

CONTROL DIGITAL PID

Diseño con uP y uC

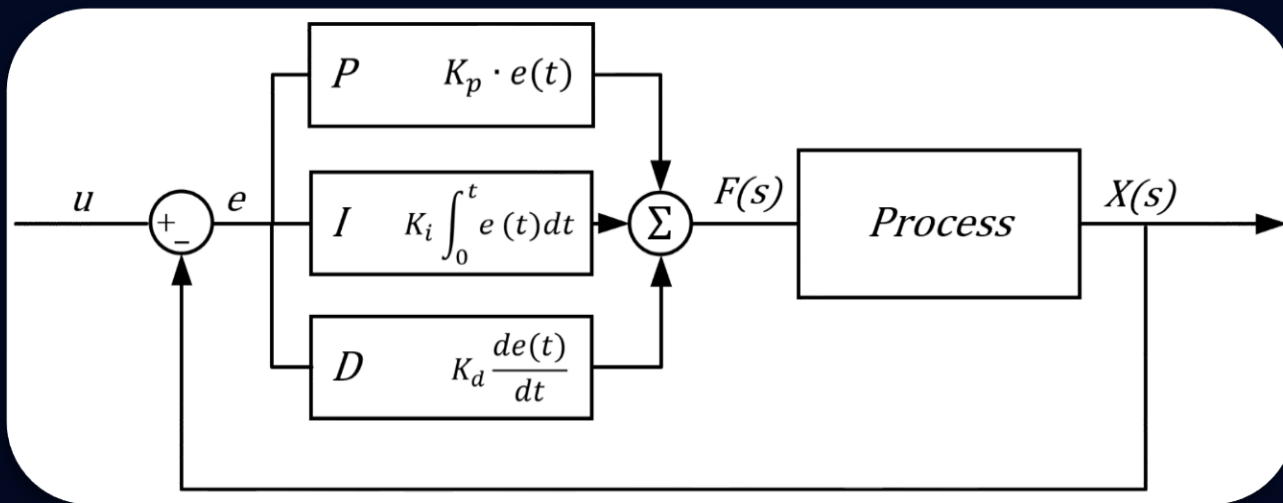
Diego Andrés García Díaz
2195533

Universidad Industrial de Santander

2024-1

¿Qué es el Control PID?

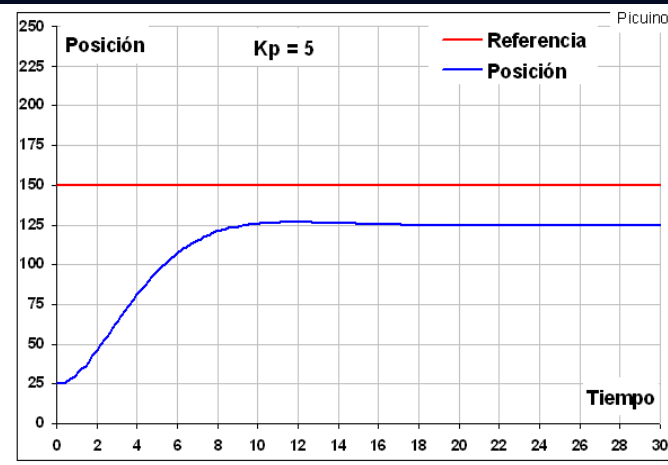
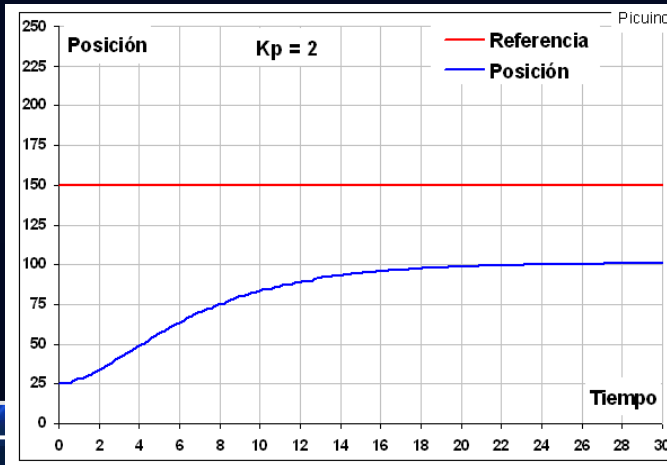
Un controlador o regulador PID es un dispositivo que permite controlar un sistema en lazo cerrado para que alcance el estado de salida deseado. El controlador PID está compuesto de tres elementos que proporcionan una acción Proporcional, Integral y Derivativa. Es un algoritmo que se usa para controlar sistemas con *I/O* analógicas.

