UNIVERSIDAD TECNOLOGICA NACIONAL
FACULTAD REGIONAL SAN NICOLAS

PROYECTO INTEGRADOR

CONTROL DE TEMPERATURA CON PC

Integrantes**:**

- Basualdo, Sergio
- Castellá, Hernán

Docentes**:**

- Profesor: Poblete, Felipe F.
- Auxiliar: Gonzalez, Mariano

AÑO 2006

INDICE

Introducción

El mundo que percibimos es "analógico". Por "analógico" entendemos la serie de cambios lineales que se manifiestan en el orden natural de las cosas, por ejemplo, la temperatura del medio ambiente cambia gradualmente, el agua fluye suavemente, etc. Todas las entradas sensoriales en un ser humano reciben datos en forma analógica. Cuando las personas empezaron a diseñar máquinas "pensantes" rápidamente descubrieron que los dispositivos que operan bajo principios analógicos eran complicados, sensibles y poco confiables. El problema necesitaba simplificarse y se lo hizo **permitiendo sólo dos estados posibles**. Entonces: ¿Cómo puede una computadora, es decir un dispositivo binario interactuar con el mundo real?

En este proyecto, además de resolver este problema, aplicaremos la teoría de Control en la implementación de un controlador PI discretizado.

Objetivo

Diseñar un control de temperatura del tipo Proporcional/ Integral implementado en PC con interfaz gráfica de operación. El operador podrá, mediante teclado, ingresar el Set Point y la curva de calentamiento deseada. Un usuario avanzado podrá actuar directamente sobre las constantes del controlador.
En pantalla se observará la evolución temporal de la variable medida y el set point, con el tiempo de barrido ajustable.

Temas que integra de Técnicas Digitales III

Arquitectura de las computadoras personales. Componentes del sistema. Buses de la arquitectura PC compatible.

Integrará las siguientes materias:

- Electrónica Aplicada 2 [Amplificadores]
- Medidas Electrónicas 2 [Conversor ADC]
- Sistemas de Control [Controlador PI discreto]
- Instrumentación [Medición de Temperatura]

Profesores consultados

Guillermo Campomar. Titular de la cátedra de Sistemas de Control.

Tema consultado: Discretización de Reguladores PID e implementación en PC.

Culasso, Victor. Titular de la Cátedra de Medidas Digitales II.

Tema consultado: Conversores Analógicos Digitales.

Ramiro Vota: Auxiliar de la Cátedra de Software en Tiempo Real.

Tema consultado: Programación en C orientada a objetos.

Aplicaciones

- *Control de temperatura en incubadoras de recién nacidos*
  En este tipo de aplicaciones se requiere una precisión de +/- 1ºC. Por otro lado no se admiten sobreimpulsos de más del 5%. El tiempo de asentamiento no es un factor crítico.

- *Control de temperatura en incubadoras para aves comerciales*

Es utilizado para finalizar la gestación del ave dentro del huevo. No requiere de un control muy preciso, con una presicion de +/- 1 o 2ºC es suficiente. Comúnmente se debe regular la temperatura a 37ºC.

- *Control de temperatura para estudios del plasma de la sangre*

En este caso, se debería recurrir a un sistema (sensor y control) de una precisión mucho mayor que en los casos anteriores, aproximadamente de una décima de grado. La temperatura debe permanecer constante a 37ºC sin sufrir variación alguna.

Nota: El siguiente artículo fue extraído de la página web del instituto Balseiro

- *Test simplificado para Detección de Coniformes Totales en leche humana:*

Se describe un método para determinación de bacterias coniformes totales en la leche humana ordeñada pasteurizada, con el objetivo de garantizar la calidad, bajo el punto de vista microbiológico, del alimento distribuido por los Bancos de Leche humana.

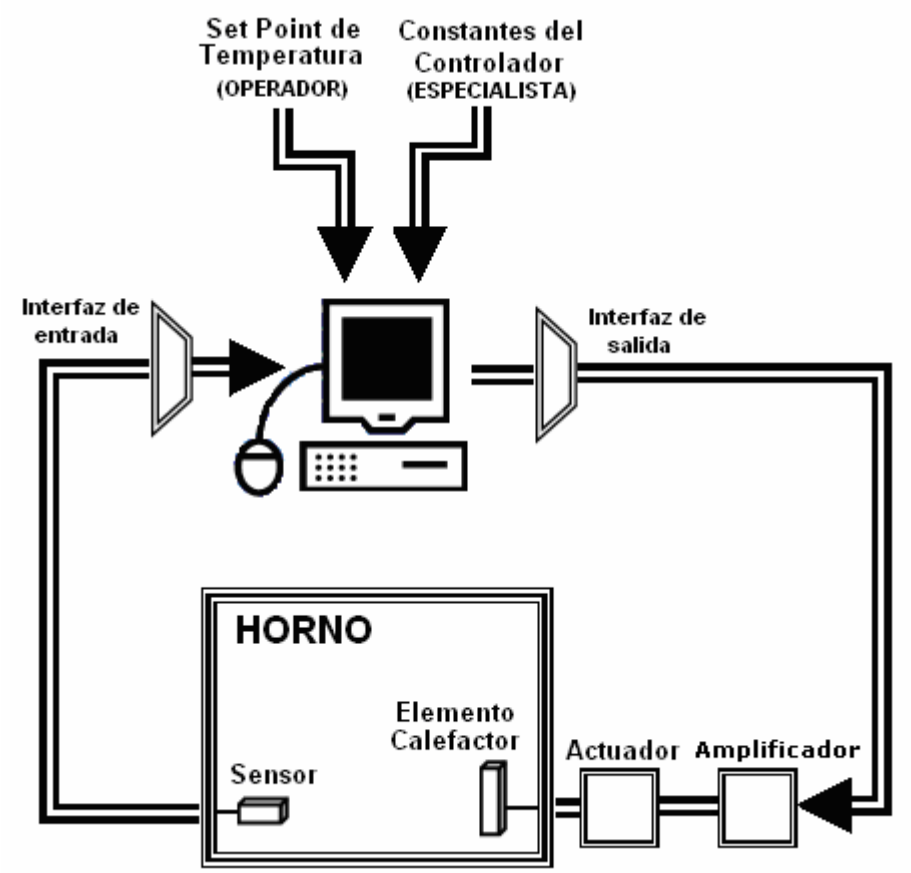
El control de calidad microbiológico de la leche humana ordeñada practicado por la Red Nacional de Bancos de Leche humana en Brasil, sigue la lógica preconizada para alimentos, que instituye la utilización de microorganismos indicadores de calidad sanitaria. En este contexto, el grupo coliforme ha ocupado un lugar destacado, por ser de cultivo simple, económicamente viable y seguro, minimizando la posibilidad de resultados falso-positivos.

