fue el último de la tabla, que dio como resultado lo siguiente:

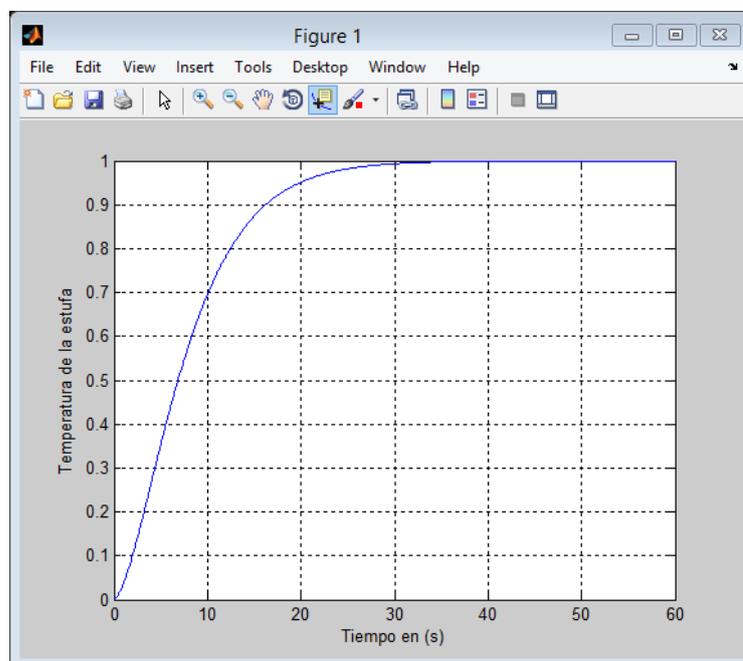


Figura 24: Figura donde se muestra la respuesta al escalón unitario en un simulador con 1000 muestras tomadas.

Elaboración: Propia

Posteriormente obtuve el valor de la matriz de ganancia KK , el cual se puede hallar de manera recursiva o con la ayuda del comando `dlqr` del simulador, obteniendo lo siguiente:

Para tener una simulación más cercana a la realidad agregamos el efecto de un observador, el cual se obtiene de la misma manera que de la ganancia del controlador, primero estimamos las matrices de ponderación Q_e y R_e , para luego hallar de manera recursiva o por ayuda de comandos el valor de K_e ; estas matrices fueron:

$$Q_e = [1000]$$

$$R_e = [10]$$

$$K_e = [0.9902]$$

Finalmente para efectos de simulación el valor del estado inicial fue 0, señal del bloque integrador $v=0$, entrada al sistema $r=0$ y temperatura ambiente (uI) igual a 20. De lo cual se obtuvo la siguiente gráfica:

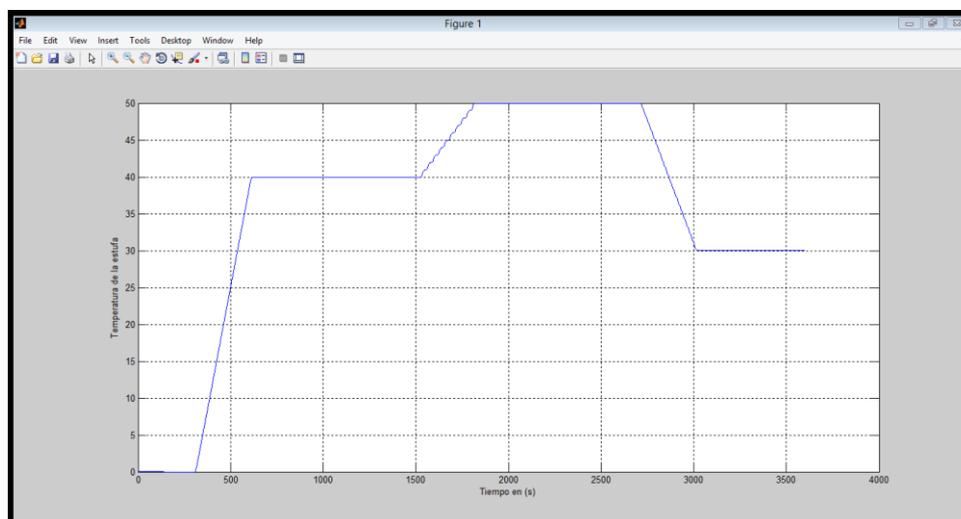


Figura 25: Figura donde se muestra las distintas pendientes de cambio de temperatura con razones de 8, 2 y -4 grados por minuto.

Elaboración: Propia

En la figura anterior podemos ver las diferentes pendientes de las rampas de cambio de temperatura. En este proyecto se definió una escala de 10 valores de razón de cambio de temperatura, esta se da en grados Celsius por minuto; el nivel uno es de $1^{\circ}\text{C}/\text{min}$ por lo

que el ultimo es $10^{\circ}\text{C}/\text{min}$ o 1°C en 6 segundos. La primera pendiente en la figura es de $8^{\circ}\text{C}/\text{min}$, la segunda es $2^{\circ}\text{C}/\text{min}$ y la ultima es $-4^{\circ}\text{C}/\text{min}$.

Las pendientes mostradas en esta simulación están conformadas por pequeñas respuestas al escalón unitario, sin embargo también conocemos que en los sistemas tanto de control PID (Proporcional Integrativo Diferencial) como Óptimo y otros existe la respuesta a la rampa, este tipo de entrada también podría ser usada para controlar la razón de cambio de temperatura, pero su desventaja radica en que su diseño es más complejo que el usado en la entrada del escalón y que es más susceptible a perturbaciones y fallas.

