

UNIVERSIDAD POLITÉCNICA DE VICTORIA



CONTROL DE HORNO

PRÁCTICA 3.1.3

ESCUELA: Universidad Politécnica de Victoria

ASIGNATURA: Control digital

MAESTRO: Dr. Yahir Hernández Mier

ALUMNO: Ángel Arturo Ramírez Suárez

GRUPO: IM 8-1

Objetivo. Comprender las operaciones involucrados en la realización de controladores en tiempo continuo y tiempo discreto mediante el análisis y la simulación de sistemas.

Instrucciones.

Práctica 3.1.3. Realice la simulación del control PID de temperatura de un horno utilizando un ATMEGA328P programado mediante la interfaz de Arduino y un esquemático realizado en Proteus con los datos facilitados en clase. Calcule las ganancias iniciales del controlador por el método de la curva de reacción de Ziegler-Nichols. Enseguida varíe los valores de las ganancias hasta obtener una respuesta con tiempo de subida corto y poco sobretiro. Fabrique un reporte en PDF y suba al sitio del curso. Concluya sobre el comportamiento del sistema para las diferentes ganancias probadas.

INTRODUCCIÓN.

El objetivo de los controladores de sistemas en tiempo discreto es mantener una señal de referencia, la cual es considerada la variable controlada de acuerdo a la modificación de una señal conocida como la variable manipulada, esto permite obtener una respuesta que se mantiene dentro de un umbral correspondiente a la respuesta deseada por parte del sistema.

En esta práctica se hace uso de las herramientas de simulación disponibles para simular la respuesta de un sistema de control de temperatura (horno, estufa, etc.) y aplicando un controlador ATMEGA328P se regula la respuesta del mismo hasta obtener un valor deseado.

DESARROLLO.

HERRAMIENTAS.

- Computadora personal.
- Proteus (aplicación de simulación de componentes y circuitos electrónicos).
- Scilab (aplicación de procesamiento matemático).
- Arduinov1.0.3 (aplicación para programación de controladores ATMEGA).

PROCEDIMIENTO.