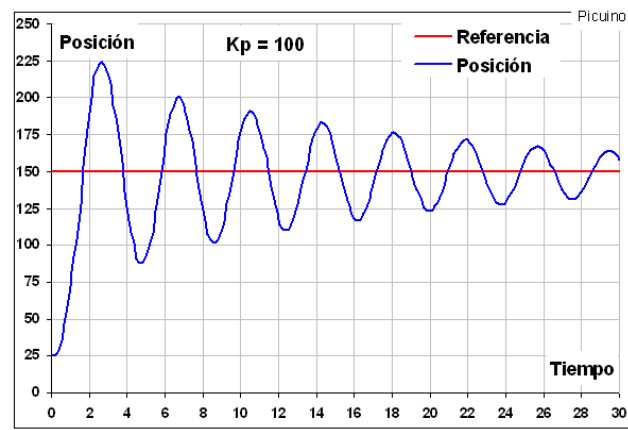
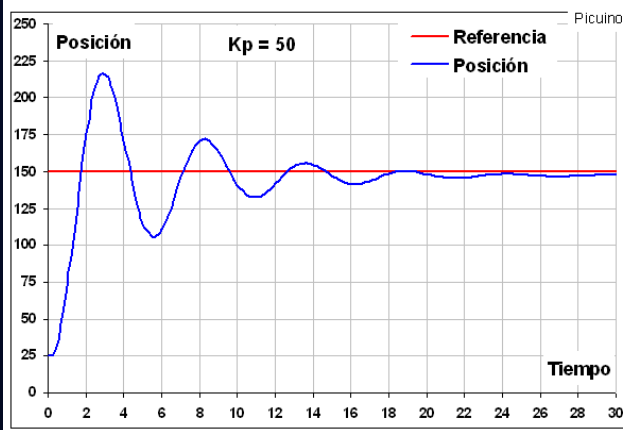
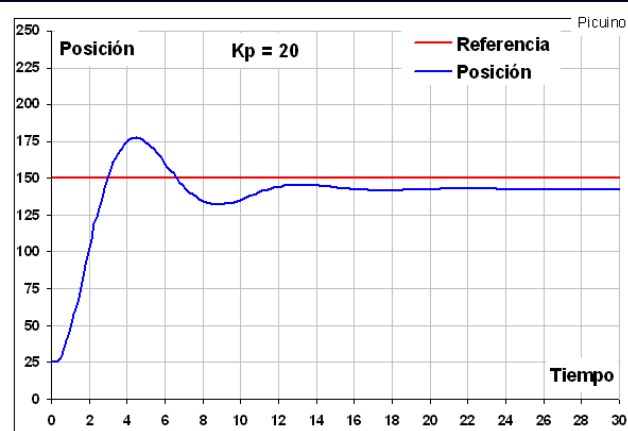
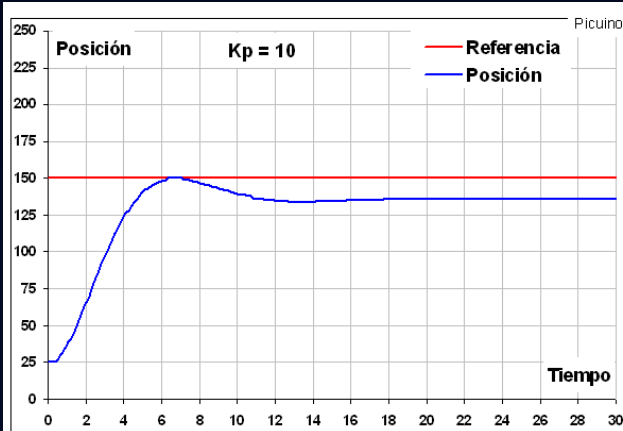


¿Cómo funciona el Control PID?

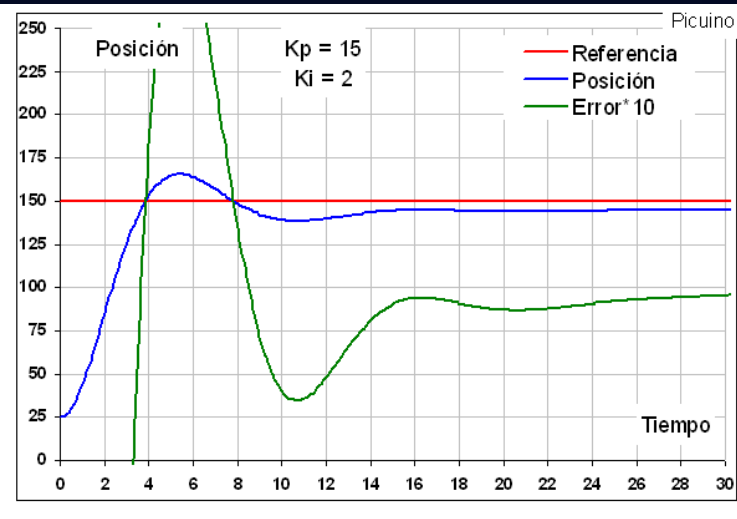
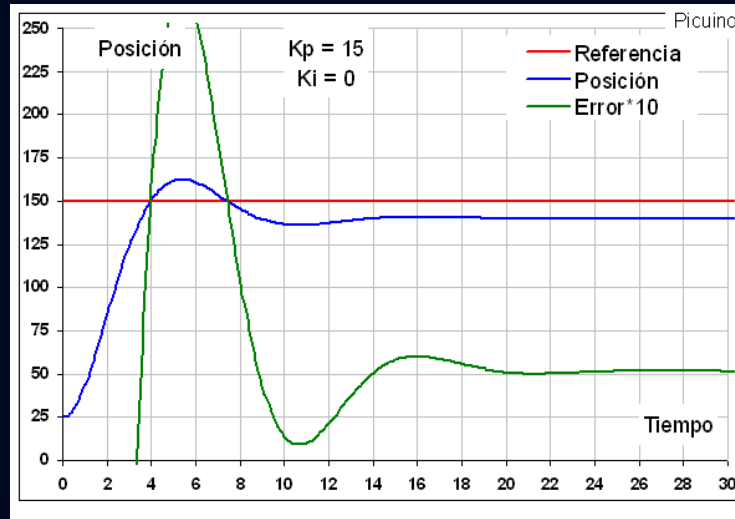
El control PID utiliza tres términos:

