

# Diseño, Simulación y Construcción de un Control PID Aplicado a un Sistema Térmico

*M. en C. Romeo Urbieto Parrazales*  
*Profesor e Investigador del CINTEC-IPN.*

**E**ste artículo trata tres aspectos mayores que son: el diseño, la simulación, y la construcción de un control **PID** (proporcional más integral más derivativo) aplicado a un sistema térmico.

El diseño del sistema de control de temperatura comprende el modelado y cálculo de los parámetros del controlador, el proceso de temperatura, y las interfaces entre la computadora y el proceso. El comportamiento de la temperatura del agua en el recipiente es un modelo de primer orden expresado con la transformada  $s$ . En este modelo se muestra una curva de la variable de temperatura con respecto al tiempo. En la curva del proceso de temperatura de lazo abierto se obtuvieron tres parámetros de interés que son: la zona muerta, ganancia del proceso, y la constante de tiempo. El algoritmo de control empleado es un sistema clásico **PID** cuyos parámetros **K<sub>p</sub>**, **K<sub>i</sub>**, **K<sub>d</sub>** se obtuvieron por el método de Ziegler & Nichols. El algoritmo de control de temperatura implantado en la PC viene expresado en una ecuación de diferencias.

La simulación trata de representar la variable de temperatura del agua de un recipiente con capacidad de un litro. El rango de temperatura del agua en el recipiente es de 25°C

a 75°C con una precisión de 0.1°C. La representación de la variable de temperatura se llevó a cabo por medio de la creación de un programa gráfico estructurado en lenguaje C++, el cual contiene las rutinas de graficación y de cálculo de la variable de temperatura.

La construcción del sistema de control **PID** aplicado a la temperatura del agua contenida en un recipiente de un litro consiste de tres módulos mayores: el módulo de cálculo, consistente en una computadora PC 386; el módulo de conversión y de potencia, consistente en una interface electrónica conteniendo un puerto paralelo para capturar y enviar datos, conversión de datos de A/D y D/A, amplificador de corriente y voltaje para excitar la resistencia térmica que calienta al agua, y amplificador para sensor de temperatura; y el módulo físico, consistente de un recipiente de porcelana con capacidad de un litro, equipado con resistencia térmica y sensor de temperatura.

---

## 1. Introducción

---

El objetivo del sistema de control de temperatura del agua contenida en un recipiente es mantener la temperatura del agua a un valor deseado ("set point") **T<sub>o</sub>**.

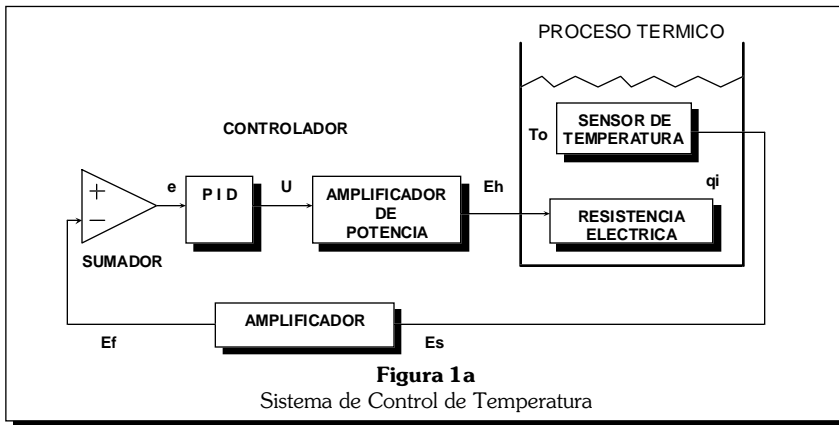
La temperatura del sensor dentro del agua en el recipiente produce

un voltaje  $E_s$ , proporcional a la temperatura **T<sub>o</sub>**. El voltaje obtenido por el sensor se amplifica a un factor **K<sub>1</sub>**, obteniendo un voltaje **E<sub>f</sub>**, el cual es aplicado a un sumador juntamente con el valor deseado de voltaje **E<sub>vd</sub>**. La diferencia de la señal de voltaje desde el sumador es una señal de error **e**.

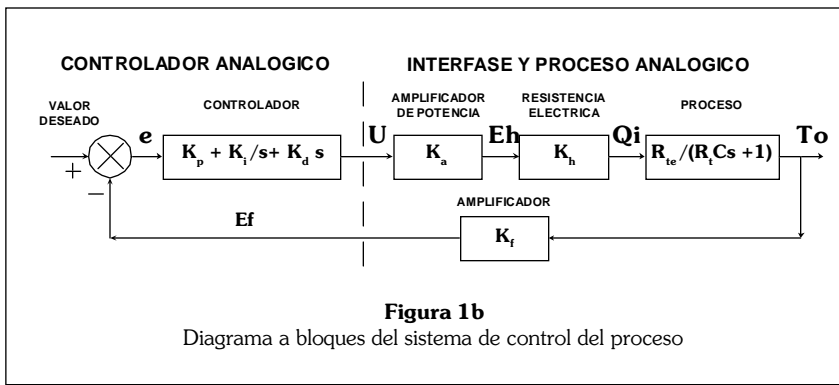
La señal de error **e** se introduce a la entrada del controlador **PID** (proporcional más integral más derivativo) generando una señal de salida de control **U**, que alimenta después a un amplificador de potencia cuya salida es **E<sub>h</sub>**, para excitar una resistencia eléctrica sumergida en el agua; este a su vez entrega una entrada de calor **q<sub>i</sub>** causando que la temperatura del agua sea **T<sub>o</sub>** y también como el voltaje de retroalimentación **E<sub>r</sub>**, para aumentarlo, y la señal de diferencia en la salida del amplificador sumador **e**, para decrementarlo (a consecuencia de la retroalimentación negativa).