Por otro lado al usar una entrada escalón unitario de manera periódica hasta llegar al valor deseado, es más sencillo. En este proyecto al simular con método mencionado anteriormente se pudo obtener la siguiente pendiente para una razón de $8^{\circ}\text{C}/\text{min}$:

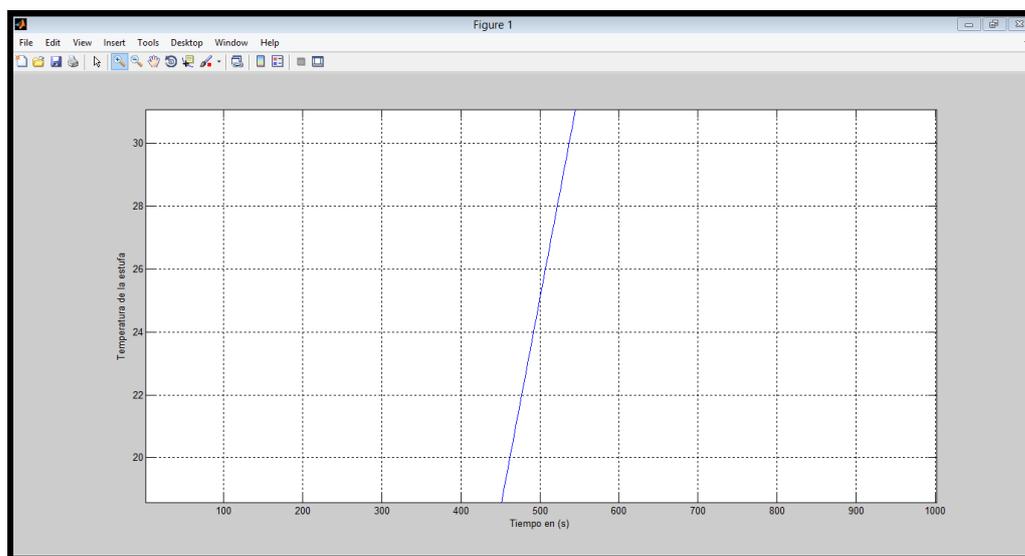


Figura 26: Figura donde se muestra la pendiente de cambio de temperatura de $8^{\circ}\text{C}/\text{min}$.

Elaboración: Propia

Como se puede observar la rampa no presenta rastros de los escalones unitarios y presenta una pendiente con una razón de 1°C por 7.5s. Se puede entender a partir de la Figura 23, que aproximadamente la señal llega a su estado estacionario a partir de los

35s; por lo que 27.5 segundos antes ya se dio la salida dada por el escalón contiguo, todo esto origina una señal de salida muy cercana a la recta.

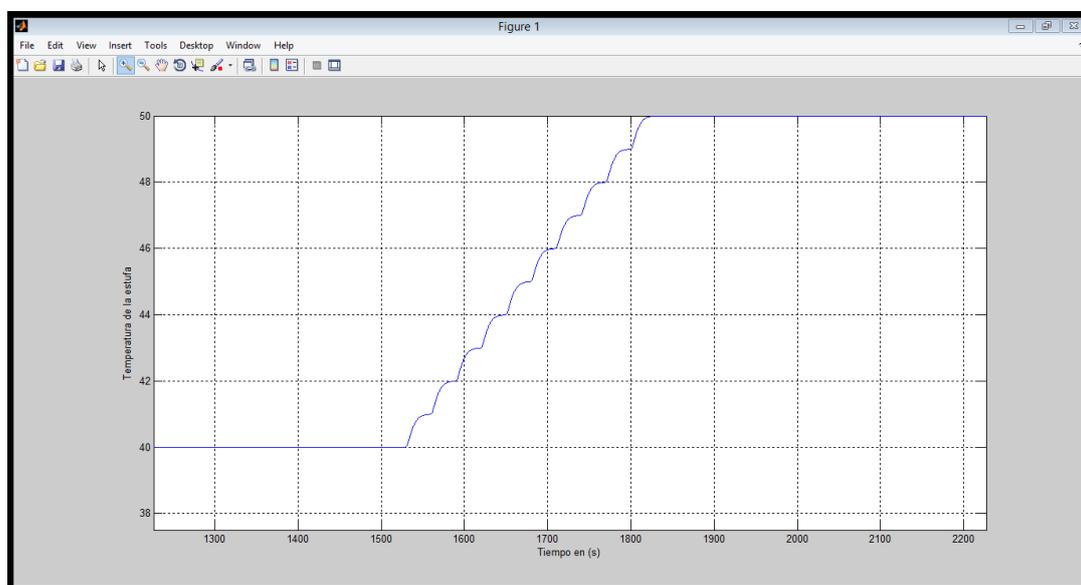


Figura 27: Figura donde se muestra la rampa escalonada de la razón de cambio de $2^{\circ}\text{C}/\text{min}$.

Elaboración: Propia

En la Figura 24 observamos una imagen más detallada de la segunda rampa, la cual debería tener una pendiente con una razón de 1°C por 30s; pero en lugar de ello nos encontramos con una señal escalonada, este problema se podría solucionar usando como señal de entrada escalones más pequeños, como de decimas de grado, disminuyendo así el error originado por las desviaciones alrededor de una rampa ideal, la cual para este caso debería cambiar en todos sus puntos a la razón anterior.