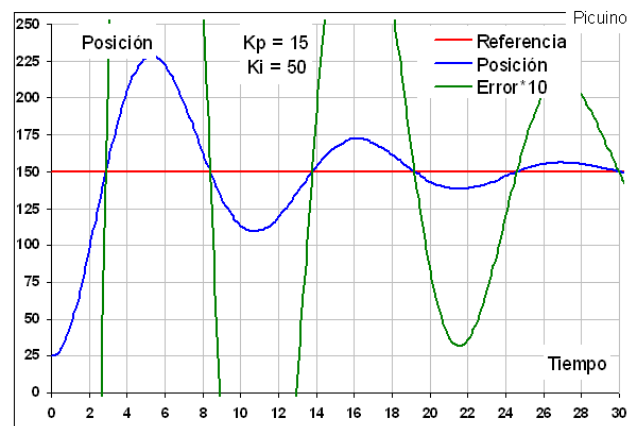
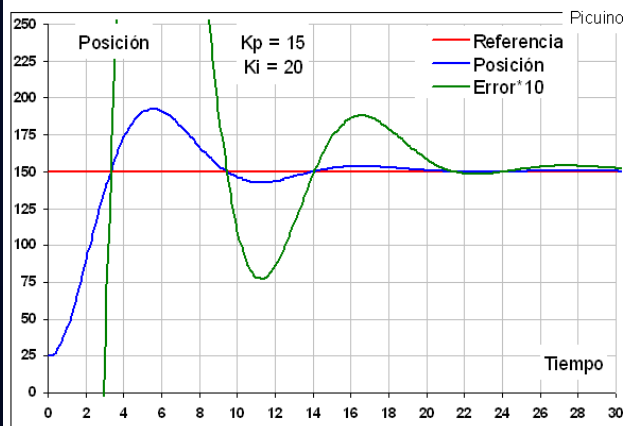
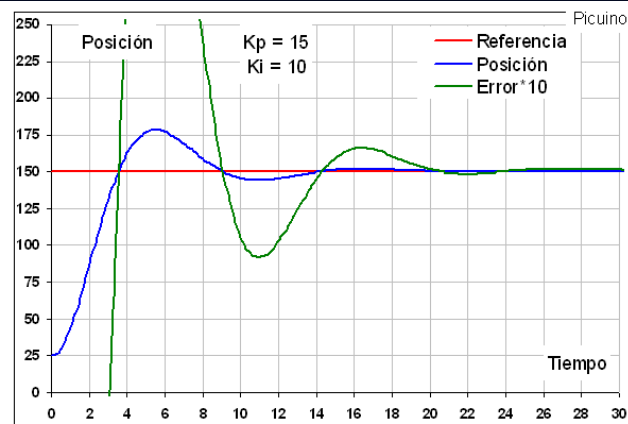
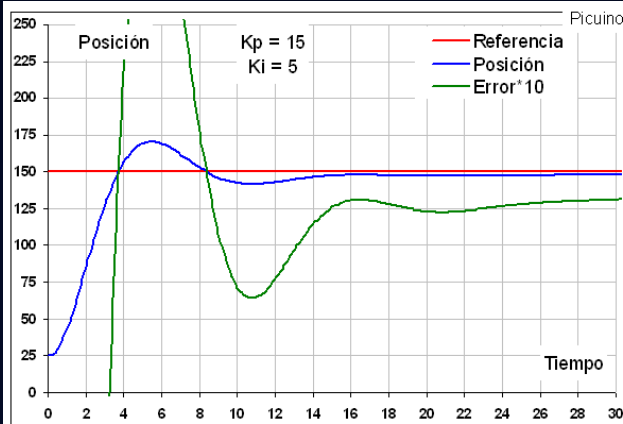
1. **Control Proporcional (P):** Responde de manera proporcional a la señal de error $e(t)$, donde internamente multiplica la señal de **error** $e(t)$ por una constante K_p que determina la cantidad de acción proporcional que tendrá el controlador. Si se aumenta K_p , se aumenta la velocidad del sistema, se disminuye el error del sistema en régimen permanente y aumenta la inestabilidad del sistema.



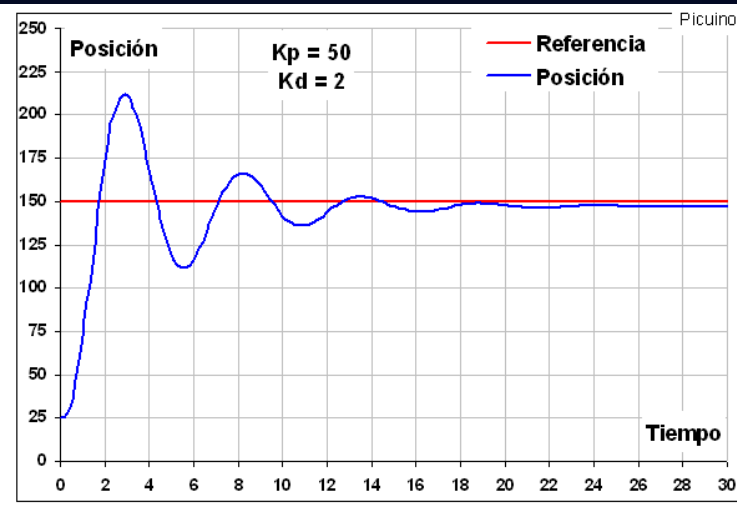
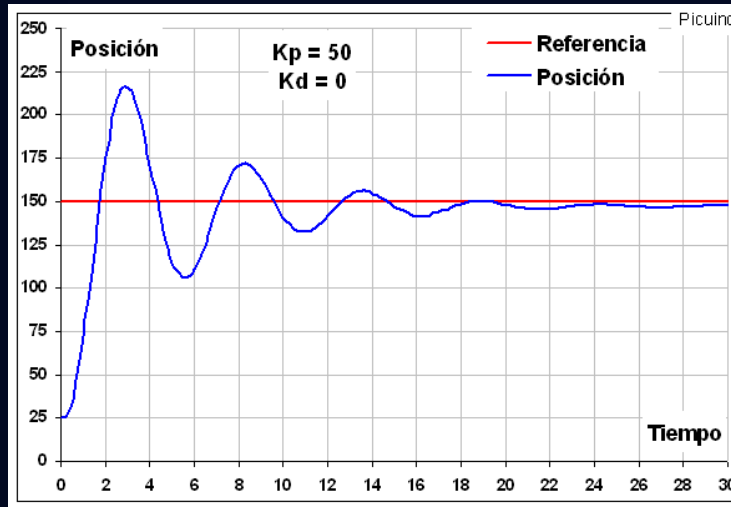