1. Se elabora el circuito del sistema de temperatura como se muestra en la **figura 1**.

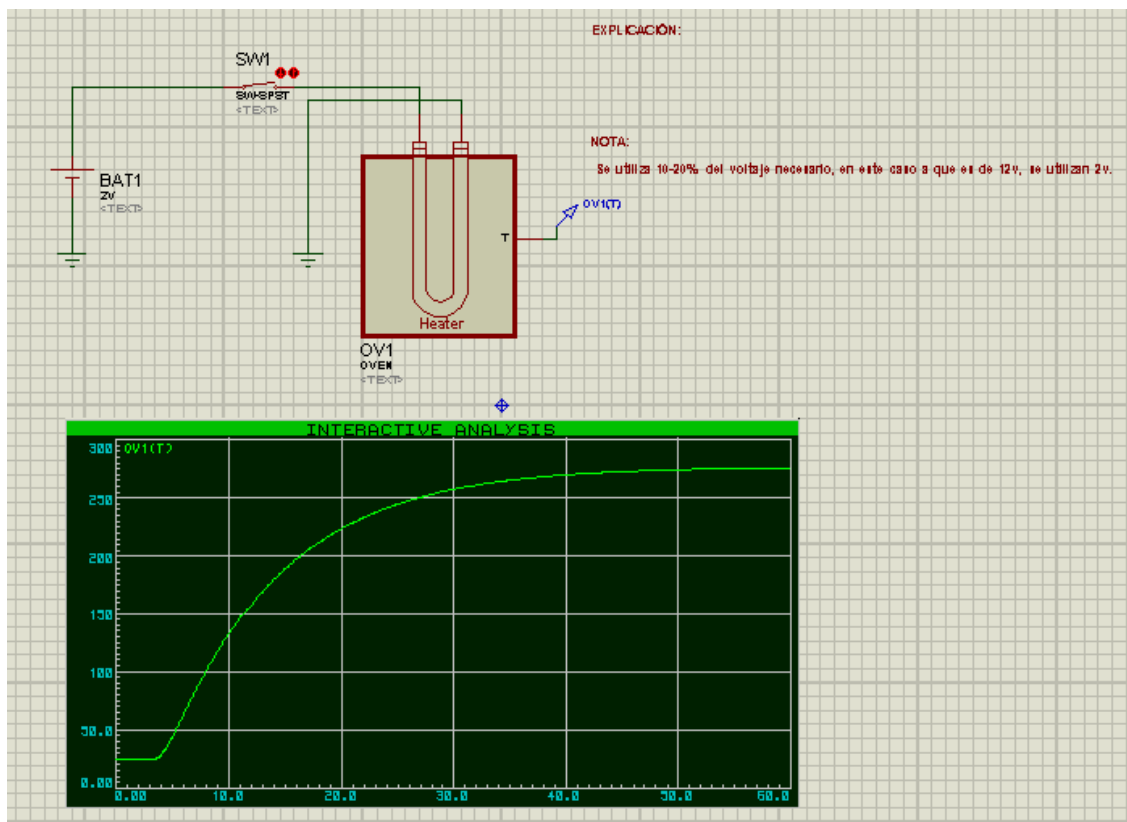


Figura 1. Sistema de temperatura.

Dicho circuito consta de una batería de 2 volts que se dirige a un interruptor normalmente abierto, el cual se dirige hacia un sistema de calefacción el cual puede representar un horno, calefactor, etc.

2. Se coloca una punta de prueba **OV1(T)** (figura 2) y se genera la gráfica de análisis interactiva para poder observar la respuesta del sistema con respecto al tiempo seleccionando el elemento **INTERACTIVE** que se encuentra en la sección **GRAPHS** tal y como puede verse en la figura 3.

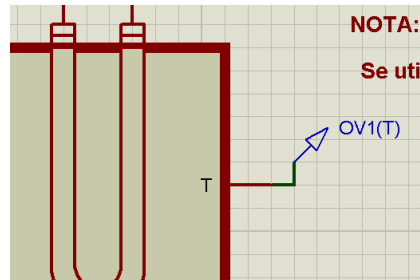


Figura 2. Punta de prueba OV1(T) colocado en el horno para observar el cambio en la temperatura.

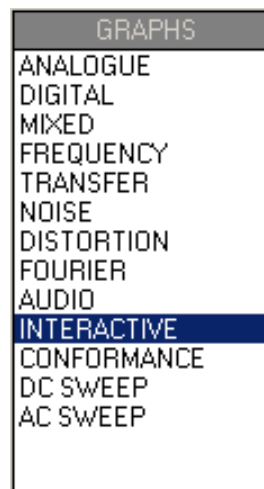


Figura 3. Selección de las distintas herramientas disponibles en Proteus para simulación.

3. Se configuran los parámetros del sistema de la siguiente manera:

Para el horno:

- Temperatura ambiente (ambient temperatura): 25°C
- Resistencia térmica al ambiente (thermal resistance to the ambience): 0.7°C/W
- Constante de tiempo del horno (oven time constant): 10seg
- Constante de tiempo del calentador (heater time constant): 1seg
- Coeficiente de temperatura (temperature coefficient): 1V/°C
- Potencia del calentador (heating power): 120W

Para la gráfica INTERACTIVE:

- Tiempo de inicio (start time): 0
- Tiempo de parada (stop time): 60

- Para iniciar la simulación y graficar la respuesta se presiona la barra espaciadora, lo cual inicia la captura de la información en un tiempo predeterminado. Se mantuvo el interruptor abierto durante 5 segundos y pasado este periodo se cerró, con lo cual se inicia el proceso. La respuesta generada se aprecia en la gráfica que se muestra en la **figura 4**.