A partir del procedimiento clásico para detección de coliformes totales, fue desarrollada una metodología alternativa que consiste en el inóculo de cuatro alícuotas de 1ml cada una, extrayendo con pipeta de forma independiente, en tubos con 10 ml de Caldo Verde Brillante (BGBL) a 5% p/v, con tubos de Dirham en su interior. Tras la inoculación e incubación **a 36 ± 1ºC**, la presencia de gas en el interior del tubo de Dirham. caracteriza resultado positivo.

Equipamiento

Entre otras cosas, se requiere de estufa bacteriológica para cultivo, regulada de 36ºC con exactitud de ±1ºC.

Esquema general



Descripción detallada

1.- PC

2.- Horno

3.-Interfaz de entrada

4.-Interfaz de salida y Elemento Final de Control (Actuador)

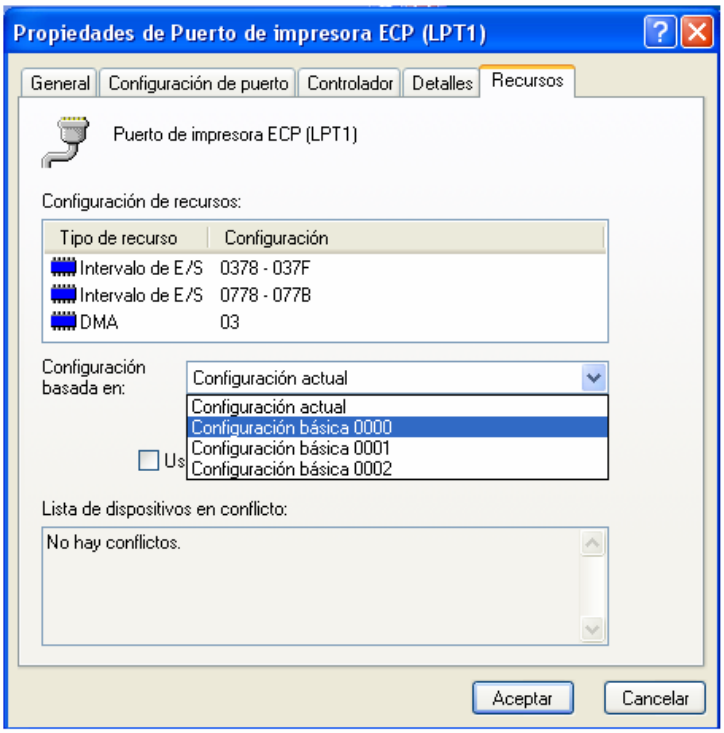
1.- PC

Se cuenta con una PC con un microprocesador Pentium. Teniendo experiencia en programación en lenguaje C (usado en materias como Software en Tiempo Real e Informática II) optamos por realizar el desarrollo de la programación en **Borland C++ 3.1.**

La PC cuenta con un puerto paralelo on-board. Dicho puerto puede ser usado en forma bidireccional. Configurando un registro en particular (registro de control del puerto) lo usaremos como entrada de medición.

Se le conecta un segundo puerto paralelo en un slot EISA. Inicialmente usamos los 8 bits de datos de este puerto como salida de control del elemento de calefacción. Luego el Profesor de la Cátedra, Felipe F. Poblete, nos recomienda que usemos una modulación del tipo PWM para el disparo del elemento final de control (implementado con un TIP 33). Se resuelve la modulación PWM en el mismo software, por lo que sólo necesitamos un bit de salida. Teniendo en cuenta esto podríamos haber usado un bit de control del puerto de entrada como salida de PWM, utilizando de esa forma sólo un puerto, pero debido a que el hardware estaba en ese momento muy avanzado seguimos usando los dos.

Configuración de los puertos paralelos: Nos aseguramos de que ambas direcciones (LPT1 y LPT2) sean distintas. Para esto vamos a Mi Pc -> Propiedades -> Hardware -> Administrador de Dispositivos -> Puertos (Com y LPT) -> Recursos. Aquí se le asigna un bloque de direcciones al puerto 1 y otro bloque al puerto 2. Escogimos para el puerto 1 el intervalo de direcciones de h378 – h37F y para el puerto 2 h278-h27F.



	Dirección del puerto			
	LPT1	LPT2	Bit Nº	DB25 Pin
DATA	378	278	0 1 2 3 4 5 6 7	2 3 4 5 6 7 8 9
STATUS	379	279	0 1 2 3 4 5 6 7	10 11 12 13 14 15 16 17
CONTROL	27A	37A	0 1 2 3 4 5 6 7	18 19 20 21 22 23 24

Modo Bidireccional

El bit 5 del registro de CONTROL (C5) está disponible sólo si se trata de un puerto bidireccional. Si C5="1", el buffer de los datos de salida se pone en alta impedancia, "desconectando" dicho buffer de los pines 2 a 9 del conector del puerto (D0 a D7). Si se escribe al registro de datos, se escribe al buffer pero no a la salida. Esto permite que al leer el puerto, se lea el estado de las entradas y no lo que hay en el buffer.