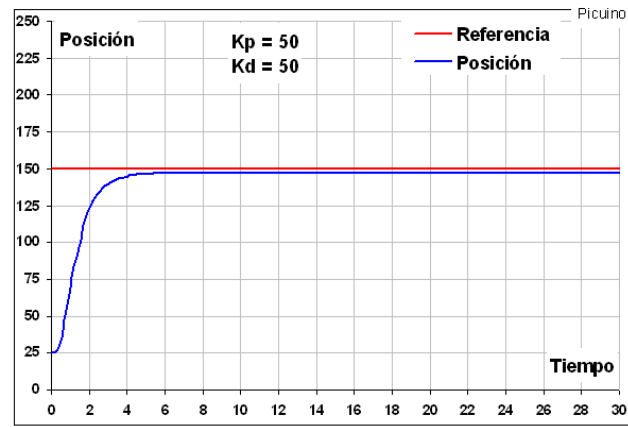
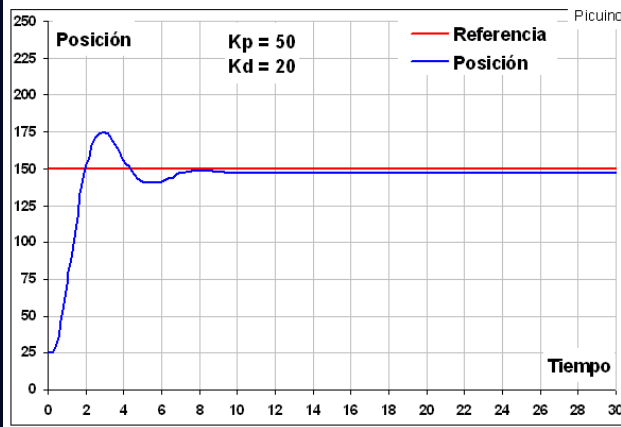
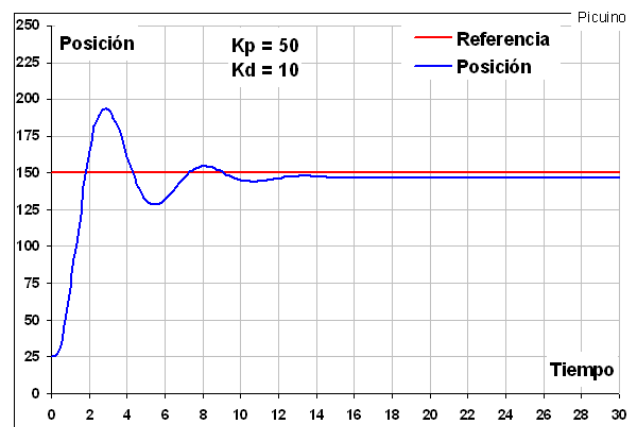
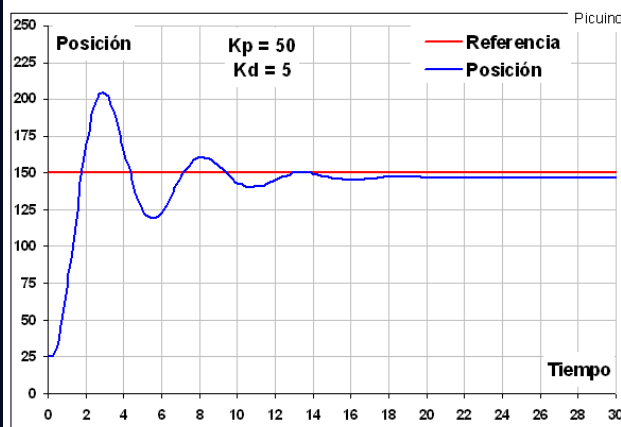
2. Control Integral (I): Se integra la señal de **error $e(t)$** y se multiplica por una constante **K_i** , donde la integral corresponde a la acumulación de la señal de error, donde se suman los diferentes errores obtenidos de tal forma que la acción integral se hace mayor y se reduce el error del sistema en régimen permanente. Si se aumenta **K_i** , se disminuye el error del sistema en régimen permanente, aumenta la inestabilidad del sistema y aumenta un poco la velocidad del sistema.

