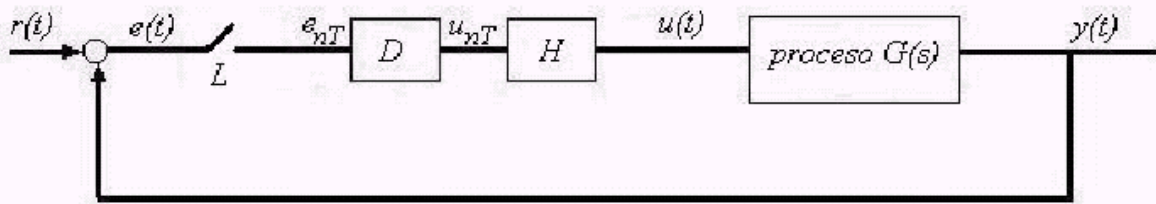


## Controladores PID digitales

### 1) Introducción

La figura



muestra un clásico esquema de control a lazo cerrado con compensación digital.  $D$  esquematiza el controlador digital, la llave  $L$  al convertidor analógico-digital (A/D) y  $H$  al convertidor digital-analógico (D/A) o reconstructor de señal.

Por simplicidad se emplea la siguiente notación:

$$e_{nT} = e(nT) \text{ y } u_{nT} = u(nT)$$

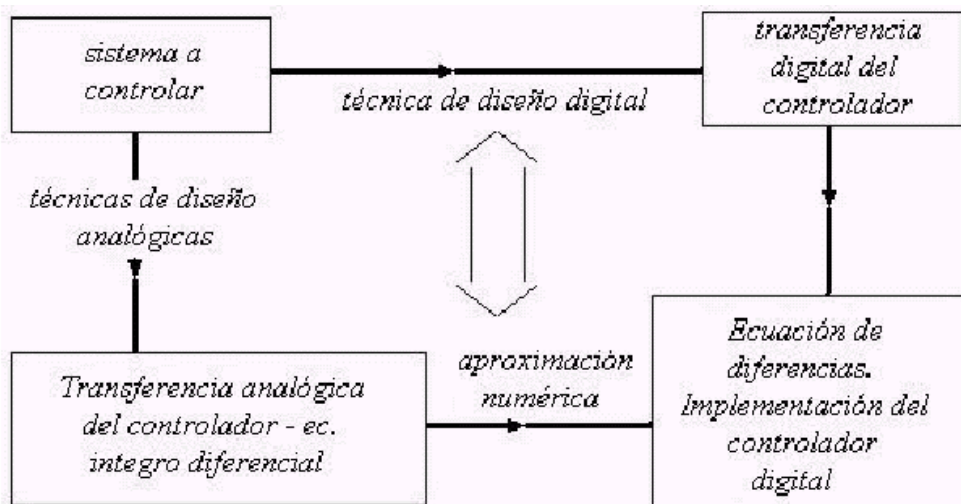
con  $n$  entero y  $T$  el período de muestreo.

El controlador digital es básicamente un procesador digital que cada  $T$  segundos resuelve un algoritmo recursivo (ecuación de diferencias). El diseño del controlador digital suele ser realizado empleando técnicas propias de los sistemas muestreados.

Estas técnicas permiten obtener controles más versátiles que los que se pueden conseguir con compensadores analógicos. Sin embargo, existen aplicaciones donde los controladores analógicos han demostrado trabajar satisfactoriamente, razón por la cual en muchas de estas aplicaciones se prefiere diseñar los controladores digitales directamente como una aproximación de los controladores analógicos. Este es, por ejemplo, el caso del controlador PID cuya implementación digital es solo una aproximación numérica de su ecuación integro-diferencial. La ecuación de un PID analógico es:

$$u(t) = K_p \cdot e(t) + K_i \cdot \int_0^t e(t) dt + K_d \cdot \frac{de(t)}{dt} = K_p \left[ e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right]$$

El siguiente esquema indica los dos caminos posibles para diseñar un controlador digital a partir de la información del sistema a controlar.



## 2) Aproximaciones numéricas

Las aproximaciones numéricas más comúnmente empleadas para discretizar la ecuación diferencial de un sistema analógico son:

- a) “Regla rectangular hacia adelante o de Euler”.
- b) “Regla rectangular hacia atrás”.
- c) “Regla trapezoidal o de Tustin”.