2.- Horno

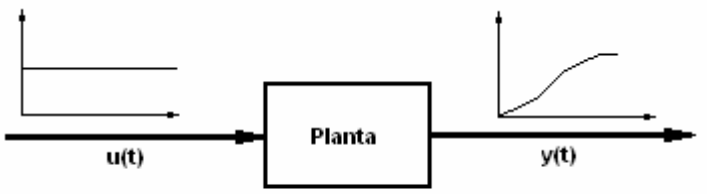
Arquitectura física

El "horno" consta de una resistencia calefactora cerámica de 4,7 ohm / 10W. Junto a ella se ubica el sensor de temperatura y el conjunto está empaquetado con un plástico "termocontraible".

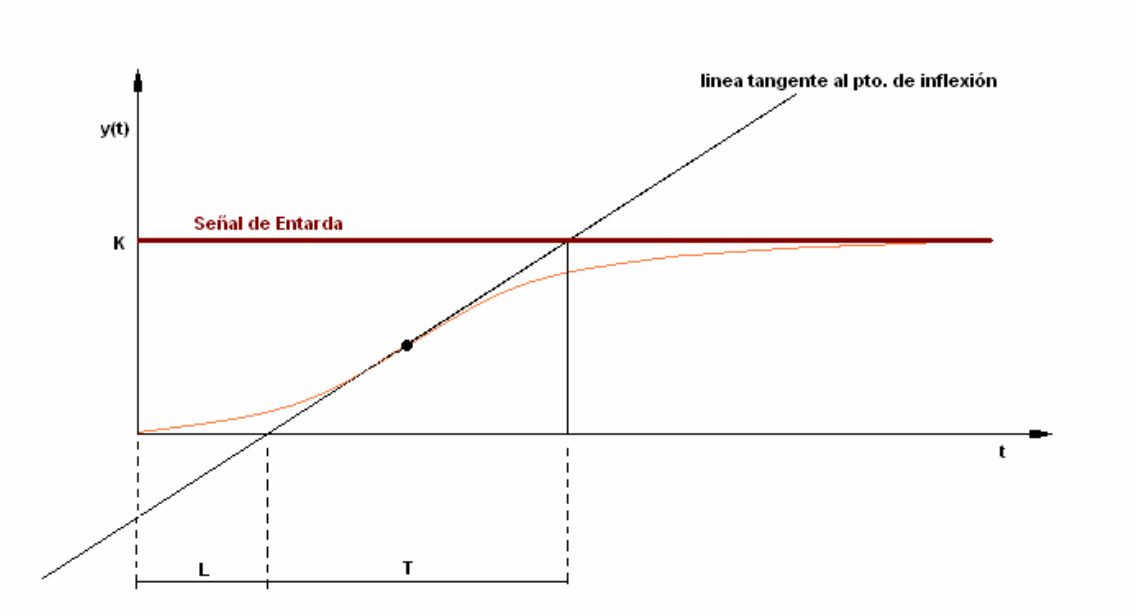
Modelo matemático. Función Transferencia

Sistemas térmicos: Los sistemas térmicos son aquellos que involucran transferencia de calor de una sustancia a la otra. Estos sistemas se analizan generalmente en términos de la resistencia térmica y capacitancia térmica. Pero para lograr modelos precisos es necesario usar modelos de sistemas distribuidos. Simplificaremos el análisis usando el método de Respuesta Transitoria para obtener la función transferencia de la planta [H(s)]. Usaremos como señal de prueba un escalón. El uso de señales de prueba se justifica porque existe una correlación

entre las señales de prueba usuales y las señales reales que el sistema deberá manejar.



Respuesta del sistema Horno a un escalón de entrada



Donde u(t)=K es la señal de prueba e y(t) es la señal de salida

Vemos que la curva de salida exhibe una forma de tipo "S". Dicha curva se caracteriza por dos parámetros: el tiempo de retardo **L** y la constante de tiempo **T**. El tiempo de retardo y la constante de tiempo se determinan trazando una recta tangente en el punto de inflexión de la curva y determinando las intersecciones entre esta recta con el eje de abscisas y el escalón de entrada. Aproximamos la función transferencia mediante un sistema de primer orden con un retardo de transporte:

$$H(s) = \frac{Y(s)}{U(s)} = \frac{e^{-Ls}}{Ts+1}$$

En la práctica obtuvimos los siguientes valores : L= 5seg.
T= 65seg.

$$H(s) = \frac{Y(s)}{U(s)} = \frac{e^{-5s}}{115s+1}$$

3.- Interfaz de Entrada

Medición de Temperatura: El sensor a usar será un LM35 de Nacional Semiconductors. La tensión de salida varía 10mV por cada ºC en forma **lineal**.. El rango de temperatura de funcionamiento óptimo es de -40 a 100ºC. Su funcionamiento se basa en la variación de la corriente que atraviesa una juntura PN a medida que varía la temperatura. La precisión es de 1ºC.

Conversión Analógica – Digital

Introducción

Si a una señal analógica queremos convertirla a una representación numérica existen dos dificultades. La primera es que una representación exacta requiere una cantidad infinita (y continua) de estados posibles, lo cual a su vez exigiría infinitos dígitos. La segunda dificultad está en que para obtener dicha representación se requiere que durante un tiempo la señal se mantenga invariable. La primera dificultad se resuelve por medio de la **cuantización**, es decir la aproximación mediante un nivel tomado de entre una cantidad finita de niveles. El proceso de cuantización será asimilable al redondeo o el truncamiento de un número de infinitas cifras decimales. La segunda dificultad se resuelve por medio del **muestreo** y la **retención**. Supondremos, por consiguiente, que la señal de entrada es constante durante el proceso de conversión.