Otro resultado fue la realización de una página web para un monitoreo no presencial del funcionamiento de esta estufa de cultivo, como es el caso de otras estufas que utilizan displays como las que se muestra a continuación:



Figura 28: Figura que muestra una estufa de cultivo INCUDIGIT

Fuente: I.C.T, S.L.

La página web que realicé está conformada por una tabla de una fila y tres celdas, a la primera celda se le puso el color de la tabla, porque en este espacio puse el nombre de la estufa (estufa multiuso).

En la segunda celda se muestra parámetros no relacionados al funcionamiento del controlador de temperatura, como el modo de trabajo de la estufa (estufa o incubadora) y temperatura del interior de la estufa; la hora se da en formato de 24 horas, una imagen de este lo vemos a continuación:



Figura 29: Figura donde se muestra la segunda celda de la página web elaborada.

Elaboración: Propia

En el tercer espacio se da la información de los diferentes cambios de temperatura, los cuales se dan a horas determinadas, en cada cambio vemos que al final se muestra un número sin especificar su unidad, este número representa la razón de cambio de temperatura en grados por minuto, la siguiente figura nos muestra esta celda:

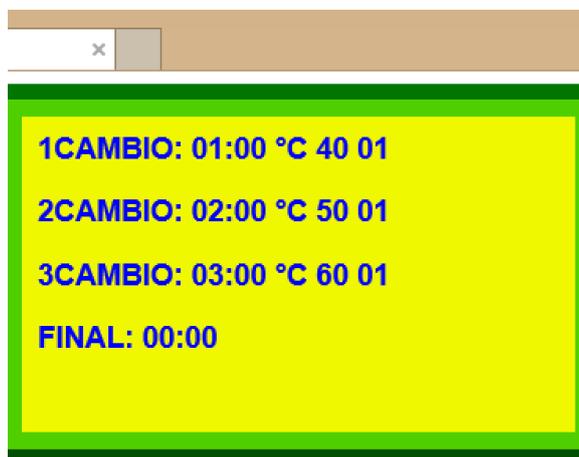


Figura 30: Figura donde se muestra la tercera celda de la página web elaborada.

Elaboración: Propia

El ESP necesita de comandos AT para su configuración, los cuales se dividen en diferentes grupos dependiendo de su función, entre los cuales tenemos:

- Comandos básicos de puesta.
- Funciones generales WiFi.
- Comandos AT relacionados a TCP/IP

Para poder visualizar esta página web, necesitamos configurarla mediante el monitor serial, por lo que desarrollare los siguientes pasos:

- Primero debemos comprobar el correcto funcionamiento del módulo, ingresando el comando básico “AT”, si nos responde OK significa que el modulo está listo para su funcionamiento.
- Después ingresamos el comando “AT+CWMODE”, al cual le concatenamos “=1”, el uno significa que el módulo estará en modo estación (cliente), si fuera dos estaría como AP.

- Luego tenemos que ver las redes disponibles para ver si tenemos la contraseña de alguna y poder conectarnos; para ello ingresamos el comando “AT+CWLAP” en el monitor serial.
- Después ingresamos el comando “AT+CWJAP=” más el nombre y contraseña de la red.
- Seguidamente iniciamos el modulo en modo multiconexion o conexión única, según nos convenga, para ello ingresamos el comando “AT+CIPMUX” seguido del igual y el numero escogido.
- Luego el asignamos el número de servicio y el puerto con el comando “AT+CIPSERVER”.
- Después de ello obtenemos la IP asignada al módulo ESP, usando el comando: “AT+CIFSR”.

Después de esto salimos del modo configuración utilizando el siguiente comando propio de este proyecto:

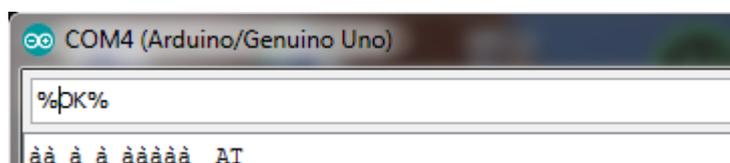


Figura 31: Aquí se muestra el comando utilizado para salir de la configuración del ESP.

Elaboración: Propia

Después de esto el ESP no atenderá comandos AT, ya se inhabilita el envío de comandos AT del Arduino al módulo; si quisiéramos volver a configurar el módulo necesitaremos resetear el Arduino.

Otra cosa que ocurrirá cuando enviemos el comando %OK% al módulo es que se habilitara el envío de los comandos necesarios para la configuración de los parámetros de funcionamiento de esta estufa. Seguidamente se muestra una imagen de la página web elaborada:



Figura 32: Aquí se muestra la página web elaborada que se mostrara al ingresar con su respectiva IP.

Elaboración: Propia

Como mencioné anteriormente se utilizaran unos comandos para configurar los parámetros de trabajo de la estufa multiuso. En general estos comandos están constituidos por las letras “CNG” seguido del carácter que pertenece al parámetro a configurar. La parte del programa encargado de distinguir estos comandos la podemos ver a continuación:

Tabla 8: A continuación se muestra la parte del programa encargado de distinguir estos comandos “CNG”.

```

if(C==1)
{
detachInterrupt(0);
if(Serial.available())
{
delay(30);//Tiempo para recibir una arreglo de datos de configuration de la computadora
while(i<3)
{
if(Serial.available())
{
d=Serial.read();
if(d==conf[i])
{
i++;
}
}
else
{
i=4;
while(Serial.available())//Esta parte del programa filtra datos que no lleven la cavecera "CNG"
{
c=Serial.read();
}
}
}
if(i==3)
{