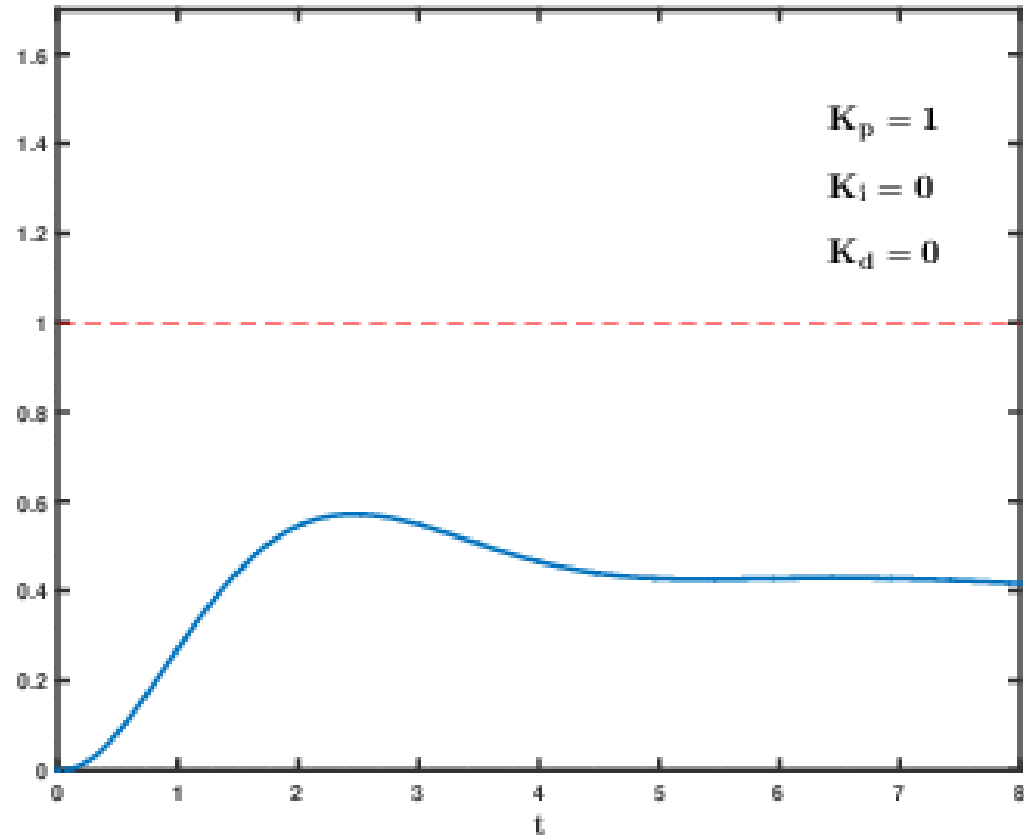




3. Control Derivativo (D): Se deriva la señal de *error* $e(t)$ y se multiplica por una constante K_d , el término derivativo se basa en la tasa de cambio del *error* $e(t)$ con respecto al tiempo y predice cómo cambia el *error* en el momento actual y usa esta información para modificar la salida suavizando la respuesta del sistema. Si se aumenta K_d , se aumenta la estabilidad del sistema controlado, se disminuye un poco la velocidad del sistema, el error régimen permanente será el mismo (no cambia).



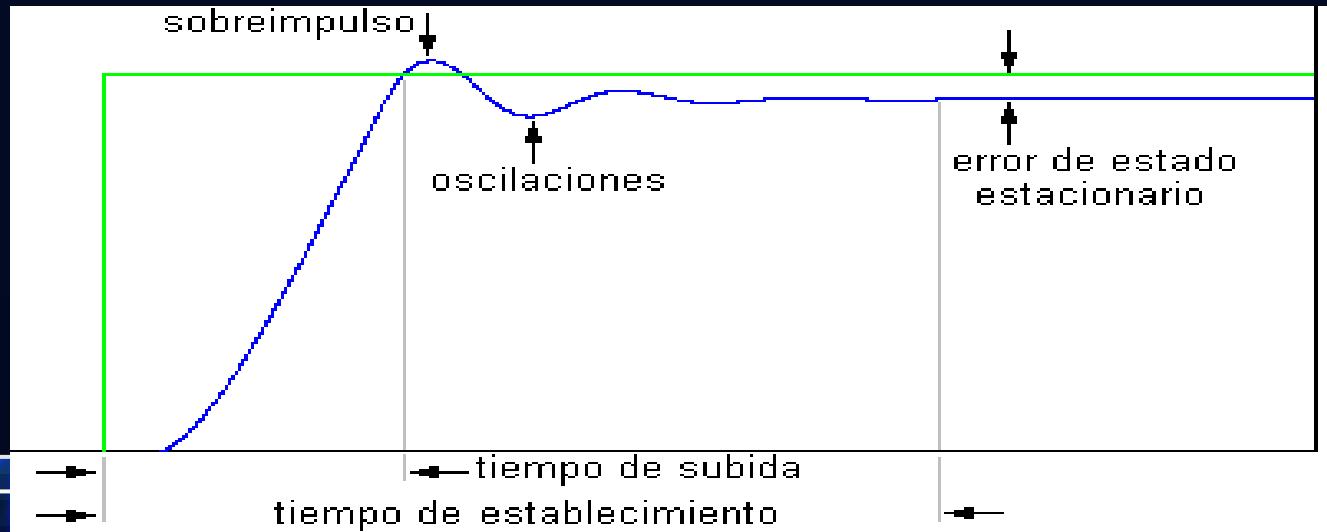




Consideraciones Importantes

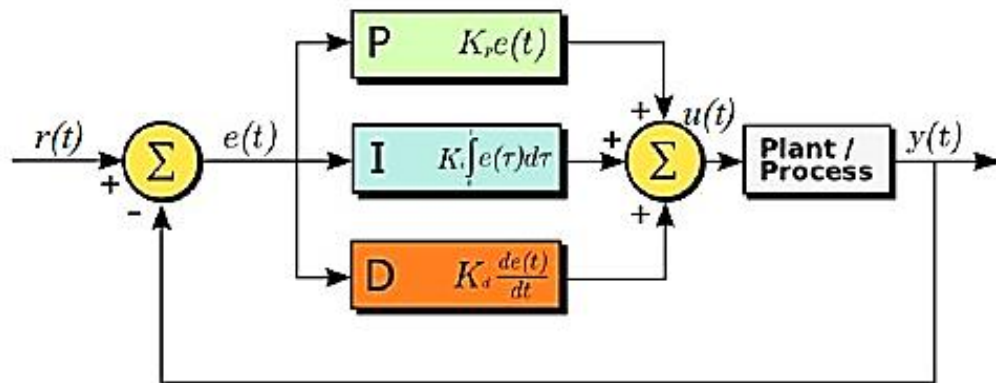
Es importante tener en cuenta que se puede sintonizar (ajustar) los diferentes parámetros (K_p , K_i , K_d) de diferentes maneras, manual, método de Ziegler-Nichols, método de Cohen-Coon, método de asignación de ceros y polos. El control PID estará sintonizado si los parámetros tienen los valores correctos para obtener la salida esperada. En un sistema ideal se necesita que sea estable, veloz y con ningún *error*.