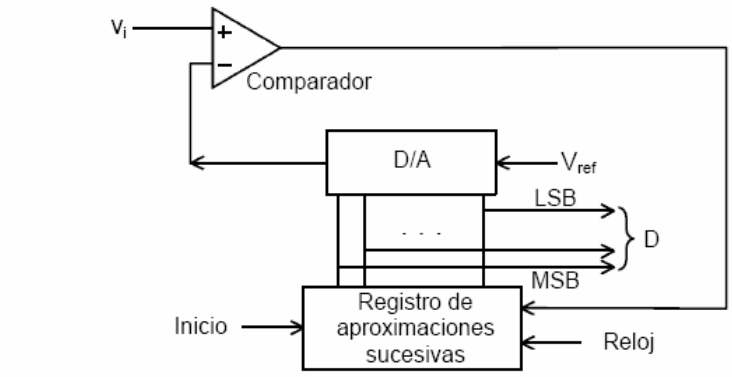
Existen varias técnicas de conversión analógica digital, que podrían clasificarse en dos grandes grupos: directas y realimentadas. Los conversores directos obtienen el dato digital por conteo o por comparación, mientras que los realimentados lo hacen mediante un conversor digital analógico que realimenta el dato digital generado por algún sistema lógico.

Métodos realimentados de conversión A/D

Operan generando digitalmente un código (de acuerdo con alguno de varios criterios), el cual se aplica como entrada digital a un conversor D/A. La salida de éste se compara con la entrada, y según el signo del error se incrementa o no el código.

Conversor de aproximaciones sucesivas

En la siguiente figura se muestra la estructura de un conversor analógico-digital de **aproximaciones sucesivas**, uno de los más utilizados en la actualidad pues permite una considerable velocidad de conversión y resolución alta a un bajo costo. La estructura es similar a la de los casos anteriores, pero reemplazando el contador por un **registro de aproximaciones sucesivas**.



El funcionamiento, ejemplificado en la figura 46, es el siguiente. Al dar una señal de inicio de la conversión, el registro aplica un 1 en el MSB (bit n) del conversor D/A y 0 en el resto de los bits. La salida del D/A ante dicho código (1000...0) se ubica en la mitad de la escala (V_{ref} / 2). Si $v_i \geq V_{ref}$ / 2, el MSB queda fijado definitivamente en 1.

Si, por el contrario, $v_i < V_{ref}$ / 2, el MSB vuelve a 0. En el paso siguiente, con independencia del valor fijado previamente para el MSB (bit n), el bit $n-1$ es llevado a 1. Nuevamente, si v_i supera el valor que ante ese código (x100...0) genera el conversor D/A, el 1 se conserva; de lo contrario, vuelve a 0. En el tercer paso se procede de igual manera: se lleva el bit $n-2$ a 1 y se compara la entrada con la salida del D/A ante ese código (xx10...0) y, según el resultado, se conserva el 1 o se lo lleva a 0. El proceso continúa hasta que se llega al LSB (bit 1). Una vez decidido el valor de éste, queda concluida la conversión.

Con este tipo de conversor el tiempo de conversión es de n ciclos de reloj, en lugar de 2^n (o aún mayor) como en los otros casos. Además de la velocidad, resulta importante el hecho de que en k ciclos de reloj ($k \leq n$) quedan garantizados los k bits más significativos. lo cual permite utilizar un mismo conversor con mayor velocidad si no se requiere la máxima resolución.

Es importante observar que, a diferencia del conversor de balance continuo o el flash, en este caso se requiere que la entrada se mantenga rigurosamente constante, de lo contrario podrían producirse errores muy groseros. En efecto, una vez que los bits más significativos han quedado fijados, ya no es posible cambiarlos hasta la próxima conversión, por lo cual el proceso continúa buscando la mejor aproximación que sea posible con los restantes bits. Por esta razón se requiere un *sample and hold* a la entrada.

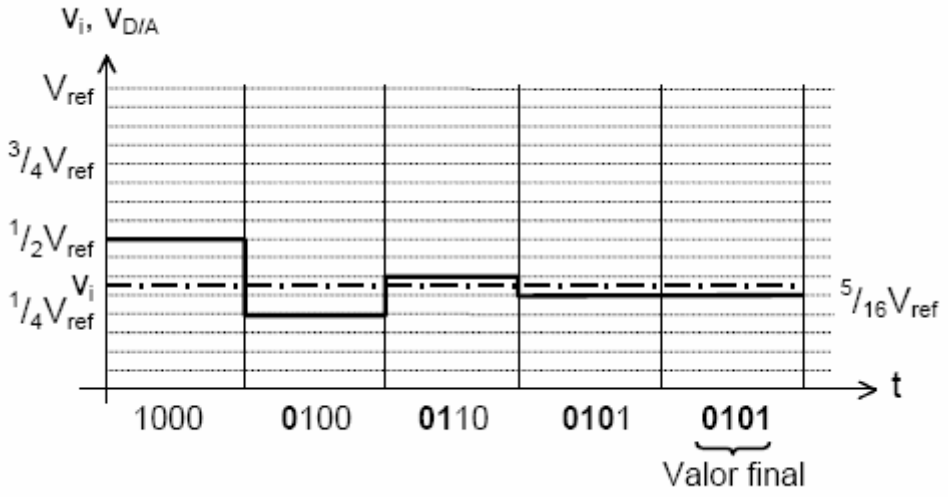
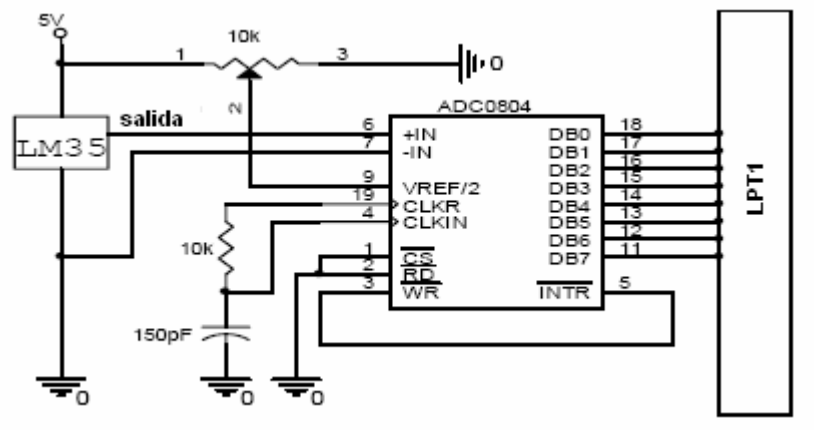