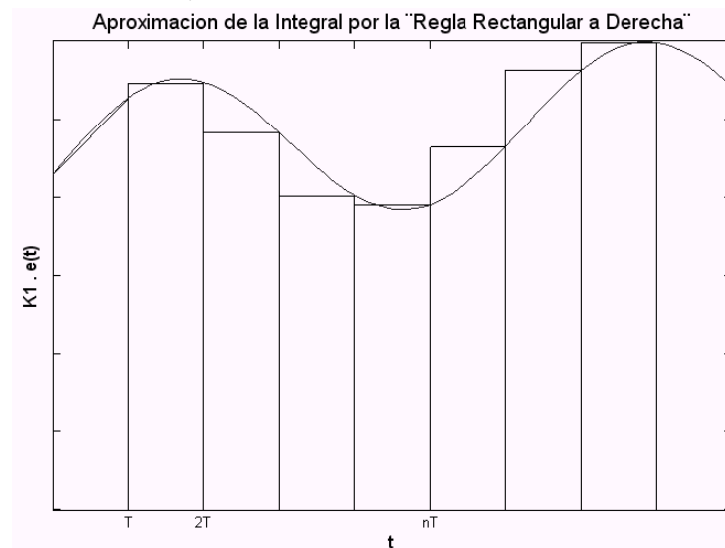
Estas reglas pueden interpretarse directamente en relación con las acciones integral y derivativa del controlador PID.

### 1. Integral

- a) “**Regla rectangular hacia atrás**”: la integral

$$u(t) = K_i \int_0^t e(t) dt$$

es decir el área bajo la curva  $K_i e(t)$ ,



es aproximada por la ecuación

$$u_{nT} \cong K_i T \sum_{j=0}^n e_{jT}$$

luego:

$$u_{nT} \cong K_i T \sum_{j=0}^{n-1} e_{jT} + K_i T e_{nT}$$

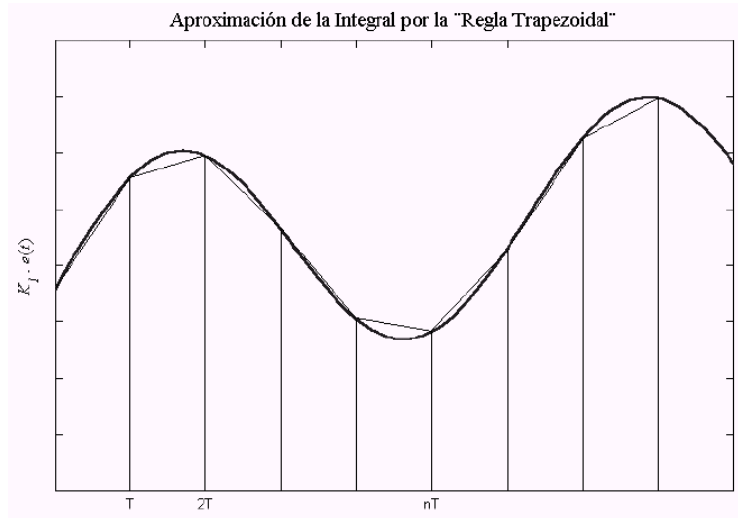
$$u_{nT} \cong u_{(n-1)T} + K_i T e_{nT}$$

La diferencia con el caso anterior radica en que se utiliza la muestra actual de la señal  $e_{nT}$  en lugar de la muestra anterior  $e_{(n-1)T}$  y el resultado de la integral en el paso previo  $u_{(n-1)T}$ .

b) **“Regla trapezoidal”**: la integral

$$u(t) = K_1 \int_0^t e(t) dt$$

es decir el área bajo la curva  $K_1 e(t)$ ,



es aproximada por la ecuación

$$u_{nT} \cong \left(\frac{T}{2}\right) \cdot \sum_{j=0}^n [e_{(j-1)T} + e_{jT}]$$

luego:

$$u_{nT} \cong u_{(n-1)T} + \left(\frac{T}{2}\right) [e_{nT} + e_{(n-1)T}]$$

Esta regla utiliza para el cálculo de la integral en el instante actual, el resultado de la integral en el paso previo ( $u_{(n-1)T}$ ) y los valores actual ( $e_{nT}$ ) y previo ( $e_{(n-1)T}$ ) de la señal.