En los sistemas reales existen limitaciones que reducen la capacidad del controlador para conseguir la respuesta requerida.



Fórmulas Principales

- $u(t)$ = señal de control.
- $e(t)$ = señal de error.
- K_p, K_i, K_d = parámetros del controlador PID.



$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt}$$



$$\frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} + K_d s$$

Proportional

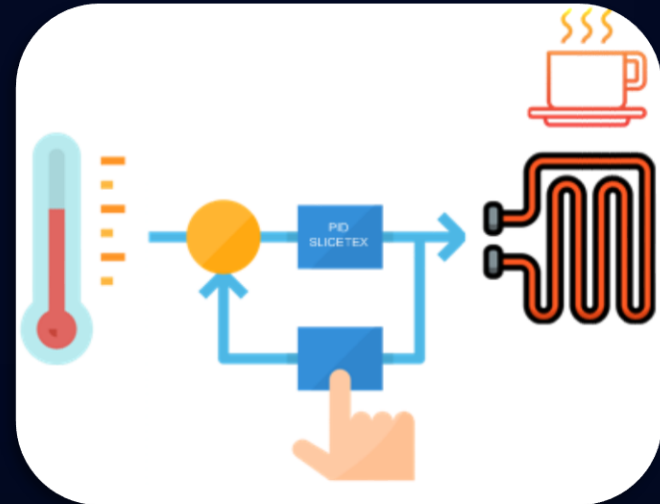
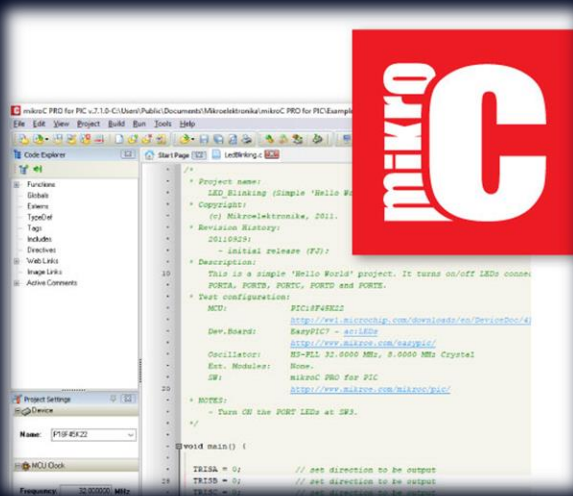
Integral

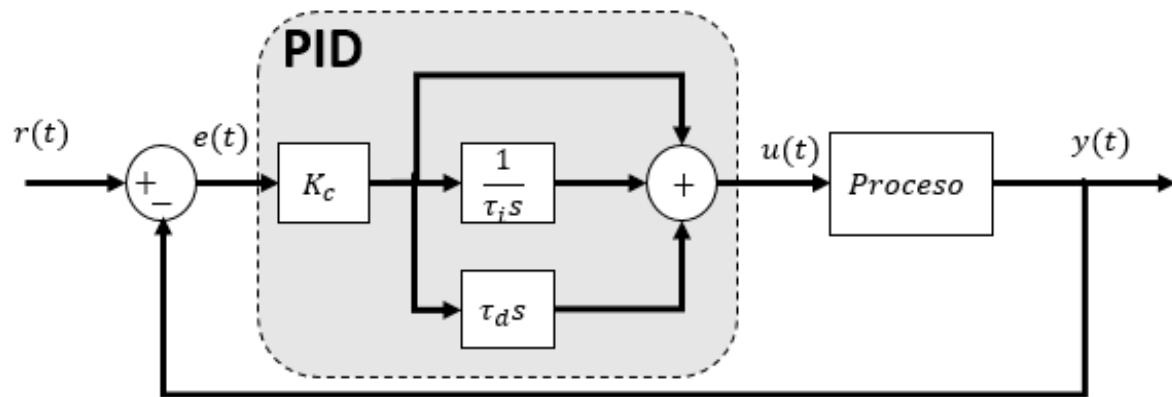
Derivative

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t)$$

Control PID usando uC PIC

El Control Digital PID usando PIC implica la implementación de algoritmos en los microcontroladores, esto permite controlar sistemas embebidos, como por ejemplo robots, sistemas de automatización, procesos, controles de temperatura, controles de giro de motores, entre muchos otros.





