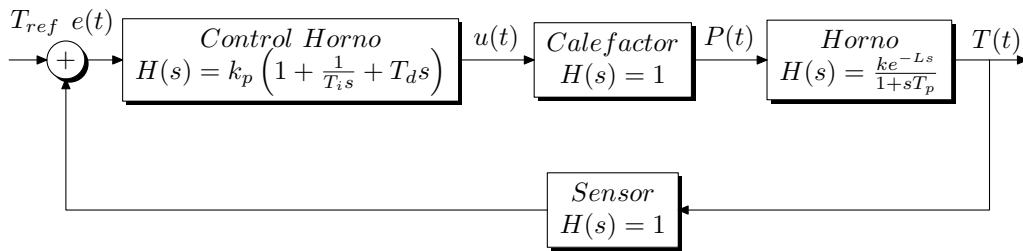


## Práctica 5

# Sistemas de control dinámico II

En esta práctica se implementará un sistema de control dinámico del horno analizado en la práctica nro. 4 pág. 1. El controlador debe seguir un esquema realimentado PID como el que se muestra en la figura 5.1.



**Fig. 5.1:** Sistema de control de un horno.

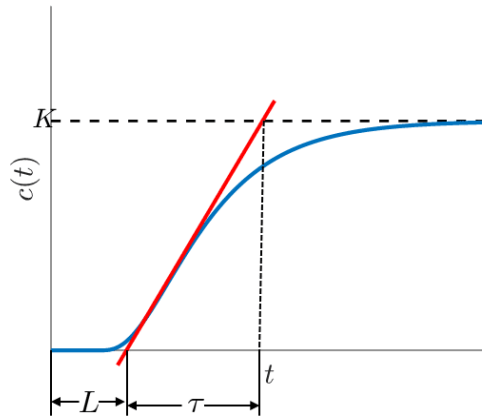
La ecuación de control, una vez discretizada, toma la forma:

$$u(n) = k_p \left[ e(n) + \frac{1}{T_i} \sum_{k=1}^n e(k) T_s + T_d \frac{e(n) - e(n-1)}{T_s} \right] \quad (5.1)$$

donde los coeficiente  $k_p$ ,  $T_i$  y  $T_d$  se calculan según la regla de Ziegler–Nichols <sup>1</sup> y  $T_s$  es el periodo de muestreo.

### 5.1 Tareas a realizar

1. Utilizando el método 1 de Ziegler–Nichols calcular los valores de las constantes  $k_p$ ,  $T_i$  y  $T_d$ . Para ello, se obtendrá una gráfica como la siguiente introduciendo una potencia cualquiera al horno y sacando la temperatura. Después se calcularán, de forma aproximada, los parámetros  $K$ ,  $L$  y  $\tau$ .



<sup>1</sup>[https://en.wikipedia.org/wiki/Ziegler-Nichols\\_method](https://en.wikipedia.org/wiki/Ziegler-Nichols_method)

Para calcular las constantes, usar la siguiente tabla:

Controlador	$K_p$	$\tau_i$	$\tau_d$
<b>P</b>	$\frac{\tau}{KL}$	$\infty$	0
<b>PI</b>	$0.9 \frac{\tau}{KL}$	$\frac{L}{0.3}$	0
<b>PID</b>	$1.2 \frac{\tau}{KL}$	$2L$	$0.5L$

2. Escribir un paquete de nombre PID que se ajuste a la especificación siguiente:

**Fichero 5.1:** Especificación del paquete PID (pid.ads)

```

1 generic
2   type Real is digits <>;
3   type Entrada is digits <>;
4   type Salida is digits <>;
5 package PID is
6   type Controlador is limited private;
7
8   procedure Programar (el_Controlador: in out Controlador;
9                       Kp, Ki, Kd: Real);
10  procedure Controlar(con_el_Controlador: in out Controlador;
11                    R, C: Entrada;
12                    U: out Salida);
13 private
14   type Controlador is record
15     -- Parámetros del controlador
16     Kp, Ki, Kd: Real;
17     -- Estado del controlador
18     S_Anterior : Real := 0.0; -- s(n-1) Condiciones de
19     Error_Anterior: Real := 0.0; -- e(n-1) reposo inicial
20   end record;
21 end PID;
```

La operación **Programar** se encarga de inicializar un objeto de tipo **Controlador**. La operación **Controlar** realiza los cálculos para implementar con posterioridad un ciclo de control (recordamos que  $e(n) = R(n) - C(n)$ ).

3. Usando el paquete PID y los paquetes **Sensor** y **Calefactor** de la práctica anterior, escribir un programa de nombre **Principal** donde se implemente el ciclo de control del horno. Este programa pedirá una temperatura de referencia y controlará el horno durante 10 minutos.