```

```
d=Serial.read();
if('0'<d<'7')
{
i=4;
}
if(i==4)
{
if(d<'4')
{
j=0;
c=Serial.read();
while((j<4)&(c!='|'))
{
leer(d,j);//Se envia la fila y grupo al que pertenece el dato del arreglo leido por el Arduino
c=Serial.read();
j++;
}
}
else if(d<'6')
{
j=0;
c=Serial.read();
while((j<2)&(c!='|'))
{
leer(d,j);
c=Serial.read();
j++;
}
}
else
{
c=Serial.read();
if(c=='|')
{
d=Serial.read();
if(d=='E')
{
MODO[9]='E';
MODO[10]='s';
MODO[11]='t';
MODO[12]='u';
MODO[13]='f';
MODO[14]='a';
MODO[15]=' ';
MODO[16]=' ';
MODO[17]=' ';
MODO[18]=' ';
}
}
else if(d=='l')
```

```

{
  MODO[9]='l';
  MODO[10]='n';
  MODO[11]='c';
  MODO[12]='u';
  MODO[13]='b';
  MODO[14]='a';
  MODO[15]='d';
  MODO[16]='o';
  MODO[17]='r';
  MODO[18]='a';
}
}
}
}
}
}
}
i=0;
}
attachInterrupt(0,sensor,RISING);
}

```

Elaboración: Propia

Podemos darnos cuenta que en el arreglo `conf[]` se encuentra el comando “CNG”, el bucle `while` es iterado hasta llegar a comprobar el tercer carácter, si algún carácter ingresado es diferente, `i` será igual a 4 y los demás caracteres del buffer serán igualados a la variable `c`, la cual en este programa es la variable que se encarga de eliminar todos los datos innecesarios. Cuando `i` sea igual a 3 entonces se leerá el siguiente carácter y se comprobará que `d` sea mayor a ‘0’ y menor a ‘7’.

Para una mejor comprensión de lo dicho describiré a continuación estos comandos:

Cambios de funcionamiento: Para configurar los cambios utilizamos “CNG#|xx|xx|xx|xx|”, el valor de # es de 1 a 3, dependiendo del cambio utilizado, en el primer espacio escribimos la hora; en el segundo espacio ponemos el minuto; en el tercero la temperatura y finalmente en el cuarto la razón de cambio para llegar a esa temperatura. Los valores no pueden exceder de 2 dígitos y siempre antepondremos un 0 si este valor es menor a 10. Anteriormente se mencionó los topes de cada parámetro, por ejemplo la hora se pone de 0 a 24, si nos excederíamos el Arduino no reconocería este

parámetro por lo que no cambiaría ese lugar en la página web. En la siguiente figura veremos un ejemplo:



Figura 33: Aquí se muestra el comando usado para el primer cambio de temperatura.

Elaboración: Propia

Hora de finalización: Para configurar la finalización del proceso de control utilizaremos “CNG4|xx|xx|”, donde el primer espacio pertenece a la hora y el segundo a los minutos.

Configuración de la hora: Para configurar la hora que lleva esta estufa utilizaremos el siguiente comando “CNG5|xx|xx|”, donde el primer espacio pertenece a la hora y el segundo a los minutos. A continuación, mostramos un ejemplo:

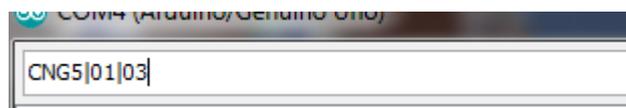


Figura 34: Se muestra el comando para cambiar la hora de la estufa multiuso.

Elaboración: Propia

Para el modo de trabajo: Para configurar este utilizaremos “CNG6|X|”, donde si ponemos “I” significa que pasaremos al modo incubadora y si ponemos “E” pasaremos al modo estufa de cultivo. La figura siguiente muestra un ejemplo:

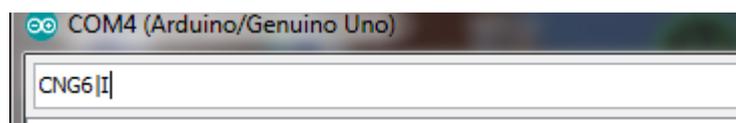


Figura 35: Se muestra el comando utilizado para pasar al modo incubadora.

Elaboración: Propia

Otra característica a aclarar es que el valor de la temperatura tomado por el puerto analógico digital del Arduino se actualizara en esta página web cada 3 segundos. Todo esto por el uso de la siguiente etiqueta:

```
<meta http-equiv="refresh" content="3">
```

4.1 Breve descripción del programa de la página web

Antes de enviar la página web el Arduino solicita la ID de la conexión, después envía “AT+CIPSEND=”, seguido del ID y la cantidad de bytes de la página web a enviar, que en este caso consta de 758.