Cuando la señal de diferencia se aproxima al valor de cero, el excitador de corriente corta la señal de entrada de la resistencia eléctrica, por lo tanto, el calor del agua deja de aumentar. Como resultado, la temperatura del agua en el recipiente es la del valor deseado de temperatura **T<sub>o</sub>**.

El sistema de control de temperatura de la **figura 1a** se puede representar por un diagrama a blo-



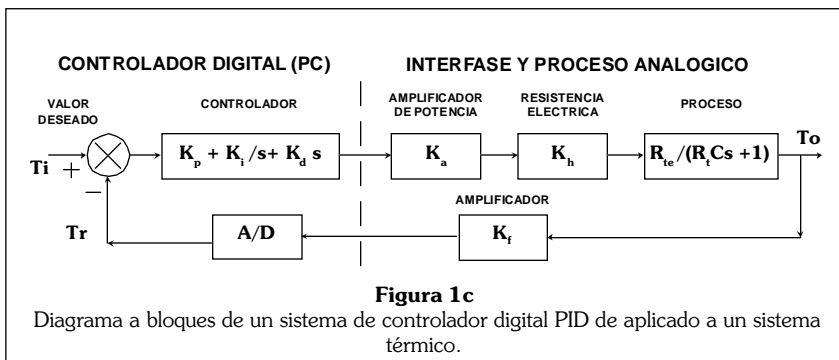
**Figura 1a**  
Sistema de Control de Temperatura



**Figura 1b**  
Diagrama a bloques del sistema de control del proceso

ques, como se puede observar en la **figura 1b**.

El sistema de la **figura 1b** es un sistema analógico y el controlador se implanta con elementos analógicos. Para el caso de estudio a efectuar el sistema de control será estructurado usando una computadora digital tipo PC. El uso de la computadora digital como controlador lleva consigo un convertidor de señal: el convertidor analógico a digital. Lo demás queda igual (ver **figura 1c**).



**Figura 1c**  
Diagrama a bloques de un sistema de controlador digital PID de aplicado a un sistema termico.

alimentado por la resistencia eléctrica al agua,  $q_i$ , y el calor perdido en el recipiente y el ambiente puede ser igual a la elevación de calor en el agua. La acumulación de calor en el agua es proporcional a la razón de cambio de la temperatura del agua, con  $C_t$  de capacitancia térmica del agua, siendo esta la constante de proporcionalidad. De esta manera,

el flujo de calor es:

$$q_i - q_o = C_t (dT_o / dt) \text{ [Btu/s]} \quad (2.1.1)$$

donde  $q_i(t)$  = calor alimentado al agua [Btu/s]

$$q_o = (T_o - T_a) / R_t \quad (2.1.2)$$

$q_o$  = flujo de calor [Btu/s] a través de agua - ambiente.

$R_t$  = Resistencia térmica del agua en el recipiente [°F, sec/Btu].

$C_t$  = Capacitancia térmica del agua en el recipiente [Btu/°F].

Sustituyendo (2.1.2) en (2.1.1) :

$$R_t C_t (dT_o / dt) + T_o = R_t q_i + T_a \quad (2.1.3)$$

$$t To + To = R_t q_i(t) + T_a \quad (2.1.4)$$

Para linealizar el sistema se tiene:

$$T_o(0) = 0 \quad (2.1.5)$$

$$T_a = R_{te} q_i - R_t q_i \quad (2.1.6)$$

donde  $R_{te}$  es la resistencia térmica efectiva de las paredes del recipiente; entonces:

$$\tau To(t) + To(t) = R_{te} q_i(t) \quad (2.1.7)$$

$\tau = R_t C_t$  es la cte. térmica del recipiente con agua.

Este tipo de ajuste matemático puede fácilmente hacerse en situaciones donde el calor  $q_i$  es aplicado a una razón constante y donde  $T_a$  lo

es también. Así mismo,  $R_{te}$  puede ser tratado como otro parámetro del sistema el cual incluye efectos de carga.