Control PID Discreto



$$u(k) = u(k-1) + q_0 e(k) + q_1 e(k-1) + q_2 e(k-2)$$

donde:

$$q_0 = k_c \left(1 + \frac{T_s}{2\tau_i} + \frac{\tau_d}{T_s} \right)$$

$$q_1 = -k_c \left(1 - \frac{T_s}{2\tau_i} + \frac{2\tau_d}{T_s} \right)$$

$$q_2 = \frac{k_c \tau_d}{T_s}$$

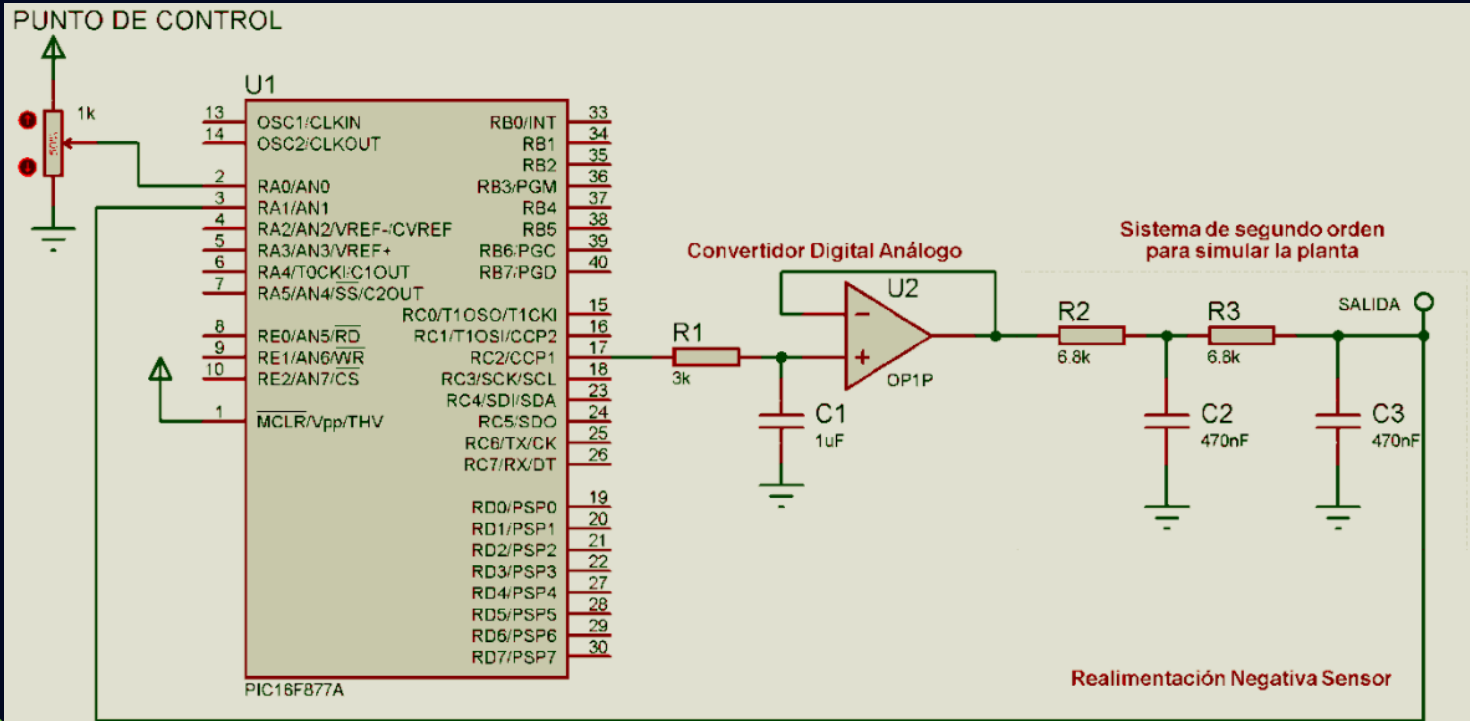
$$C(z) = \frac{U(z)}{E(z)} = \frac{q_0 + q_1 z^{-1} + q_2 z^{-2}}{1 - z^{-1}}$$