Entonces el Arduino envía la siguiente información:

Tabla 9: A continuación se muestra como el Arduino envía la página web mostrada anteriormente.

```
html("<html>");
html("<head>");
html("<meta http-equiv=\"refresh\" content=\"3\">");
html("<head>");
html("<body text=\"blue\">");
html("<font face=\"Arial\">");
html("<table width=\"70%\" height=\"25%\" bgcolor=\"50cf00\" border=\"10\" bordercolor=\"009f00\"
                                     cellspacing=\"10\" cellpadding=\"10\">");
html("<tr align=\"left\" valign=\"top\" bordercolor=\"50cf00\">");
html("<td width=\"30%\">");
html("<font face=\"bauhaus 93\" size=\"6\" color=\"007f00\">");
html("<p>ESTUFA</p>");
html("<p>MULTIUSO</p>");
html("</font>");
html("</td>");
html("<td width=\"30%\" bgcolor=\"A8f000\">");
html("<h2>");
html("<p>Temperatura:</p>");
html(TEMP);
html(HORA);
html(MODO);
html("</h2>");
html("</td>");
html("<td width=\"40%\" bgcolor=\"f0f800\">");
html("<h3>");
for(i=0;i<3;i++)
{
  j=0;
  serialESP.print("<p>");
  serialESP.print(CAMBIO[i][j]);
  serialESP.print("CAMBIO: ");
  for(j=1;j<3;j++)
  {
    serialESP.print(CAMBIO[i][j]);
  }
  serialESP.print(":");
  for(j=3;j<5;j++)
  {
```

```

    serialESP.print(CAMBIO[i][j]);
  }
  serialESP.print(" &deg;C ");
  for(j=5;j<7;j++)
  {
    serialESP.print(CAMBIO[i][j]);
  }
  serialESP.print(" ");
  for(j=7;j<9;j++)
  {
    serialESP.print(CAMBIO[i][j]);
  }
  serialESP.println("</p>");
}
i=0;
j=0;
html(FINAL);
html("</h3>");
html("</td>");
html("</tr>");
html("</table>");
html("</font>");
html("</body>");
html("</html>");

```

Elaboración: Propia

Como podemos observar la función html envía cadenas de caracteres entre comillas, pero en algunos caso podemos ver que se envían arreglos como: TEMP, HORA y MODO. Estos arreglos son modificados con los comandos de configuración ingresados en el monitor serial del Arduino.

Por otro lado podemos ver un bucle for, que por cada ciclo envía una fila de datos de la segunda celda de la página web, el cual anida a otros bucles de este tipo, los cuales envían la hora de cambio, minuto de cambio, temperatura y razón de cambio respectivamente.

Imágenes del circuito elaborado se observarán en la Fig. A.1 y Fig. A.2 de Anexos, donde se verá el modelo de ESP WiFi utilizado, también se verá el PIC utilizado como las conexiones realizadas.

Para el correcto funcionamiento del módulo ESP tuve que utilizar un circuito que convierta el voltaje de salida del puerto serial de 5V a 3.3V aproximadamente, también de manera similar la señal serial del módulo ESP se amplificó de 3.3V a 5V.

4.2 Contratación de Hipótesis

- Logré modelar un controlador de temperatura entre los rangos de la temperatura ambiente y $^{\circ}\text{C}$ 90, pero no se comprobó su funcionamiento con organismos ni cultivos.
- Obtuve una configuración más sencilla de parámetros de funcionamiento en comparación con las estufas que utilizan LCD.
- Pude comprobar en la simulación que fue posible obtener un control de la razón de cambio de temperatura, pero en la realidad no se realizaron pruebas.
- Con respecto al monitoreo inalámbrico pude contrastar que el monitoreo dependerá de las buenas condiciones de instalación de nuestro WiFi, esto en el caso de ambientes grandes, pero en espacios pequeños no ocurrirá lo mismo. Con relación a las alarmas convencionales tiene ventaja a más distancia, pero desventaja por la instalación de antenas WiFi.

CONCLUSIONES

PRIMERO: Se realizó la simulación de un controlador de temperatura, con un sobrepaso máximo nulo y un tiempo de establecimiento corto, lo que me permitió modelar una estufa de cultivo con una mayor precisión. Todo ello se logró usando un controlador óptimo cuadrático con un regulador óptimo proporcional Integral.

SEGUNDO: Se logró obtener una configuración más sencilla de varios parámetros, como la temperatura, tiempo de inicio, razón de cambio y cambio del modo de trabajo de la misma. Todo ello mediante él envió de comandos por el monitor serial del Arduino.

TERCERO: Por otro lado, logré simular el controlador de la razón de cambio de temperatura, todo gracias al uso de consecutivos escalones unitarios, que me permitió tener un control de la pendiente de la rampa de cambio de temperatura.

CUARTO: Finalmente al montar un pequeño servidor WiFi se pudo un monitoreo inalámbrico de las características de funcionamiento de esta estufa – multiuso, todo gracias a la elaboración de una página web.

RECOMENDACIONES

PRIMERO: Durante el desarrollo de este proyecto observé varias cosas que se podrían mejorar en el mismo, la primera de ellas es el modelado matemático incluyendo la acción de un ventilador que permita evacuar el aire caliente del interior de la cámara, lo que permitiría una disminución más rápida del calor al apagar la resistencia calefactora, también podríamos obtener un tiempo de establecimiento más corto.

SEGUNDO: Por otro lado, la razón de cambio de temperatura se podría mejorar, ya que si el tiempo de establecimiento es menor se puede tener rampas de cambio de temperatura con escalones menos notorios.

TERCERO: Por el lado de una configuración más sencilla, se podría tener una configuración auxiliar mediante el uso de unos pulsadores y un LCD, o algo similar.