Pasando a transformada de Laplace (2.1.7):

$$T_o(s) = \frac{Q_i(s) R_{te}}{\tau s + 1} \quad (2.1.8)$$

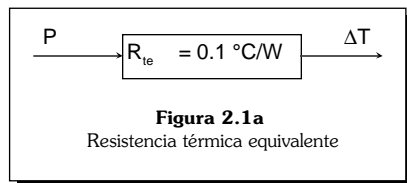
La resistencia térmica equivalente o ganancia del proceso térmico se puede obtener por medios experimentales midiendo los watts de entrada o aplicados a la resistencia eléctrica y el rango de temperatura del agua. Los datos que se tienen de la resistencia eléctrica son :

$$\begin{aligned} R &= 28.3 \text{ W} \\ I &= 4.22 \text{ A} \\ V &= 124 \text{ VAC} \end{aligned}$$

$$P = R I^2 = (28.3)(4.22)(4.22) = 504 \text{ W} \quad (2.1.9)$$

$$\Delta T = T_{75} - T_{25} = (75^\circ\text{C} - 25^\circ\text{C}) = 50^\circ\text{C} \quad (2.1.10)$$

$$\begin{aligned} R_{te} &= \Delta T / P = 50^\circ\text{C} / 504 \text{ W} \\ &= 0.1^\circ\text{C/W} \end{aligned} \quad (2.1.11)$$

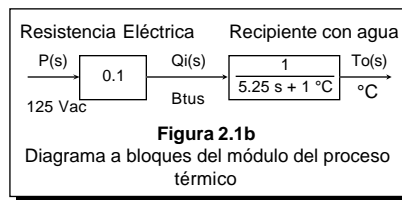


La constante de tiempo térmica  $\tau$ , y la zona muerta  $q$ , también se pueden obtener por método empírico, calculando el tiempo de la temperatura al 63.2 % de  $T_o$ , esto es:

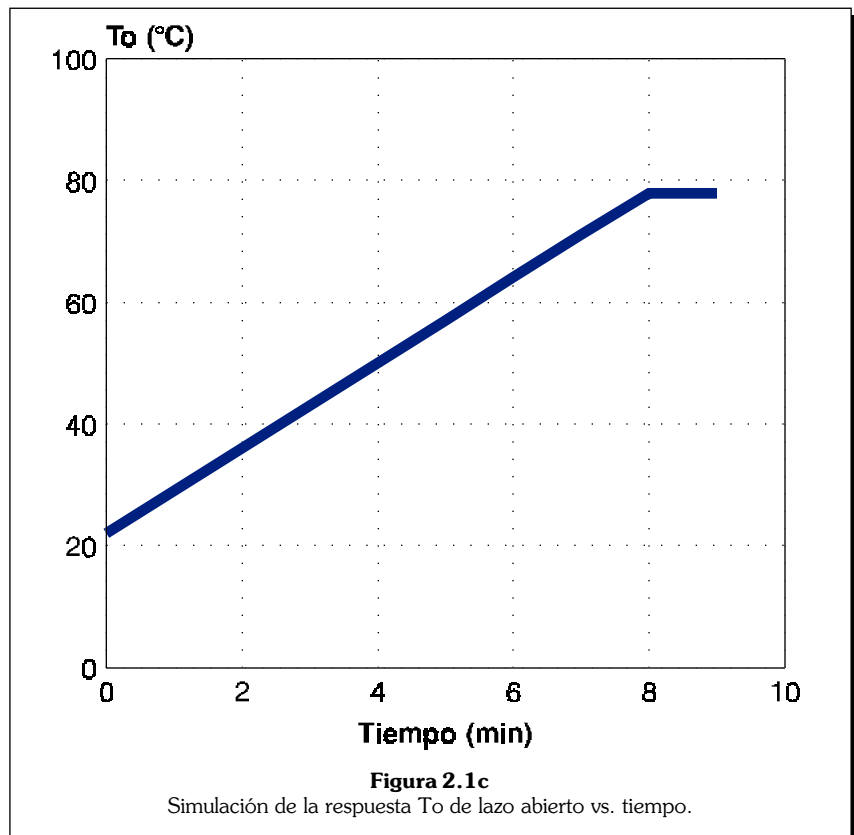
$$\begin{aligned} T_i &= (0.632)(50^\circ\text{C}) + 25^\circ\text{C} \\ &= 31.6^\circ\text{C} + 25^\circ\text{C} \\ &= 56.6^\circ\text{C} \end{aligned}$$

Calentando el agua a este valor se obtiene un tiempo  $\tau = 5.25$  minutos. La zona muerta  $q$  fue de

15 segundos. El modelo térmico correspondiente queda como sigue:

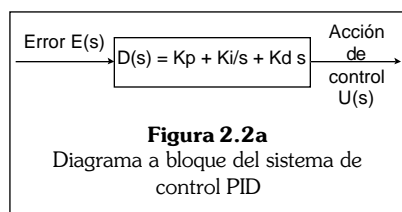


La respuesta en el tiempo se puede apreciar en la **figura 2.1c**.



## 2.2 Módulo de Control

La función de transferencia de un algoritmo **PID** está dada por :



se pueden calcular los parámetros del controlador **PID**:  $K_p$ ,  $K_i$ , y  $K_d$ .