Figura 46. Ejemplo del proceso de acercamiento al valor final en un conversor analógico-digital de aproximaciones sucesivas de 4 bits. Los dígitos en **negrita** representan los que en cada etapa han quedado estabilizados

La técnica de conversión por aproximaciones sucesivas es utilizada por el CI ADC0804. Optamos por este CI debido a su bajo costo y suficiente precisión (8 bits) Usamos el conversor en modo "free-running" (pata INTR y WR interconectadas). En este modo el dispositivo está convirtiendo permanentemente. Luego conectamos la señal binaria a la PC a través del puerto paralelo. Debido a que el conversor análogo digital tiene una alta impedancia de entrada y nos permite ajustar la resolución, es posible conectar nuestro sensor directamente al conversor.

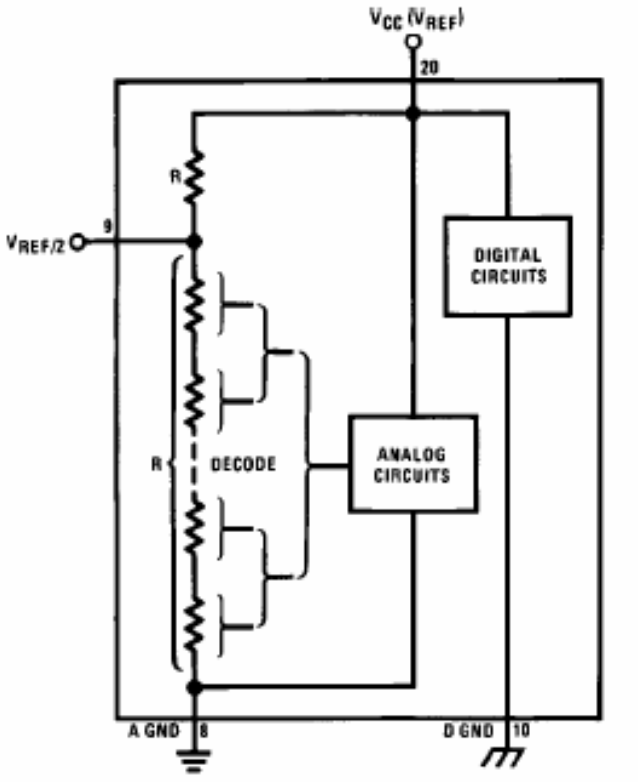


Calibración de la etapa de medición

Ajuste de Span

Para tener la mayor flexibilidad posible ha sido diseñado para trabajar con voltajes de referencia de 5V, 2,5V o un voltaje externo de referencia. Esto puede observarse en el esquema eléctrico que se muestra más abajo.
Notar que este voltaje puede ser la mitad de la tensión aplicada como Vcc o puede ser externamente forzado a través del pin Vref/2. Esto aumenta la flexibilidad del dispositivo. Por ejemplo, podría usarse esta opción si la tensión a convertir es reducida.

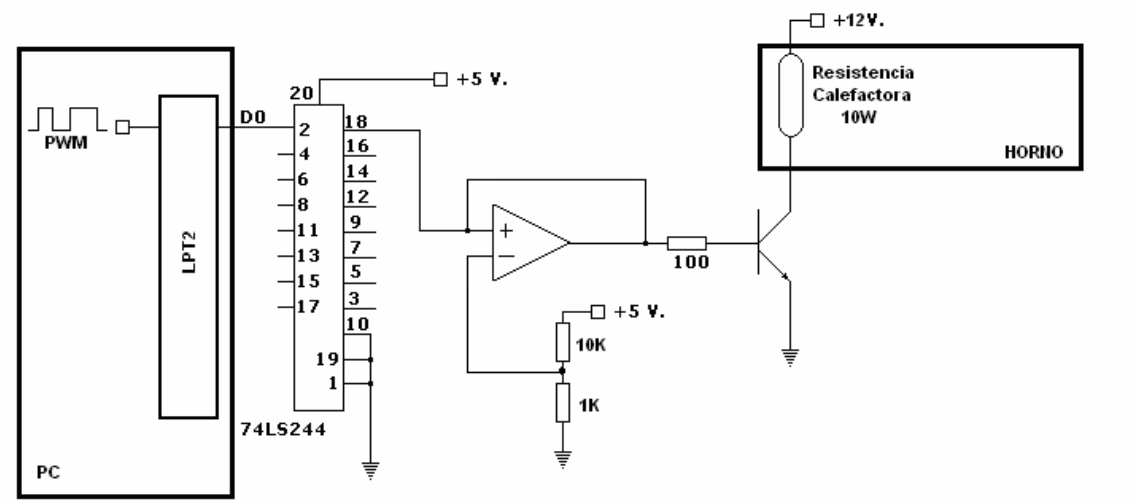
Ej: Si la tensión a convertir va desde 0.5 a 3.5V, en vez de 0 a 5V el span debería ser 3V. Con 0,5V aplicado al pin de corrección de offset (Vin -). De esta forma la tensión de referencia debería ser 3V/2=1.5V. El conversor ahora codificará una señal de forma que la salida será mínima ("00000000") para 0,5V de entrada y máxima ("11111111") para 3,5V de entrada.