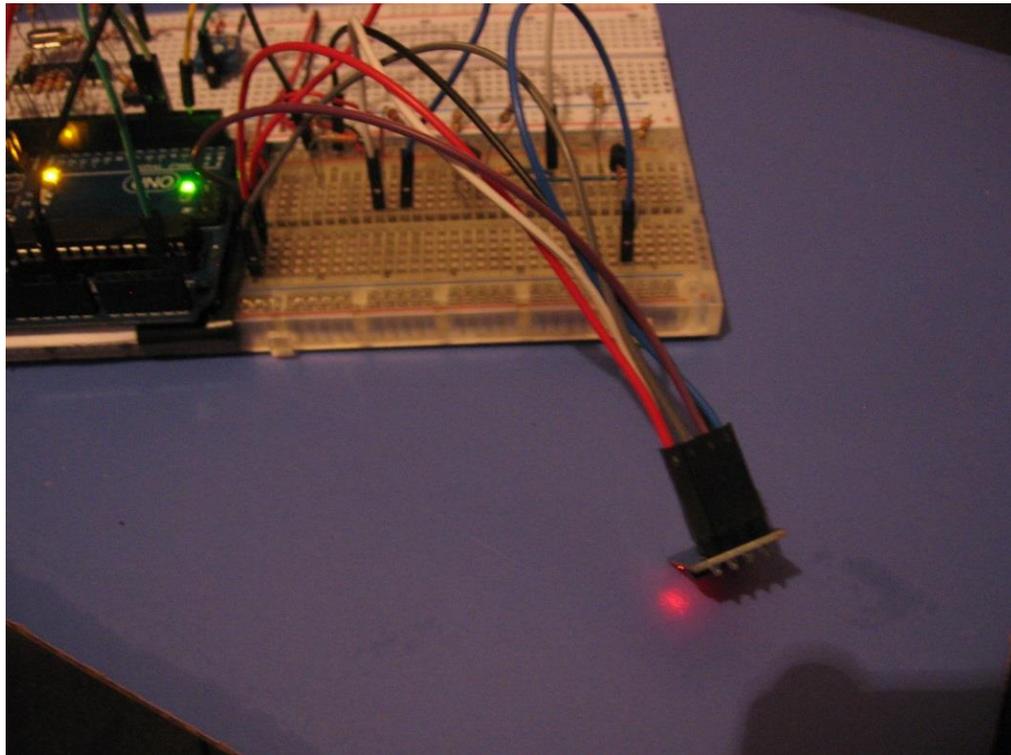
CUARTO: El monitoreo inalámbrico se mejoraría mediante el uso de una página web que nos permita el llenado de formularios, lo que daría también la posibilidad de una configuración de las características del equipo a distancia.

REFERENCIAS

- Arian. (s.f.). *Pt100, su operación, instalación y tablas*. Obtenido de Arian: <http://www.arian.cl/downloads/nt-004.pdf>
- ATICA. (s.f.). *Manual Básico de Creación de Páginas Web*. Obtenido de Universidad de Murcia: <https://www.um.es/atica/documentos/html.pdf>
- Catedu. (19 de Junio de 2014). *Arduino Referencia*. Obtenido de Arduino Apuntes: http://facilitamos.catedu.es/secundariatecnologia/wp-content/uploads/sites/19/2014/06/ARDUINO_APUNTES_2013.pdf
- COMPIC. (s.f.). *COMPIC*. Obtenido de MOC3041: <http://www.complic.es/con-salida-atric/363-optoacoplador-moc-3041.html>
- del Valle Hernández, L. (s.f.). *93. ESP8266 todo lo que necesitas saber del módulo WiFi para Arduino*. Recuperado el 2017, de Programarfacil: <https://programarfacil.com/podcast/esp8266-wifi-coste-arduino/>
- Ermanno. (17 de Junio de 2010). *Introducción a las redes WiFi*. Obtenido de EsLa Red: http://www.eslared.org.ve/walc2012/material/track1/05-Introduccion_a_las_redes_WiFi-es-v2.3-notes.pdf
- I.C.T, S.L. (s.f.). *I.C.T, S.L*. Obtenido de ESTURA DE CULTIVO BACTERIOLOGICA INCUDIGIT: <http://www.ictsl.net/productos/01d63694a80f7db0d/01d636948a1087e25.html>
- Moreno, A. R. (2001). *Control Avanzado*. Lima, Peru.
- Naylamp. (s.f.). *Tutorial ESP8266 Parte I*. Recuperado el 17 de Marzo de 2017, de Naylamp Mechatronics: http://www.naylampmechatronics.com/blog/21_Tutorial-ESP8266-Parte-I.html
- Palacios, E., Remiro, F., & López, L. (2004). *Microcontrolador PIC16F84*. Del Valle: ALFAOMEGA GRUPO EDITOR, S.A.
- Profeyony. (s.f.). *Arduino*. Obtenido de Diseño y Manufactura: <https://sites.google.com/site/temasdedisenoymanufactura/arduino>
- TBEM. (s.f.). *TBEM*. Obtenido de 4N25, Opto Aislador con salida de Transistor: <http://teslabem.com/optoacoplador-4n25.html>
- TECNOLOGIA. (s.f.). *TECNOLOGIA*. Obtenido de OPTOACOPLADOR: <http://www.areatecnologia.com/electronica/optoacoplador.html>
- upna. (s.f.). *Tecnologías Wi-Fi*. Obtenido de unavarra: <http://www.tlm.unavarra.es>
- Zuffli. (16 de Mayo de 2014). *Módulo L298 para Motores Puente H de 2 Canales 4A Pico*. Obtenido de Teslabem: <https://electronilab.co/tutoriales/tutorial-de-uso-driver-dual-l298n-para-motores-dc-y-paso-a-paso-con-arduino/>