Sustituyendo sus valores :  
 $K = 0.1, \tau = 5.25 \text{ min}, q = 0.25 \text{ min}.$  (2.2.1)

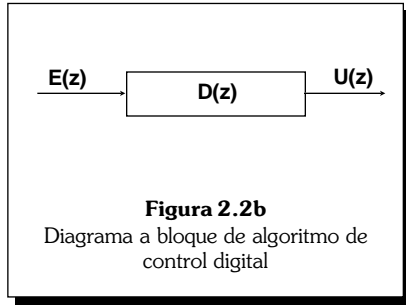
$$\begin{aligned} K_p &= 1.2 t / K q \\ &= (1.2)(5.25\text{min}) / (0.1^\circ\text{C/W})(0.25 \text{ min}) \\ &= (6.3) / (0.025) = 252 \end{aligned} \quad (2.2.2)$$

$$K_i = 1/2 q = 1/(2)(0.25) = 1/0.5 = 2 \quad (2.2.3)$$

$$K_d = 0.5 q = (0.5)(0.25) = 0.125 \quad (2.2.4)$$

Se asume que  $T = 0.5$  min.

En el modo digital, el controlador **PID** toma la forma:



El controlador PID se expresa como:

$$D(z) = K_p + K_i T \frac{z+1}{2(z-1)} + K_d \frac{z-1}{Tz} \quad (2.2.5)$$

donde:

$D(z)$  es el algoritmo de control digital PID.

$T$  es el período de tiempo entre muestreos.

$K_i$  ganancia integral.

$K_p$  es la ganancia proporcional.

$K_d$  es la ganancia derivativa.

Entonces, la variable controlada  $U(z)$  es:

$$U(z) = D(z)E(z) = K_p E(z) + [K_i T \frac{z+1}{2(z-1)}] E(z) + [K_d \frac{z-1}{Tz}] E(z) \quad (2.2.6)$$

Tomando la transformada inversa de  $z$ , se tiene:

$$u(k) = u(k-1) + K_p e(k) - K_p e(k-1) + [K_i T e(k)]/2 + [K_i T e(k-1)]/2 + K_d e(k)/T - 2K_d e(k-1)/T + K_d e(k-2)/T \quad (2.2.7)$$

Sustituyendo los valores de  $K_p$ ,  $K_i$ , y  $K_d$  encontrados en la simulación se tiene que:

$$u(k) = u(k-1) + 252.75 e(k) - 252 e(k-1) + 0.25 e(k-2) \quad (2.2.8)$$

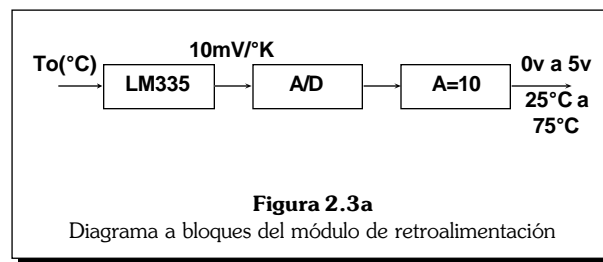
Este será el algoritmo de control **PID** final a implementar, siendo  $u(k-1)$  la señal de control o comando de control retardado un período de tiempo,  $e(k-1)$  la señal de error también retrasado un período de tiempo, y  $e(k-2)$  dos períodos de tiempo.

### 2.3 Módulo de Retroalimentación

Otro módulo de importancia en el sistema de control térmico es su retroalimentación, el cual sustituye al hombre en su intento de controlar un proceso. El módulo de retroalimentación consiste básicamente de: un sensor de temperatura y un sistema de amplificación o acondicionamiento de señal de medida de 0 a 5 voltios.

El sensor de temperatura se diseñó con un circuito integrado LM335 de National Semiconductor, basado en el principio del diodo. El circuito integrado, mediante una polarización externa adecuada, provee a su salida un voltaje del orden de los 10mV/°K, a una corriente de 1mA. (Ver diagrama esquemático final).

El rango de temperatura de operación de este circuito integrado va desde los -55°C hasta los 100°C. En este caso de estudio trabajará desde los 25°C a los 75°C. El circuito integrado para 25°C da un voltaje de salida de 2.98 voltios.



Este voltaje de salida se obtiene a partir de:

0°C equivalen a 273°K, como el dispositivo avanza cada 10mV/°K, entonces para esta temperatura se tienen 2.73 voltios. Entonces para 25°C que va ser el comienzo serán (273 + 25)°K que es igual a 298°K por 10 mV/°K, se llega a 2.98 voltios de salida en el sensor; así, 75°C serán 3.48 voltios. La señal de este sensor se aplica directamente a un convertidor Analógico a Digital para capturar los voltajes entre 2.98 voltios a los 3.48 voltios, siendo una diferencia de 0.5 voltios.