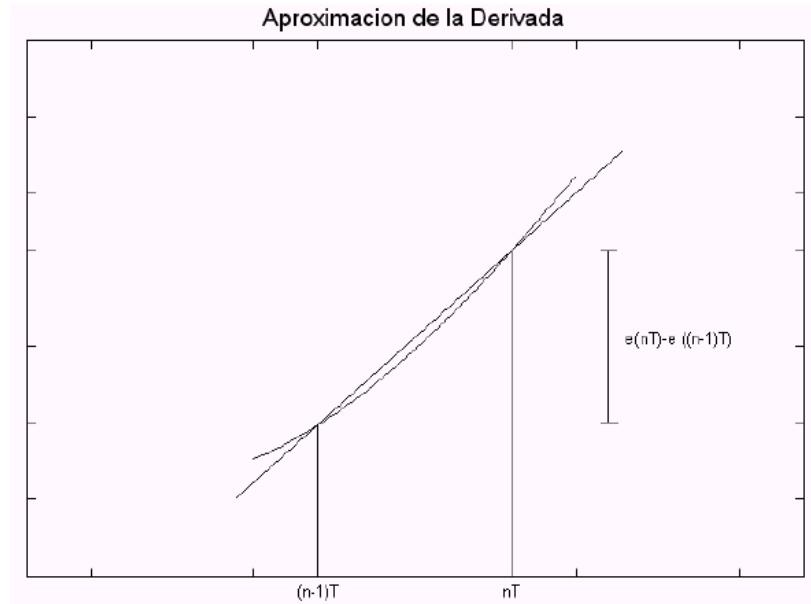
## 2. Diferenciación

a) **“Diferencia hacia atrás”**: esta es la aproximación más difundida para aproximar la derivada de una señal. La misma está dada por:

$$\frac{de(t)}{dt} \cong \frac{e(t) - e(t-T)}{T}$$

$$\frac{de(t)}{dt} \Big|_{t=nT} \cong \frac{e_{nT} - e_{(n-1)T}}{T}$$

esta última expresión solo requiere, para aproximar el valor de la derivada de una señal en un dado instante de tiempo ( $nT$ ), el conocimiento del valor de la señal en dicho instante ( $e_{nT}$ ) y en el instante de muestreo previo ( $e_{(n-1)T}$ ). Ver figura



Las aproximaciones citadas son las más empleadas para discretizar la ecuación integro-diferencial del controlador PID.

La discretización de un sistema analógico mediante alguna de las reglas previamente discutidas equivale a realizar alguna de las siguientes aproximaciones en el dominio de la frecuencia compleja s:

Regla de Euler:  $s \cong e^{sT} - 1$

Backward difference:  $s \cong \frac{e^{sT} - 1}{Te^{sT}}$

Regla de Tustin:  $s \cong \frac{2}{T} \cdot \frac{2e^{sT} - 1}{e^{sT} + 1}$

A partir de las expresiones anteriores puede obtenerse la función de transferencia aproximada del sistema a discretizar y a partir de ésta la ecuación de diferencias del sistema digital.

Las aproximaciones anteriores son válidas y presentan un comportamiento similar en la medida que el período de muestreo es chico frente a la dinámica del proceso a controlar; condición que suele verificarse en procesos industriales.

La aproximación de Euler presenta como inconveniente, que puede dar origen a un sistema digital inestable a partir de la discretización de un sistema analógico estable.

Al emplear las aproximaciones previamente discutidas surge un inconveniente, el de distorsión de la escala de frecuencia. Por ejemplo, si se desea discretizar un filtro analógico pasa banda, el filtro digital que se obtiene a partir de alguna de las aproximaciones previas no tendrá los valores correctos de frecuencias características.

Considérese, por ejemplo, la aproximación de Tustin para discretizar un sistema de función de transferencia  $G(s)$ , la función de transferencia del sistema aproximado se obtiene reemplazando la variable  $s$  de  $G(s)$  de acuerdo a la expresión

$$s \cong \frac{2}{T} \cdot \frac{2e^{sT} - 1}{e^{sT} + 1}$$

La respuesta en frecuencia del sistema aproximado está dada por:

$$G\left(\frac{2}{T} \cdot \frac{e^{j\omega T} - 1}{e^{j\omega T} + 1}\right) = G\left(j \frac{2}{T} \tan\left(\frac{\omega T}{2}\right)\right)$$

Luego si el sistema analógico  $G(s)$  presentaba comportamiento particular a una pulsación característica  $\omega_1$  el sistema aproximado presenta un comportamiento similar a una pulsación  $\omega_1'$ , la cual verifica:

$$\omega_1' = \frac{2}{T} \tan\left(\frac{\omega_1 T}{2}\right)$$

Es importante notar que si el rango de frecuencias de interés verifica que  $\omega T \ll 1$ , prácticamente no se presenta este problema. Por otra parte, si se desea eliminar este efecto de distorsión de escala a una determinada frecuencia  $\omega_1$ , se puede emplear la siguiente transformación:

$$s' \cong \frac{\omega_1}{\tan(\omega_1 T / 2)} \cdot \frac{e^{j\omega T} - 1}{e^{j\omega T} + 1}$$