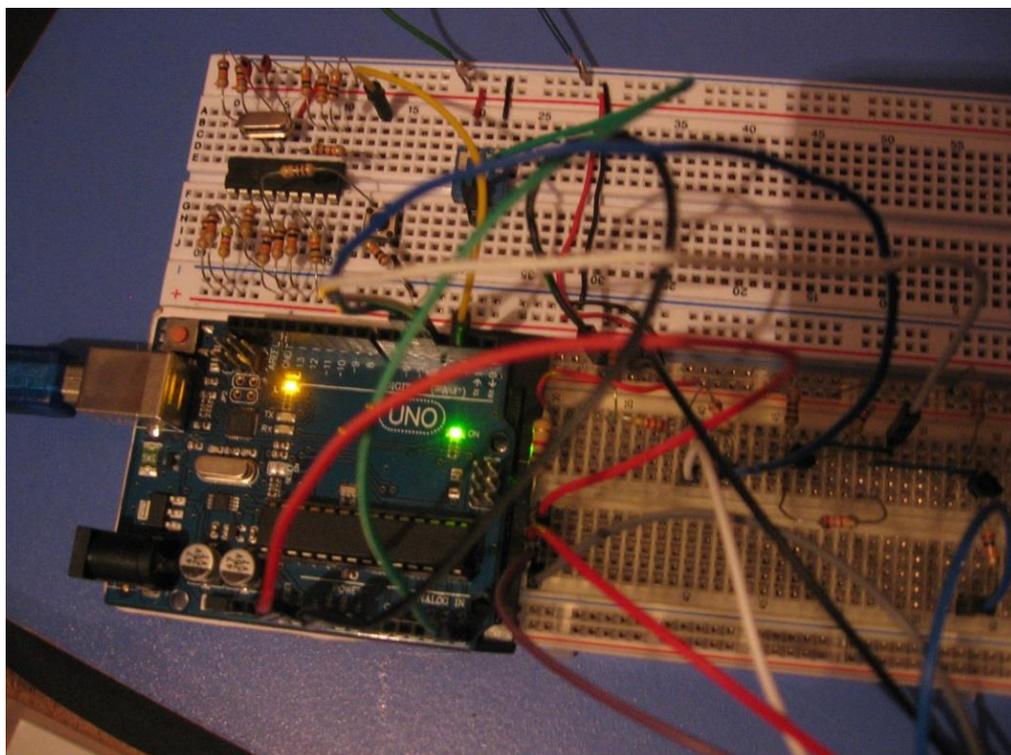
ANEXOS

ANEXO A 1: En esta figura se muestra un Arduino con el conexionado y en funcionamiento de la configuración de parámetros.



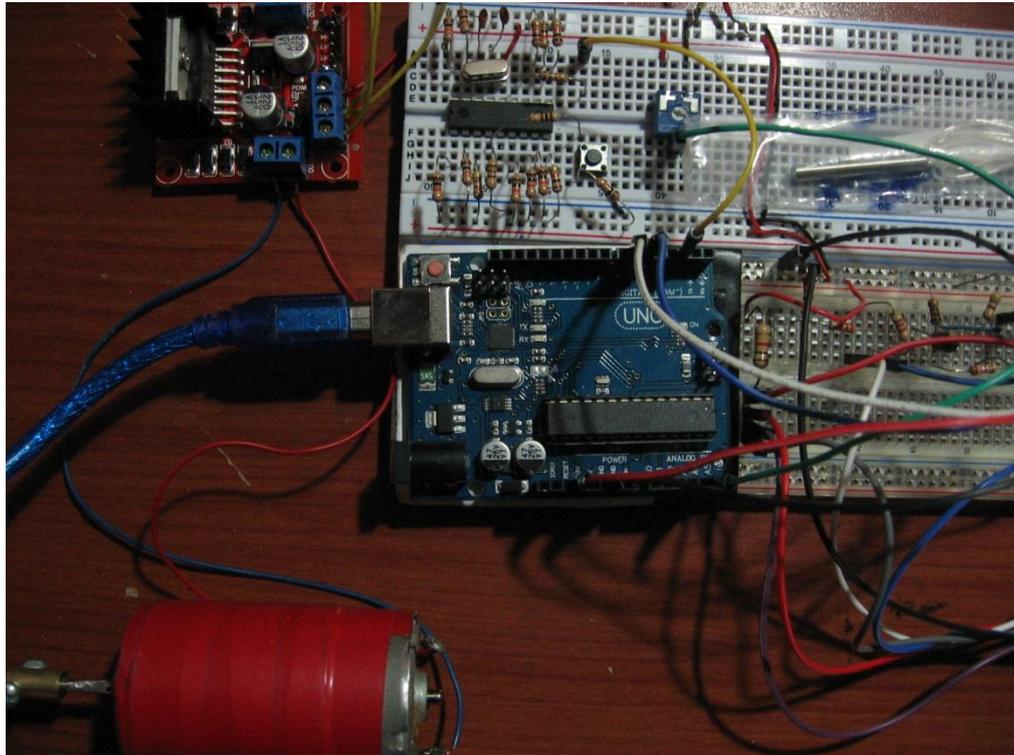
Elaboración: Propia

ANEXO A 2: En esta figura se muestra el conexionado y funcionamiento del módulo ESP8266, cuando se encuentra activo se le enciende un LED Rojo como se muestra en la figura.