Usando el programa de simulación antes expuesto, se incluye un circuito amplificador diferencial con referencia en 2.98 voltios para compararlo con la lectura de los 25°C y que se obtengan 0 voltios. El amplificador tendrá una ganancia de 10 para obtener un rango de voltaje de 0 a 5 voltios @ 10 mA, que representan el rango de 25°C a 75°C. El diagrama final a bloques de retroalimentación quedará como en la **figura 2.3a y 2.3b**.

### 2.4 Módulo de Excitación

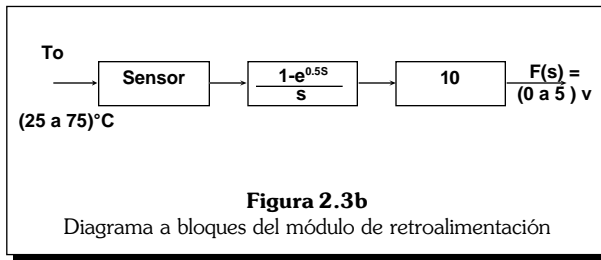
La potencia  $u$  aplicada a la resistencia térmica es una función del error entre el comando de temperatura y la temperatura medida por el sensor, y es calculada por la PC. Esta potencia es aplicada a la resistencia con la ayuda de circuitos de estado sólido, como un dispositivo opto-electrónico y un triac (ver esquema electrónico).

El dispositivo optoelectrónico

aislará la PC del potencial aplicado a la resistencia y además dará el prendido y apagado inteligente al interruptor de potencia o triac. El triac es un interruptor de potencia del estado

sólido, y será el que dé la potencia final a la resistencia térmica. Este interruptor se cerrará cada  $tp$  se-

gundos o minutos según sea el error; si el error es grande el período de preñado será grande y viceversa, y el apagado será equivalente al tiempo de muestreo  $T_s$ .



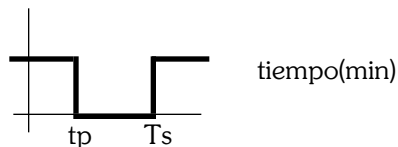
Entonces, la potencia promedio aplicada a la resistencia térmica en el intervalo  $T_s$  es:

$$u = \frac{1}{T_s} \int_0^{T_s} V^2 dt = \frac{V^2 t_p}{R T_s} \quad (2.4.1)$$

$$t_p = \frac{u T_s}{V^2 / R} \quad (2.4.2)$$

Dependiendo de la potencia requerida a ser aplicada a la resistencia eléctrica,  $t_p$  es calculada en la PC. Un pulso de ancho  $t_p$  se dispone cada vez que exista un instante de muestreo.

P(Watts)

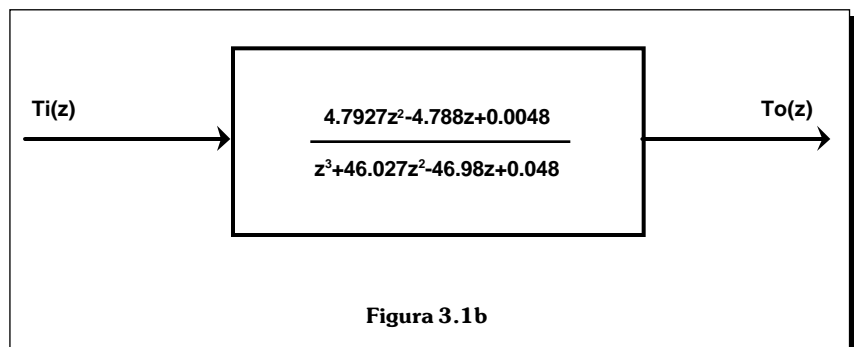
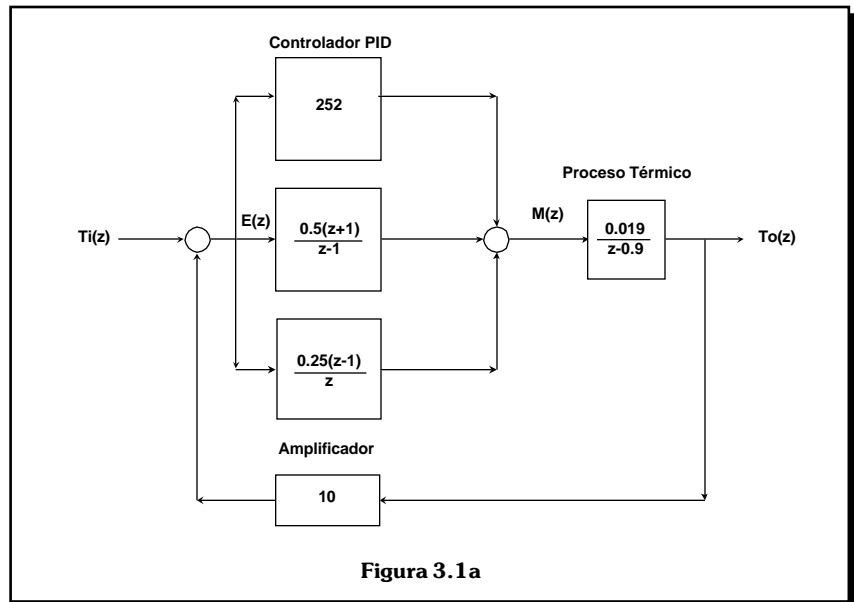


### 3. Simulación

Para la simulación del sistema de control se requiere la función de transferencia total  $T_i(z)T_o(z)$ . Las **figuras 3.1a y 3.1b** muestran esta función tanto en su representación gráfica como en forma matemática.