El ciclo de control tiene la forma siguiente:

```

loop -- Bucle con periodicidad Ts
  esperar al siguiente instante de muestreo
  leer T en el sensor
  e := T-Tref
  calcular u(t)=P(t)=PID(e)
  escribir P en el calefactor
  escribir T en pantalla
```

end loop;

El diagrama de componentes de la aplicación se muestra en la figura 5.2. Recordamos que los paquetes Horno y Retardadores no los tiene que el escribir el alumno.

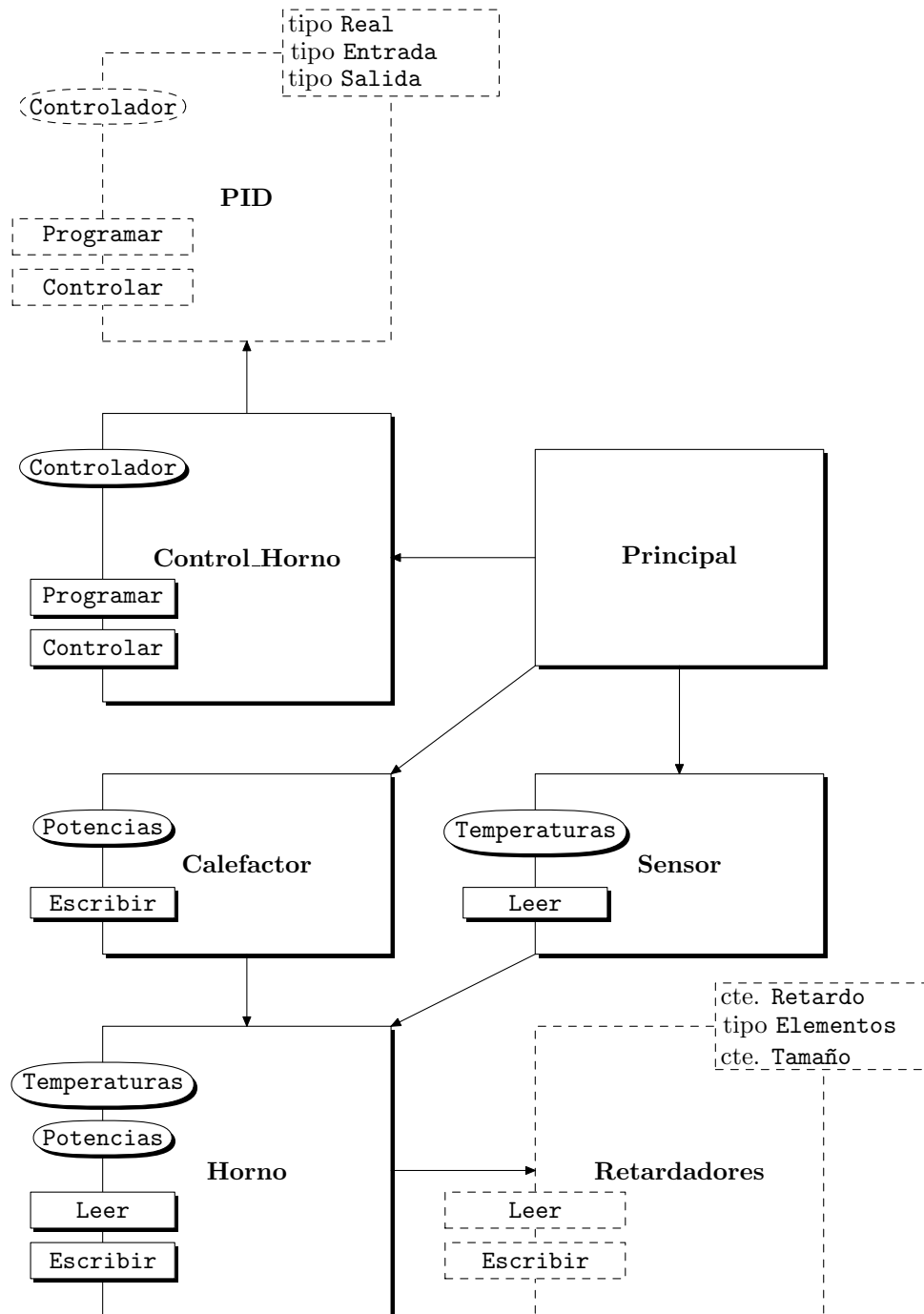
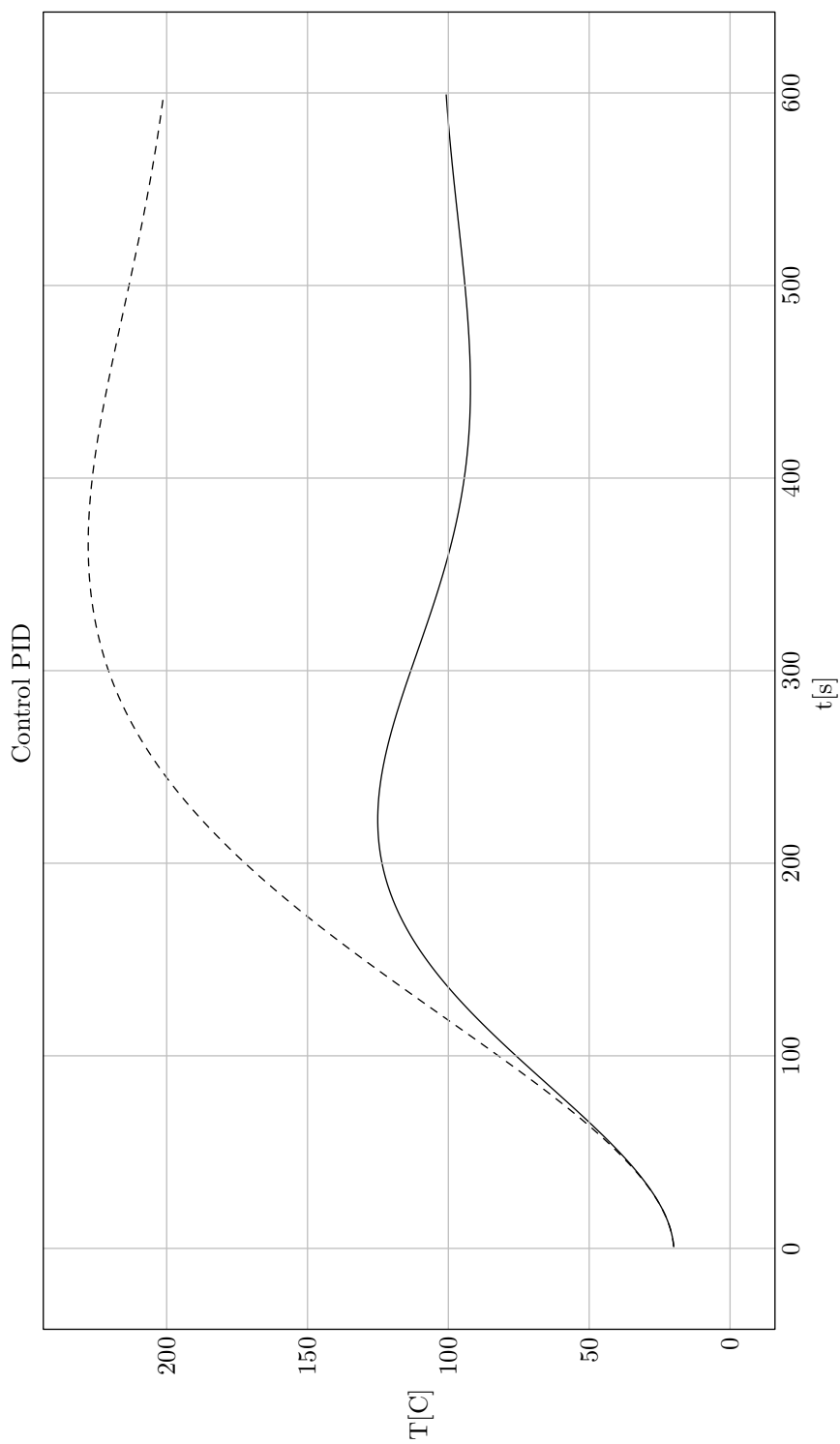


Fig. 5.2: Diagrama de componentes del controlador del horno.

4. Ejecutar el programa considerando las temperaturas de referencia 100 C y 200 C. A partir de los datos obtenidos realizar una representación gráfica de la evolución de la temperatura. Para obtener esta representación se puede utilizar cualquier herramienta de presentación de datos, por ejemplo Matlab. En la figura 5.3 podemos ver un ejemplo de los resultados que se obtienen. En [https://castilloinformatica.es/wiki/index.php?title=Archivo:PID\\_Compensation\\_Animated.gif](https://castilloinformatica.es/wiki/index.php?title=Archivo:PID_Compensation_Animated.gif) puedes ver un ejemplo de cómo afectan las constantes a la respuesta del sistema.



**Fig. 5.3:** Resultado de la ejecución del controlador para dos temperaturas de referencia  $T_{ref} = 100\text{ }^{\circ}\text{C}$  (trazo continuo) y  $T_{ref} = 200\text{ }^{\circ}\text{C}$  (trazo discontinuo).