4.- Interfaz de Salida

Esquema final

La señal de PWM (bit 0 del puerto usado como salida) es conectada por seguridad a una etapa separadora (buffer CI 74LS244). Luego a la señal se le da un nivel de potencia necesario como para que sature al transistor de salida (transistor de potencia NPN TIP33, encapsulado TO220) La señal es amplificada con un amplificador operacional (TL081)



El hecho de que el transistor de salida trabaje en las zonas de "corte" y "saturación" permite obtener un rendimiento óptimo, reduciendo el tamaño y potencia necesarios de la fuente de alimentación.

Esquema original e inconvenientes

Inicialmente usamos los 8 bits del puerto de salida (previo al esquema PWM). Esas 8 líneas se conectaban a un bufffer, luego se convertían en una señal analógica, por medio de un conversor digital - analógico DAC0808 y finalmente se amplificaban para luego excitar el transistor de salida que trabajaba polarizado en zona lineal. Dicho esquema era ineficiente desde el punto de vista del rendimiento.

Al pasar a un control PWM eliminamos el conversor. Esto produjo una serie de inconveniente. Primero, la señal de salida que entrega el DAC0808 es negativa, por lo que la configuración original del amplificador operacional era inversora. La cambiamos a no inversora. Luego al iniciar la prueba notamos que el transistor siempre quedaba en estado de saturación. Observando el circuito notamos que debíamos conectar la entrada inversora del operacional a un pequeño potencial positivo para forzarlo a saturación negativa al tener 0v en la entrada invasora (estado bajo de la señal de PWM). .

5.- Algoritmo de control

Sistemas de control en lazo abierto

Los sistemas de control en los cuales la salida no afecta la acción de control se denominan sistemas de control en lazo abierto. En estos sistemas no se mide la salida. A cada entrada de referencia le corresponde una condición operativa fija; como resultado la precisión del sistema depende de la calibración. Ante la presencia de perturbaciones un sistema de control en lazo abierto no realiza la tarea deseada.

Sistemas de control realimentados

Estos sistemas también se denominan sistemas de control en lazo cerrado. En un sistema de control en lazo cerrado, se alimenta al controlador con la señal de error de actuación, que es la diferencia entre la señal de entrada y la señal de realimentación (que puede ser la señal de salida misma o una función de esta) a fin de reducir el error y llevar la salida del sistema a un valor conveniente.

Teniendo en cuenta las perturbaciones externas a la que será sometido nuestro sistema (variación de temperatura ambiente, envejecimiento de componentes, variaciones en la fuente de alimentación, etc.) utilizaremos un control de lazo cerrado. Decidimos implementarlo directamente en la PC con el consiguiente ahorro de hardware y la flexibilidad de poder ajustar el control de una manera más rápida y práctica.

Acciones básicas de Control

Acción proporcional: Para un controlador con acción de control proporcional, la relación entre la salida del controlador u(t) y la señal de error e(t) es:

$$u(t) = Kp\ e(t)$$

donde Kp es la ganancia proporcional. Cualquiera sea la forma de operación el controlador proporcional es en esencia un amplificador con una ganancia ajustable.

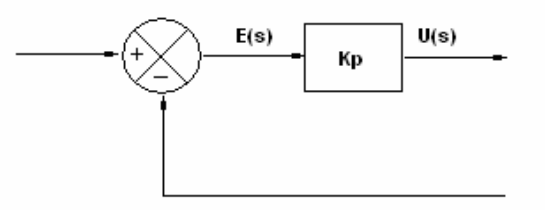


Diagrama en bloque de un controlador proporcional

Acción integral: En un controlador con acción de control integral, el valor de la salida del controlador u(t) se cambia a una razón proporcional a la señal de error e(t). Es decir:

$$\frac{du(t)}{dt} = Ki\ e(t)$$

$$u(t) = Ki \int_0^t e(t)\, dt$$

$$H(s) = \frac{Ki}{s}$$

$$H(s) = \frac{1}{Ti\, s}$$

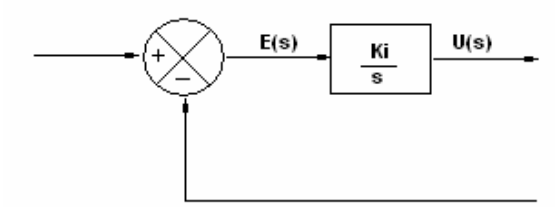


Diagrama en bloque de un controlador integral

En el control proporcional de una planta, cuya función de transferencia no posee un integrador, existirá un error en estado estable o desplazamiento (offset) en la respuesta para una entrada escalón. Tal offset se elimina si se incluye la acción integral en el controlador.

En el control integral de una planta la señal de control, que es la señal de salida a partir del controlador, es en todo momento el area bajo la curva de la señal de error hasta tal momento. La señal de control u(t) tiene un valor distinto de cero cuando la señal de error e(t) es cero. Esto es imposible en el caso de un control proporcional ya que un señal de control distinta de cero requiere una señal de error distinta de cero.

Aunque la acción integral elimina el offset o error en estado estable, puede conducir a una respuesta oscilatoria del sistema, decreciente lenta o incluso creciente.