A partir de esta función de transferencia se obtiene la respuesta del sistema térmico en tiempo discreto  $k$ .

$$T_o(k) = -46.027 * T_o(k-1) + 46.98 * T_o(k-2) - 0.048 * T_o(k-3)$$



$$+ 4.7927 * T_i(k-1) + 4.788 * T_i(k-2) + 0.0048 * T_i(k-3) \quad (3.1)$$

La simulación del sistema de control aplicado al proceso térmico tiene como interés ajustar los parámetros de control  $K_p$ ,  $K_i$ , y  $K_d$  del controlador para una sintonía perfecta. A continuación se presenta el programa que lleva a cabo esto.

```

/*
                INSTITUTO
        POLITECNICO NACIONAL
        CENTRO DE INVESTIGACION
        TECNOLOGICA EN COMPUTACION

Proyecto:
    Software gráfico para un sistema de control PID aplicado a un
    sistema térmico.

Participantes:
    M. en C. Romeo Urbieta Parrazales
    Col. Granjas México, Delg. Iztacalco
*/

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <graphics.h>
#include <conio.h>
#include "dos.h"
#include <string.h>
#include "borracad.c"

/* Definición de variables para el 8255 */
#define PA 0x100
#define PB 0x101
#define PC 0x102
#define PCONTROL 0x103

/* Configuración de los puertos del 8255 */
#define PALABRA1 0x82
#define PALABRA2 0x8B

/* Constantes para el ADC y DAC */
#define BAJO 0
#define ALTO 55000
#define RETAR 0
#define LIM 500

/* Definición de Funciones */
void ini_graficos(void); /* rutina de inicia gráfico */
void cie_graficos(void); /* rutina de cierra gráfico */
void ecua_graf(void); /* rutina ecuaciones a graficar */
void pinta_ejes(void); /* rutina de pintar ejes */
int adc(void);
void dac(void);

int main()
{
    ini_graficos(); /* llama subrutina de iniciar gráficos */
    pinta_ejes(); /* llama subrutina de pintar ejes */
    ecua_graf(); /* llama subrutina de pintar ejes */
    getche(); /* espera tecla */
    cie_graficos(); /* llama subrutina de cerrar gráfico */
    return 0;
}

void ini_graficos(void)
{
    int modo, controlador=DETECT;
    int error_graf; /* controlador = 0; */
    modo = 4;
    initgraph(&controlador, &modo, ""); /* "c:\tgc"; */
    error_graf = graphresult();
    if (error_graf != grOk)
    {
        printf("Error en gráficos -> ", grapherrormsg(error_graf));
        exit(1);
    }
}

void cie_graficos()
{
    closegraph();
}

```

```

}
void ecua_graf()
{
    char buffer[100];
    double T0, T1, T2, T3, R0, R1, R2, R3;
    int xorg = 50, yorg = 440; /* El origen esta en 50,440 */

    k = 0;
    T0 = 0.0;
    x = xorg + k;
    y = yorg - T0;

    putpixel(x, y, color);

    k = 1;
    R0 = R1 = R2 = cajon[i].M

    T1 = 4.7927*R0;
    x = xorg + k;
    y = yorg - T0;

    putpixel(x, y, color)

    k = 2;
    R0 = R1 = R2 = R3 = cajon[i].M
    T2 = -46.027*T1 + 4.7927*R1 + 4.788*R0;
    x = xorg + k;
    y = yorg - T0;

    putpixel(x, y, color)

    k = 3;

    T3 = -46.027*T2 + 46.98*T1 + 4.7927*R2 + 4.788*R1 + 0.0048*R0;
    x = xorg + k;
    y = yorg - T0;
    putpixel(x, y, color);

    for (k = 4; k <= 500; k++) {

        T4 = -46.027*T3 + 46.98*T2 - 0.048*T1 + 4.7927*R3 + 4.788*R2 +
        0.0048*R1;

        C0 = C1;
        C1 = C2;
        C2 = C3;
        C3 = C4;

        x = xorg + k;
        y = yorg - T0;
        putpixel(x, y, color);
    }

void pinta_ejes()
{
    char buffer[200];
    int pasov = 50; /*45 */
    int pasoh = 50;
    int xorg = 45, xorg0 = 50, xorg1 = 550;
    int yorg = 189, yorg0 = 435, yorg1 = 440, y=390;

    setbkcolor(BLUE);
    setcolor(WHITE);

    rectangle(70, 1, 550, 50);
    outtextxy(240, 5, "CINTEC");
    outtextxy(150, 20, "Medición de voltaje y temperatura");

    rectangle(70, 100, 550, 147);
    setttextstyle(DEFAULT_FONT, HORIZ_DIR, 1);
    outtextxy(225, 103, "Valores a Graficar ");
    outtextxy(160, 120, " Voltaje Temperatura");

    moveto(xorg, yorg); /* POSICION INICIAL */
    lineto(xorg, yorg1); /* MARCO VERTICAL */
}
}