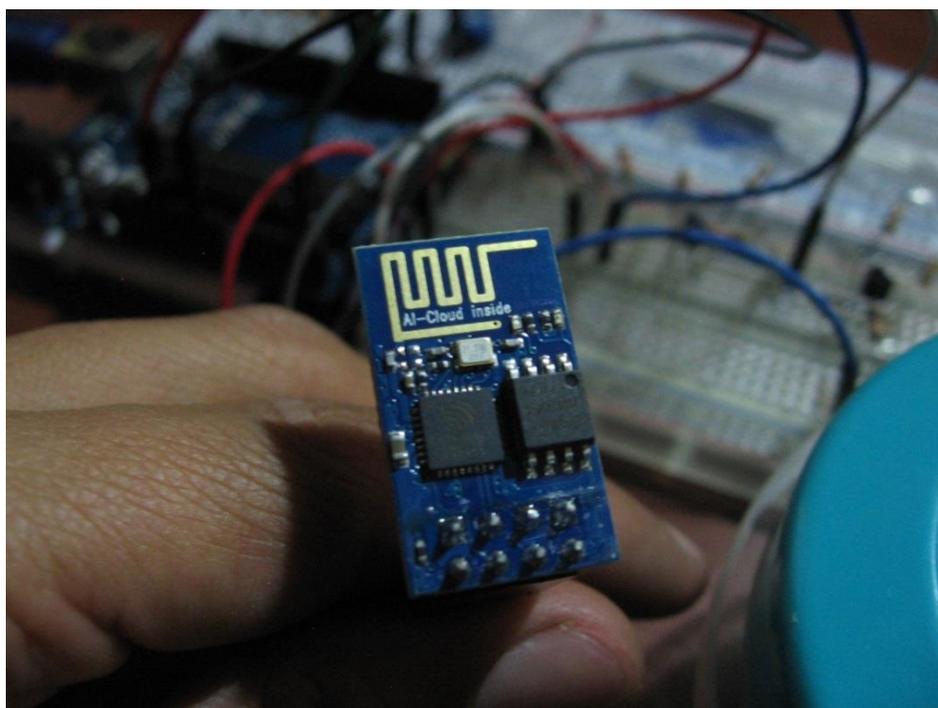
Elaboración: Propia

ANEXO A 3: En esta figura se muestra los principales componentes de este proyecto, sin incluir la estufa y su resistencia calefactora, el motor visto sirve para mover los huevos en la cámara cuando estemos en el modo incubadora.



Elaboración: Propia

ANEXO B 1: Aquí se muestra el módulo ESP8266 utilizado para la elaboración de nuestro servidor web para monitoreo inalámbrico.



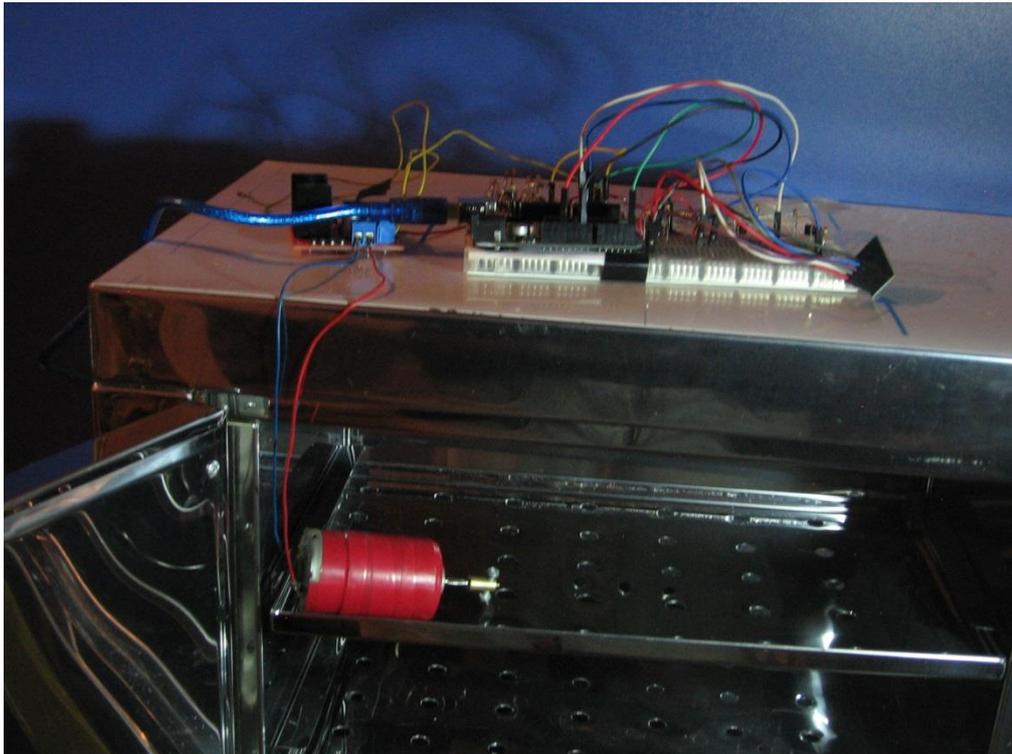
Elaboración: Propia

ANEXO B 2: En esta figura se puede observar una prueba de funcionamiento de la página web elaborada.



Elaboración: Propia

ANEXO C 1: En esta fotografía se observa la estufa que adquirí para realizar posibles pruebas, y calibraciones del funcionamiento del controlador elaborado.



Elaboración: Propia

ANEXO C 2: En esta imagen se puede observar que el motor DC está a altura de las parrillas de esta estufa, el motivo es porque este motor sirve para realizar el movimiento oscilatorio de las parrillas cuando el equipo está en modo incubadora.



Elaboración: Propia