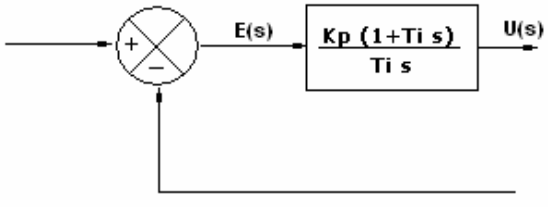
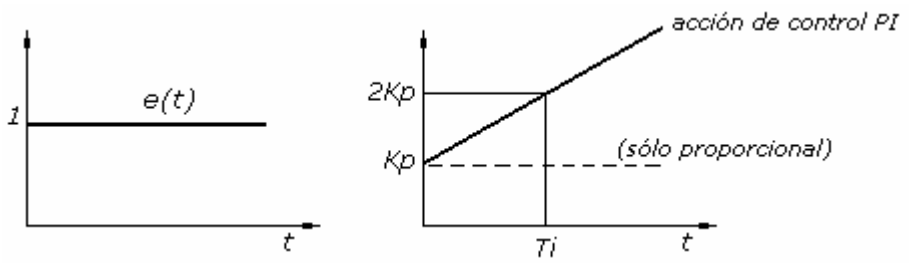
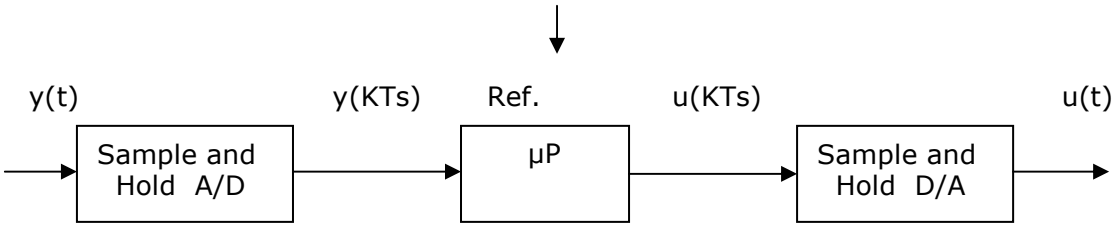


Diagrama en bloque de un controlador proporcional- integral



Arquitectura básica de un sistema de control digital (Control Digital Directo)



Siendo u(t) la señal de control e y(t) la variable de proceso.

Discretización de un Controlador PI

u(KTs) = Kp e(KTs) + [Kp Ts /Ti] e(KTs)

e(KTs)= [Ref.] – [y(KTs)]

Siendo Ts el tiempo de muestreo (determinado por el software pero siempre mayor o igual al tiempo de conversión), Kp la constante proporcional, Ti el tiempo integral y e(KTs) la diferencia entre el valor deseado (Ref.) y el valor medido (y) en el instante KTs.

El siguiente paso fue encontrar las constantes para el controlador. Aplicamos el método de Ziegler-Nichols.

Reglas de Ziegler-Nichols para sintonización de controladores PID

Primer método de Ziegler-Nichols: Si la planta no contiene integradores ni polos dominantes complejos conjugados la curva de respuesta al escalón unitario exhibirá como respuesta una rampa de tipo "S". En estos casos se aplicará el primer método de Ziegler-Nichols.

Como vimos, nuestra planta exhibe una respuesta de este tipo por lo que aplicaremos este método.

Ziegler y Nichols propusieron establecer los valores de Kp y Ti de acuerdo con la fórmula que aparece en la siguiente tabla:

Tipo de Controlador	Kp	Ti	Td
P	T/L	?	0
PI	0,9 T/L	L/0,3	0
PID	1,2 T/L	2L	0,5L

Donde L es el retardo de transporte y T es la constante de tiempo. Dichos valores fueron obtenidos al realizar el ensayo de respuesta al escalón (L=5 y T =115)

Por lo tanto: Kp = 13 y Ki = 16.5

Nota: Los métodos de Ziegler-Nichols parten de la base de tratar de obtener un sobrepaso máximo del 25% en la respuesta escalón. Si bien dicha premisa puede no satisfacer nuestro requerimiento proporcionan una conjetura razonada para los valores de los parámetros y ofrecen un punto inicial para una sintonización conveniente.

6.- Explicación general del programa

En el comienzo del programa se definen las variables a utilizar y se inicializan las variables destinadas al control (Kp, Ki, Ts) y se setea el valor del PWM y el error_ant en "0".

Posteriormente se utiliza una función que detecta los puertos de la PC y luego seteamos el puerto paralelo LPT2 como entrada poniendo a 1 el bit número 5 del puerto. El puerto LPT1 se utiliza como salida.

Mas adelante inicializamos el modo gráfico, que utilizaremos para displayar el comportamiento de la temperatura y del set point deseado.

Para graficar el set point y la temperatura es necesario escalar los valores leídos por el puerto a las dimensiones de la pantalla. También se grafican tanto los ejes como la escala de temperatura y se realiza un barrido de la pantalla cada Ts segundos, donde Ts es el tiempo de muestreo (realizado mediante un delay).

Por último calculamos el control PI, cuyo valor variara entre 0 y 50, y con este valor de PI se calculara el ciclo útil del PWM. El valor de PI será el tiempo en que la salida del PWM este en estado alto, para ello seteamos el bit número 1 del puerto LPT1 (de salida) en "1". El tiempo en que el PWM se encuentre en estado bajo, se calcula como el periodo de muestreo menos el tiempo en que el PWM esta en estado alto, seteando el bit 1 del puerto LPT1 en "0".

Por ultimo, se displaya en pantalla, los valores actuales de Kp, Ki y set point, pudiendo aumentar o disminuir los mismos mediante teclado. Para poder realizar esto fue necesario realizar una conversión de entero a cadena de caracteres para poder displayarlo.

7.- Bibliografía

Puerto Paralelo. Ing. Rubén M. Lozano. UTN. Facultad Regional Bs. As.