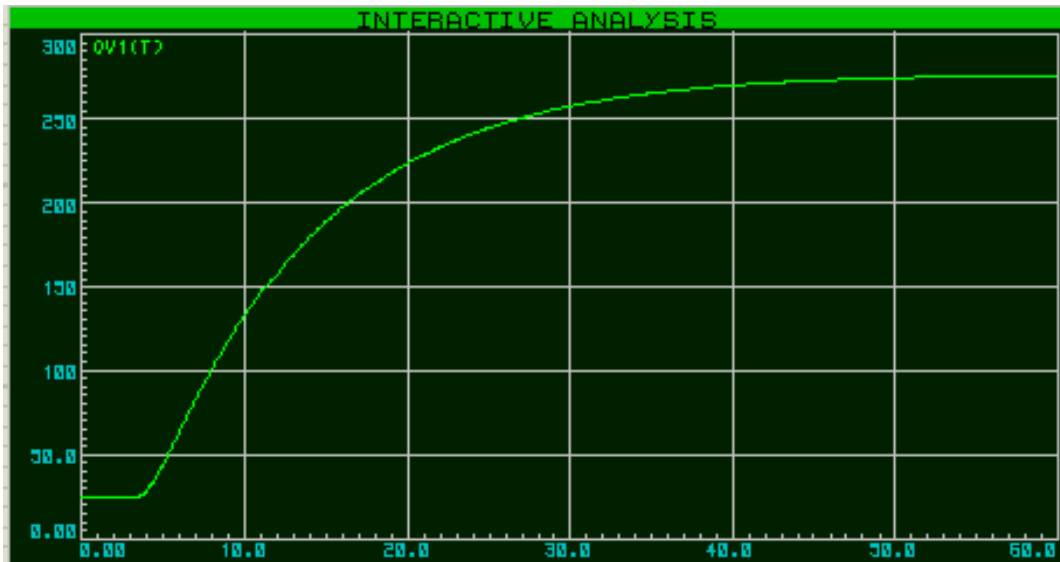


Figura 4. Gráfica de respuesta del sistema.

- Posteriormente se elaboró un código para Scilab (**apéndice A**) el cual permite obtener las ganancias originales en el sistema por medio del método de la curva de reacción, obteniendo la siguiente respuesta (**figura 5**):

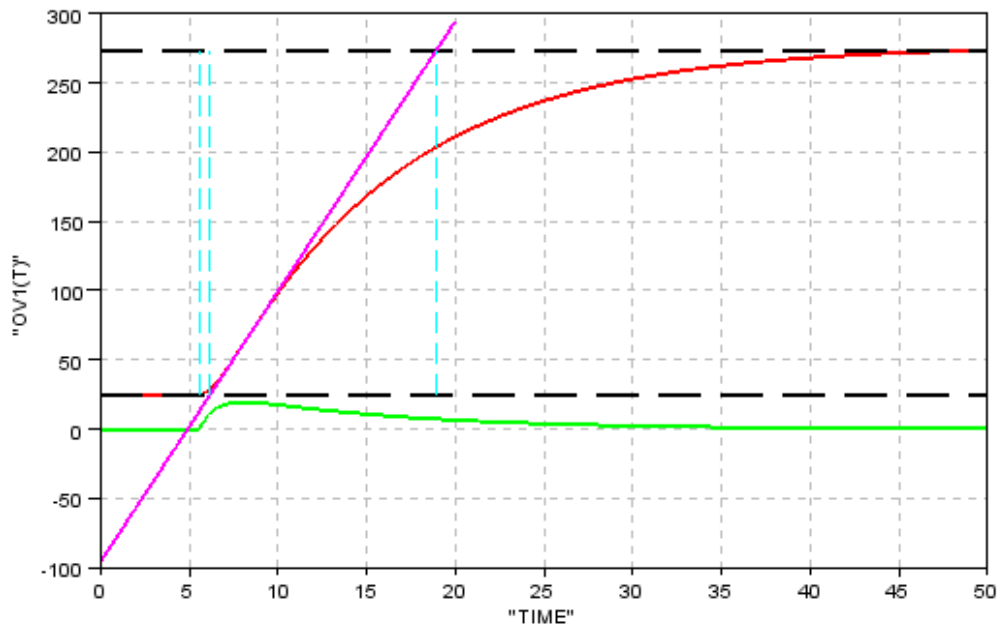


Figura 5. Respuesta del sistema graficando límites superiores, inferiores y tiempos de respuesta.

- Mediante el método de Ziegler-Nichols es posible obtener las ganancias iniciales del sistema K_p , K_i y K_d .

```

Kpd=0.186618639984673
Ki=0.015340300296754
Kd=0.615179188821862
    
```

Figura 6. Ganancias generadas utilizando el método de Ziegler-Nichols.

- Posteriormente se varían los valores de las ganancias hasta lograr una respuesta estable y rápida en el sistema.
- Hecho esto, se elaboró el programa en arduinov1.0.3 para introducir estos valores al controlador y con ello modificar la respuesta del sistema de calefacción en base a las ganancias obtenidas en el análisis anterior (**apéndice B**).
- Se realizaron las modificaciones pertinentes al circuito para incluir el controlador en el sistema como se observa en la **figura 7**.