https://youtu.be/C5q1_TlybGs?si=pK9SImkKh20-xKe7



Ejemplo del libro:



```

1 //Declaración de constantes y variables.
2 const float Kp=1.08, Ki=52.42718447, Kd=0.005562, Fs = 152.5878906;
3 //Declaración de coeficientes de integración y derivación.
4 const float Fi=Ki/(2.0*Fs), Fd=2.0*Fs*Kd;
5
6 float SENSOR, PUNTO_DE_CONTROL=127.0, YN=0.0;
7 int ADQUI;
8 unsigned short SALIDA;
9
10 float e0=0.0, e1=0.0, yi0=0.0, yi1=0.0, yd0=0.0, yd1=0.0, ypid=0.0;
11
12 //Función de interrupciones para el Timer 0.
13 void interrupt()
14 {
15     if( INTCON.F2 )// 6,5536ms :: 152,5878906 Hz
16     {
17         //Adquisición de la variable controlada.
18         SENSOR = (float)((ADC_Read(1)>>2)&0xFF);
19         //Adquisición del punto de control.
20         PUNTO_DE_CONTROL = (float)((ADC_Read(0)>>2)&0xFF);
21         //Calculo del nivel de error.
22         e0 = PUNTO_DE_CONTROL - SENSOR;
23         //Ecuación en diferencias.
24         //Ecuación integral.
25         yi0=Fi*(e0+e1)+yi1;
26         //Ecuación derivativa.
27         yd0=Fd*(e0-e1)-yd1;
28         //Resultado PID.
29         ypid=Kp*e0+yi0+yd0;
30         //Ajuste y corrección de la SALIDA Y(n)
31         //delimitada por los límites 0 y 255.
32         YN += ypid;
33         if(YN>255.0)YN=255.0;

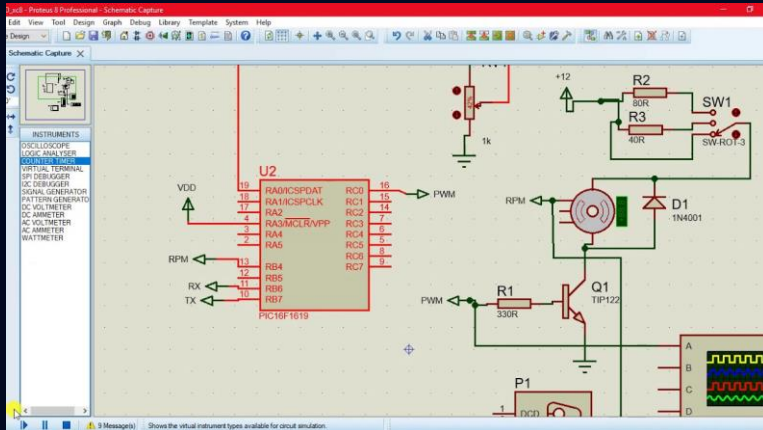
```

```

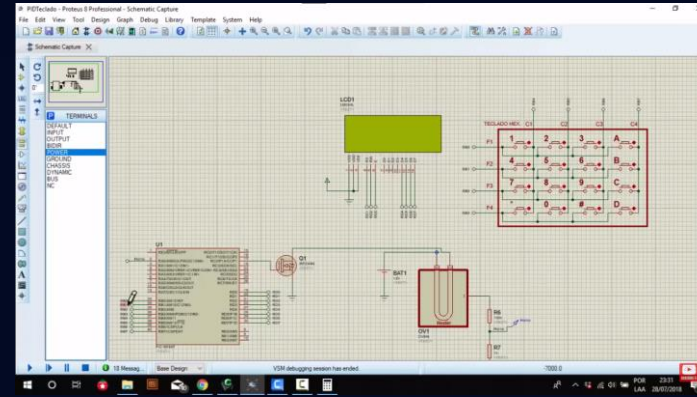
33         if(YN>255.0)YN=255.0;
34         if(YN<0.0)YN=0.0;
35         SALIDA = (unsigned short)(YN);
36         PWM1_Set_Duty(SALIDA);
37         //Actualización de muestras.
38         e1=e0;
39         yi1=yi0;
40         yd1=yd0;
41         INTCON.F2=0;
42     }
43 }
44
45 void main( void )
46 {
47     //Configuración del modulo PWM.
48     PWM1_Init(10000);
49     PWM1_Start();
50     //Configuración de la interrupción de Timer 0.
51     //a 6,5536ms
52     OPTION_REG = 0b00000110;
53     INTCON = 0b10100000;
54     while( 1 ) //Bucle infinito.
55     {
56     }
57 }
58

```

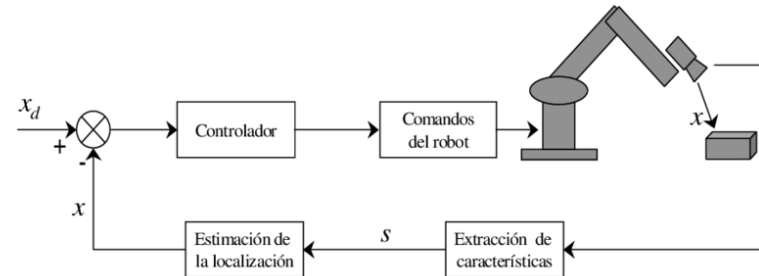
Otros ejemplos de Control PID:



<https://www.youtube.com/watch?app=desktop&v=EBw3giRaSM8>



https://youtu.be/dj51tIC76no?si=6nDj0Tadvg4Ygp_f



Referencias

- J. R. Clavijo Mendoza, “14.9 Control digital PID”, en *Diseño y simulación de sistemas microcontrolados en lenguaje C*. Bogotá DC: Clavijo Mendoza, Juan Ricardo, 2011, pp. 238–242.
- “Controlador PID - Control Automático - Piduino”. Página principal - Piduino. Accedido el 31 de marzo de 2024. [En línea]. Disponible: <https://www.piduino.com/es/control-pid.html>
- S. A. Castaño Giraldo. “Control-PID-Discreto”. Control Automático Educación. Accedido el 3 de abril de 2024. [En línea]. Disponible: <https://controlautomaticoeducacion.com/control-realimentado/controladores-pid-discreto/>
- “¿Qué es un controlador PID? - Control, ajuste y cálculo de variables | Qué es”. Qué es. Accedido el 1 de abril de 2024. [En línea]. Disponible: <https://quees.com/controlador-pid/>
- “Control PID con PIC: Microcontrolador [Paso a Paso en CCS C]”. Control Automático Educación. Accedido el 2 de abril de 2024. [En línea]. Disponible: https://controlautomaticoeducacion.com/microcontroladores-pic/control-pid/amp/#Control_de_Temperatura_PID_con_Microcontrolador_PIC
- “PID ¿Cómo sintonizar un lazo PID? - Control Real Español”. Control Real Español. Accedido el 3 de abril de 2024. [En línea]. Disponible: <https://controlreal.com/es/pid-como-sintonizar-un-lazo-pid/>
- “Controlador PD – Proporcional Diferencial – Sistemas de control”. dademuchconnection. Accedido el 3 de abril de 2024. [En línea]. Disponible: <https://dademuchconnection.wordpress.com/2019/05/22/controlador-pd-proporcional-diferencial-sistemas-de-control/>
- “PIC16F877A Datasheet(PDF)”. ALLDATASHEET.ES - Sitio de Búsqueda de Datasheet, Sitio de Búsqueda de Datasheet de Componentes Electrónicos y Semiconductores y otros semiconductores. Accedido el 3 de abril de 2024. [En línea]. Disponible: <https://www.alldatasheet.es/datasheet-pdf/pdf/82338/MICROCHIP/PIC16F877A.html>



¡GRACIAS!