P. Childs, J. Greenwoods and C. Long, "Review of temperature measurements ", *Rev. Sc. Instrum.* **71**, 2959 (2000).

Katsuhiko Ogata, "Ingeniería de Control Moderna"

Centro de Referencia Nacional para Bancos de Leche humana – Instituto Fernandes Figueira / Fundación Oswaldo Cruz / Ministerio de Salud. João Aprígio Guerra de Almeida; Vander Guimarães & Franz Reis Novak

8.- Anexos

Listado de Fuentes

```
#include<stdio.h>
#include<dos.h>
#include<conio.h>
#include <graphics.h>
#include <stdlib.h>
#include<math.h>
#define LPT2 0X278
#define LPT1 0X378
#define LPT2_CONT 0X27A

void main(void)
{
unsigned int far *PTRDIR;
int x,y,I,cont,temp,error,error_ant,SP,set_p,escala,i,j;
int kp,ki,PWM,PWM_ANT,PI;
unsigned int ON,OFF;
char *set_point;
char *const_int;
char *const_prop;
char *ciclo_util;
char opcion;
float Ts=0.05;

//Valores iniciales del controlador

ki=1;
kp=10;
PWM=0;

// ESTA FUNCION BUSCA LOS PUERTOS PARALELOS DE LA PC//

PTRDIR=(unsigned int far *) 0x00000408;
for(I=0;I<3;I++)
{
 if (*PTRDIR==0) printf ("No se encontro puntero asignado a
LPTD%d",I+1);
 else
 printf("La direccion asignada a LPTD%d es 0x%p\n",I+1,*PTRDIR);
 PTRDIR++;
       }

      //ESTA FUNCION CONFIGURA EL LPT2 COMO ENTRADA//
      //ESCRIBIENDO EL BIT 5 DEL REGISTRO DE CONTROL CON UN "1" //

cont=inportb(LPT2_CONT);
cont=cont | 0x20;
outportb(LPT2_CONT,cont);

//Ingreso del Set Point

printf("\n Ingrese el Set Point:");
scanf("%d",&SP);

delay(100);
clrscr();

/* autodetecta placa de video e inicializa modo grafico */
```

```
int gdriver = DETECT, gmode, errorcode;
int xmax, ymax;
setgraphbufsize(8192);
initgraph(&gdriver, &gmode, "c:\\tc\\bgi");
errorcode = graphresult();
if (errorcode != grOk)
{  printf("Graphics error: %s\n", grapherrormsg(errorcode));
   printf("Press any key to halt:");
   getch();
   exit(1);
               }


//Condicion inicial del regulador PI

error_ant=0;


for (;;)
{
   for(x=40;x<460;x++)    //barrido horizontal
   {
      temp=inportb(LPT2);
      temp=(0.35*temp)-16;  //Leo y escalo temperatura
      y= (((-1)*1.607)*temp*3.6) + 450;
      putpixel(x,y,WHITE);                //salida a pantalla
      set_p= (((-1)*1.607)*SP*3.6) + 450;  //escalo Set Point
      putpixel(x,set_p,100);             //salida a pantalla


      error=SP-temp;                  //calculo de error
      PI = (error*kp) + (error_ant);  //regulador PI

      if (PI>50) PI=50;
      if (PI<0) PI=0;      // limito la salida del PI


     outportb(LPT1,0x01); //Salida PWM en "1"
      ON=PI;
      PWM_ANT=PWM;
     PWM=2*PI;
      OFF=50-ON;  //calculo del tiempo en "0" de la salida PWM
      delay(ON);

     outportb(LPT1,0x00);  //Salida PWM en "0"
      delay(OFF);

      error_ant=error_ant + (error*kp*Ts/ki);  //regulador PI

      if (error_ant>25) error_ant=25;   //limite integral
      if (error_ant<0) error_ant=0;


      itoa(SP,set_point,10);    //Conversion entero a char
      itoa(ki,const_int,10);     //para displayar
      itoa(kp,const_prop,10);
      itoa(PWM,ciclo_util,10);

      //Grafico los ejes cartesianos y coloco etiquetas//

      setbkcolor(GREEN);  //elijo color de fondo
      setcolor(BLUE);        //elijo color actual
```

```
      rectangle(460,20,620,450);  //dibujo rectangulo
      line(40,450 , 450, 450);  //dibujo linea
      line(40,450, 40,40 );
      outtextxy(400,460,"Tiempo"); //escribo texto
      outtextxy(2,30,"Temp.");
      outtextxy(8,392,"10");
      outtextxy(8,334,"20");
      outtextxy(8,273,"30");
      outtextxy(8,218,"40");
      outtextxy(8,160,"50");
      outtextxy(8,102,"60");
      outtextxy(8,45,"70");
      outtextxy(480,70,"SET POINT:");
      outtextxy(570,70,set_point);
      outtextxy(480,100,"Ki(x1000):");
      outtextxy(570,100,const_int);
      outtextxy(480,130,"Kp:");
      outtextxy(570,130,const_prop);
      outtextxy(480,160,"PWM(%):");
      outtextxy(570,160,ciclo_util);
      outtextxy(470,200,"Inc. Set Point: +");
      outtextxy(470,230,"Dec. Set Point: -");
      outtextxy(470,260,"Inc. Ki:        u");
      outtextxy(470,290,"Dec. Ki:        j");
      outtextxy(470,320,"Inc. Kp:        o");
      outtextxy(470,350,"Dec. Kp:        l");
      outtextxy(470,380,"Salir:          s");

      if (PWM!=PWM_ANT)  //refresco de pantalla
      {
       for(i=139;i<=170;i++)
      {for(j=565;j<=610;j++) putpixel(j,i,GREEN);   }}

// eleccion desde teclado del operador

      if (kbhit()) {    for(i=65;i<=170;i++)
                                {for(j=565;j<=610;j++)
putpixel(j,i,GREEN);   }
                           opcion=getch();
                            if (opcion=='+') SP++;
                            if (opcion=='-') SP--;
                            if (opcion=='o') ki++;
                            if (opcion=='l') ki--;
                            if (opcion=='u') kp++;
                            if (opcion=='j') kp--;
                            if (opcion=='s') return;

                          /*   switch(opcion)
                             {case 's': return;
                             flushall();
                             case '+': SP++;
                             case '-': SP--;
                             case 'o': ki++;
                             case 'l': ki--;
                             case 'u': kp++;
                             case 'j': kp--;

                                                  } */
                                               }

}
 cleardevice();
}
```

Pantalla de control de la aplicación

Placa de adquisición y control