```

```

lineto (xorg1, yorg1);      /* MARCO HORIZONTAL */
moveto (xorg, yorg1);
moveto (xorg, yorg);      /* POSICION INICIAL */

while (y > yorg)
{
    /* CONTADOR VERTICAL DE MARCAS VERTICALES */
    moveto (xorg, y);      /* POSICION INICIAL */
    lineto (xorg0, y);     /* PRIMERA MARCA VERTICAL SUPERIOR */
    y = y - pasov;        /* ESPACIO ENTRE MARCA Y MARCA */
}

setttextstyle(SMALL_FONT, VERT_DIR, 5);
sprintf(buffer," Voltaje");
outtextxy(0,175,buffer);

setttextstyle(SMALL_FONT, HORIZ_DIR, 4);
sprintf(buffer,"20");
outtextxy(18,390,buffer);
setttextstyle(SMALL_FONT, HORIZ_DIR, 4);
sprintf(buffer,"40");
outtextxy(18,340,buffer);
setttextstyle(SMALL_FONT, HORIZ_DIR, 4);
sprintf(buffer,"60");
outtextxy(18,290,buffer);
setttextstyle(SMALL_FONT, HORIZ_DIR, 4);
sprintf(buffer,"80");

outtextxy(18,240 ,buffer);
setttextstyle(SMALL_FONT, HORIZ_DIR, 4);
sprintf(buffer,"100");
outtextxy(18,190,buffer);
sprintf(buffer," TEMPERATURA ( C)");
outtextxy(10,160,buffer);

moveto (xorg, yorg1);     /* POSICION INICIAL */
xorg = xorg + pasoh;      /* DATO DE PRIMER MARCA HORIZON- */
                          /* TAL IZQUIERDA */

while (xorg <= xorg1)
{
    /* CONTADOR HORIZONTAL DE MARCAS */
    moveto (xorg, yorg1); /* POSICION DE PRIMER MARCA */
                          /* HORIZONTAL */
    lineto (xorg, yorg0); /* PRIMERA MARCA HORIZONTAL */
    xorg = xorg + pasoh; /* ESPACIO ENTRE MARCA Y MARCA */
                          /* HORIZONTAL */
}

setttextstyle(SMALL_FONT, HORIZ_DIR, 5);
sprintf(buffer,"250 300 350 400 450 500 550 600 650 700
75");
outtextxy(49,450,buffer);
setttextstyle(SMALL_FONT, HORIZ_DIR, 5);
sprintf(buffer," TIEMPO (°mS)");
outtextxy(200,460,buffer);
}

```

## 4.0 Implantación

La implantación del sistema consiste de: algoritmo de control instalado en la computadora como “software”; una tarjeta que contiene un convertidor A/D como entrada de señal y un circuito integrado opto-electrónico, así como dos interfaces electrónicas, una para acondicionar la señales del sensor y la otra para la resistencia eléctrica.

### 4.1 Comparador y Algoritmo de Control

El comparador del sistema de la figura 1c es como sigue:

$$e(k) = T_i(k) - T_f(k) \quad (4.1.1)$$

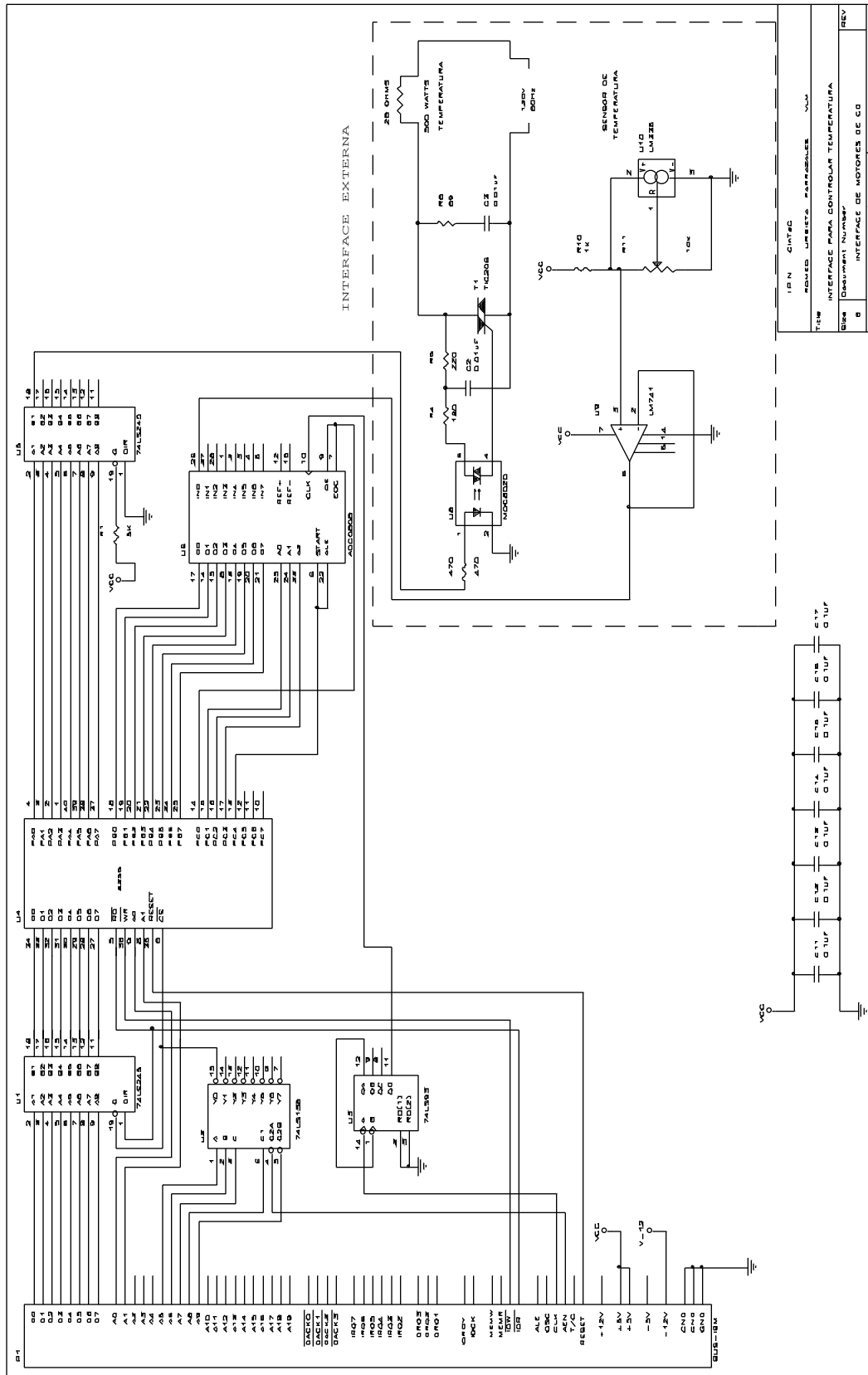
Tomando la transformada inversa de  $z$ , de la ecuación 2.2.7 se tiene:

$$\begin{aligned}
 u(k) = & u(k-1) + K_p * e(k) \\
 & - K_p * e(k-1) + [K_i * T * e(k)] / 2 \\
 & + [K_i * T * e(k-1)] / 2 + K_d * e(k) / T \\
 & - 2K_d * e(k-1) / T \\
 & + K_d * e(k-2) / T
 \end{aligned} \quad (4.1.2)$$

Sustituyendo los valores de  $K_p$ ,  $K_i$ , y  $K_d$  encontrados en la simulación y desarrollando, se tiene que la acción de control es:

$$\begin{aligned}
 u(k) = & u(k-1) + 252.75 * e(k) - 252 * e(k-1) \\
 & + 0.25 * e(k-2) \quad (4.1b)
 \end{aligned} \quad (4.1.3)$$

el cual será el algoritmo de control PID final a implementar en programa en tiempo real, siendo  $u(k-1)$  la señal de control o comando de control retardado un período de tiempo,  $e(k-1)$  la señal de error también retrasado un período de tiempo, y  $e(k-2)$  dos períodos de tiempo. El programa de control está estructurado en lenguaje C++ en tiempo real.



IB N	CINTAC
TIM	RODED UMRETC. SAPPORACE. VVV
TIW	INTERFACE PARA CONTROLAR TEMPERATURA
SIZE	DRAWING NUMBER
B	INTERFACE DE MOTORES DE 50
DATE	MARZO 31 1995
	STAGE
	AN



### Conclusiones

El sistema de control de temperatura responde muy bien a los cambios de "set point" de temperatura pedidos, mientras que la inercia térmica es abatida por el sistema de control computarizado. La precisión de este sistema de control es del orden de 0.1%. La implantación del sistema resulta ser barata y versátil.

### Bibliografía

- [1] M. Gopal. Digital Control Engineering. Capítulo 8. Ed. John Wiley & Sons Limited. 1989.
- [2] R. Gayakwad & L. Sokolof. Analog and Digital Control Systems. Pags. 469-482. Ed. Prentice Hall. 1988
- [3] Data Acquisition Linear Devices. Data Book. Pags. 6-21 a 6-29. National Semiconductor . 1989.
- [4] Joseph J. Carr . Sensor and Circuits. Cap.4 Temperature sensors. Ed. Prentice Hall. 1993.
- [5] P.C. Sen. Thyristor DC Drives. Cap.6 Closed Loop Control. Ed. Krieger P.C. 1991.