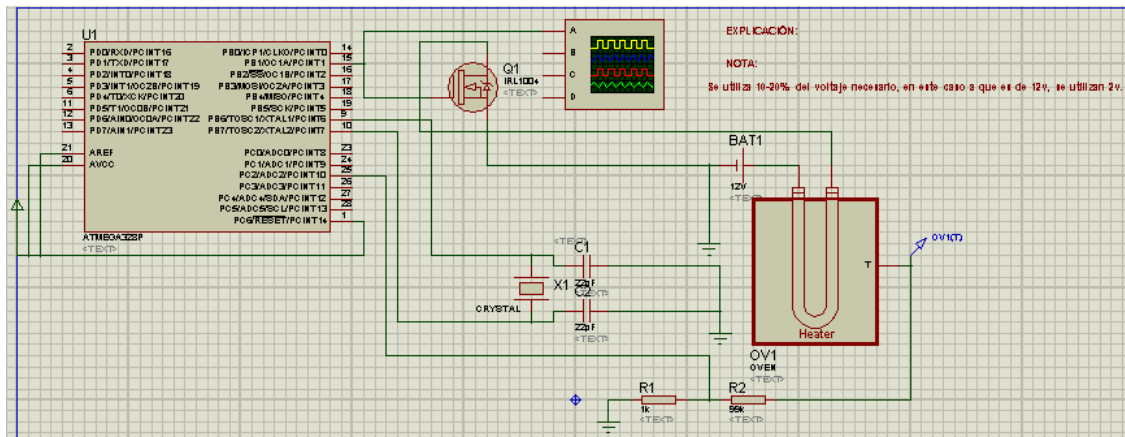


Figura 7. Circuito de calefacción con sistema de control.

- Se observa la respuesta de la gráfica del sistema al control, la cual se observa en la **figura 8**.

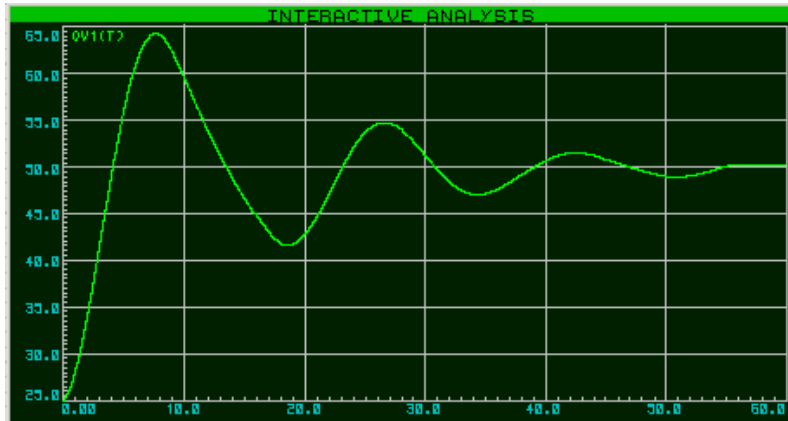


Figura 8. Respuesta del sistema bajo la acción de control.

CONCLUSIONES.

Mediante la elaboración de esta práctica fue posible observar la respuesta de un sistema a una entrada tipo escalón y calcular los parámetros del mismo de manera práctica, haciendo uso del método de la curva de reacción de Ziegler-Nichols para determinar las ganancias K del sistema y posteriormente, mediante una modificación empírica de las mismas, lograr modificar el tiempo de respuesta del sistema a la misma entrada.

Para ello se hizo uso de numerosas herramientas computacionales que permitieron obtener y analizar datos de diversas índoles, como es el caso de Proteus, con el cual se elaboraron los circuitos del sistema y que generó los datos que fueron analizados por medio del uso de Scilab.

Finalmente el controlador arduino permitió generar un código para el control del sistema y que puede ser implementado en un sistema físico para la alteración de las ganancias.

APÉNDICE A.

Aplicación desarrollada en Scilab para la obtención de las ganancias del sistema.

```
clear
M= read_csv('D:\Control Digital\DCControl - Simulación 1 - Respuesta a entrada escalón\DCControl-
datos_horno_v1.DAT' );
[nFil, ncol]=size (M);
t=strtod(M(2:$,1));
v=strtod(M(2:$,2));
scf();
plot(t,v,'r','LineWidth',2)
xgrid
xlabel(M(1,1))
ylabel(M(1,2))

//Limite superior
plot(t, ones(1, length(t))*v($),'--k','LineWidth',2)
//limite inferior
plot(t, ones(1, length(t))*v(1),'--k','LineWidth',2)

//Aproximacion de las derivadas
dvdv = zeros(1, length(t))
dvdv(1) = (v(2) - v(1)) / (t(2) - t(1));
dvdv($) = (v($) - v($-1)) / (t($) - t($-1));
for i = 2 : length(t) - 1
    dvdv(i) = (v(i+1) - v(i-1)) / (t(i+1) - t(i-1));
end
plot(t,dvdv, 'g', 'LineWidth',2)//grafica de la derivada

//Pendiente de la recta tangente.
[m, cas] = max(dvdv); //m es el elemento y cas es la casilla del vector; por cierto m es la pendiente maxima
y_cero = v(cas);
x_cero = t(cas);
//Grafica de la recta tangente a la derivada

x = 0:0.1:20;
y = y_cero + m*(x-x_cero);
plot(x,y, 'm', 'LineWidth', 2)

//Instante en que la curva empieza a crecer
indx = find(dvdv > 0.4);
t0 = t( indx(1));
//Instante en que la tangente cruza por y_cero
t1 = (v(1) - y_cero + m* x_cero) / m;
//Instante en que la tangente cruza por y_inf
t2 = (v($) - y_cero + m*x_cero) / m;

//Grafica de las lineas punteadas en t0, t1 y t2

plot( [t0 t0], [v(1) v($)], '--c')
plot( [t1 t1], [v(1) v($)], '--c')
plot( [t2 t2], [v(1) v($)], '--c')

//Calculo de L, T y K del metodo de Ziegler-Nichols
L= t1 - t0;
```

```
Tzn = t2-t1;
K = (v($) - v(1)) / (2 - 0); //ganancia total
//calculo de las ganancias del PID
Kp = 1.2*Tzn / (K*L);
Ti = 2*L;
Td = L/2;
T = 0.1; //Periodo de muestreo
Kpd = Kp - ((Kp * T) / (2*Ti));
Ki = Kp * T / Ti;
Kd = Kp * Td/T;
mprintf('Kpd=%2.15f\n',Kpd)
mprintf('Ki=%2.15f\n',Ki)
mprintf('Kd=%2.15f\n',Kd)

ha=gca();
ha.grid = [color('gray'), color('gray')];
```

APÉNDICE B.

Aplicación del controlador de temperatura para el microcontrolador ATMEGA328P.

//Control de temperatura

int sensorPin = A2; **//Pin para leer el sensor. Pin 25**

int salPWM = 9; **//La salida PWM utilizando el PIN15 del ATMEGA328P.**

//VARIABLES PARA HACER OPERACIONES

int valADC = 0; **//Salida del ADC**

float mkT = 0.0; **//Variable que contendra la señal de control.**

int mkT_ent = 0; **//Señal de control, dato entero.**

float ckT = 0.0; **//Temperatura leida a la salida.**

float rkT= 500.0; **//Temperatura de referencia deseada.**

float ekT = 0.0; **//Error.**

float sum_ekT = 0.0; **//Error acumulado.**

float ekT_ant = 0.0; **//Error anterior.**

float Kp = 0.186627251306975;

float Ki = 0.015341379569262;

float Kd = 0.615193857599314;

void setup ()

{

//Inicializacion de la comunicacion serial.

Serial.begin(9600);

Serial.println("PWM"); **//Encabezado.**

valADC = 0;

}

void loop(){

//Temperatura leída.

valADC = analogRead(sensorPin);

```

//Conversion a voltaje (mV)
ckT = (valADC /1023.0)*5000.0;

//Diferencia entre la referenica y el valor leído.
ekT = rkT - ckT;

//Error acumulado.
sum_ekT = sum_ekT + ekT;

//Cálculo de la señal de control.
mkT = Kp*ekT + Ki*sum_ekT + Kd*(ekT - ekT_ant);

//Conversión a entero.
mkT_ent = mkT;

//Para que la señal salga de rango(Windup)
if (mkT>1023)
mkT_ent=1023;
else if (mkT <0)
mkT_ent = 0;

//Serial.println
//Se divide entre cuatro para ajustar a 8 bits.
analogWrite(salPWM, mkT_ent/4);

//Actualización del error anterior.
ekT_ant = ekT;

//Periodo de muestreo (T) en ms
delay(100);
}

```

BIBLIOGRAFÍA.

1. Katsuhiko Ogata. *Sistemas de control en tiempo discreto*. Pearson. Segunda Edición.