### 3) Algoritmo PID digital

La ecuación integro-diferencial del controlador PID dijimos que es:

$$u(t) = K_p \cdot e(t) + K_i \cdot \int_0^t e(t) dt + K_d \cdot \frac{de(t)}{dt} = K_p \left[ e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right]$$

en su discretización se utiliza la Regla Trapezoidal y la diferencia hacia atrás

$$u_{nT} = K_p e_{nT} + \frac{T}{2} \frac{K_p}{T_i} \sum_{j=0}^n (e_{jT} + e_{(j-1)T}) + \frac{K_p T_d}{T} (e_{nT} - e_{(n-1)T})$$

$$u_{nT} = u_{(n-1)T} + K_p (e_{nT} - e_{(n-1)T}) + \frac{T}{2} \frac{K_p}{T_i} (e_{nT} + e_{(n-1)T}) + \frac{K_p T_d}{T} (e_{nT} - 2e_{(n-1)T} + e_{(n-2)T})$$

$$u_{nT} = u_{(n-1)T} + K_p (e_{nT} - e_{(n-1)T}) + \frac{K_i}{2} (e_{nT} + e_{(n-1)T}) + K_d (e_{nT} - 2e_{(n-1)T} + e_{(n-2)T})$$

asignamos:

$$A = K_p + \frac{K_i}{2} + K_d$$

$$B = -K_p - \frac{K_i}{2} - 2K_d$$

y

$$C = K_d$$

nos queda:

$$u_{nT} = u_{(n-1)T} + A e_{nT} + B e_{(n-1)T} + C e_{(n-2)T}$$

La aproximación anterior es válida si el período de muestreo es chico frente a la dinámica del proceso a controlar.

$$un = un\_1 + kp * (en - en\_1) + ki * (en + en\_1) / 2 + kd * (en - 2 * en\_1 + en\_2);$$

$$en\_2 = en\_1;$$

$$en\_1 = en;$$

$$un\_1 = un;$$

### **Efectos a tener en cuenta para seleccionar T**

Como se ha mencionado, es conveniente que  $T$  sea lo suficientemente pequeño para:

- que la aproximación empleada en la discretización sea correcta
- que no se afecte la estabilidad del lazo.
- que no se pierda información de la señal muestreada.

Efectivamente, por el teorema de muestreo de Shanon la frecuencia de muestreo debe ser por lo menos, el doble que la máxima frecuencia de la señal muestreada.

Sin embargo, el período de muestreo no puede ser reducido todo lo que se quiera" ya que existen, entre otras, limitaciones debidas a la velocidad de cálculo del procesador digital y dificultades debidas al 'redondeo' en las operaciones.

Otros factores a tener en cuenta en la selección del periodo de muestreo  $T$  son:

- espectro de las perturbaciones
- tipo de actuador empleado
- equipo de medida
- costo computacional

En muchos casos prácticos el período de muestreo se selecciona en base al tiempo de crecimiento de la respuesta del proceso a un escalón, eligiéndose  $T$  entre la sexta y vigésima parte del tiempo de crecimiento.

### **Algoritmos reset - anti windup**

Este aparece al arrancar el sistema, o en cualquier otra situación en la que aparece un error grande durante un tiempo prolongado. Sucede entonces que el integrador aumenta su salida para reducir el error aun cuando la señal de control este saturada. Esto hace que cuando el sistema se acerca al punto de referencia el término integral haya crecido demasiado y se sobrepase el valor deseado (overflow), ya que la única forma de reducir el valor del término integral es mediante un cambio de signo a su entrada (señal de error).

Hay diferentes tácticas para evitar este fenómeno denominadas “algoritmos anti reset windup (ARW)”, todas ellas de fácil implementación digital:

- limitar el término integral en un valor determinado (saturación).
- cortar la acción integral durante la saturación.

```
.
en_int = ent + ent_int

if(en_int >= 254)
    en_int = 254;

if(en_int <= 0)
    en_int = 0;
.....
```

### **1) Control proporcional integrativo**

Incrementando la ganancia del controlador, se reduce el error en régimen permanente, pero aumentará el sobrepico y se reducirá el amortiguamiento. Es necesario un controlador más complicado para reducir el error en régimen permanente con aceptable sobrepico y amortiguamiento, para ello introducimos el controlador proporcional integrativo PI.

La forma del controlador PI que usaremos es:

$$m(k) = K_p \left[ \Delta(k) + K_i T_s \sum_{\ell=0}^k \Delta(\ell) \right]$$

donde  $\Delta(k)$  es la entrada del controlador y  $m(k)$  es la salida, y  $\ell$  es una variable de la sumatoria que aproxima a la integral. La transformada  $z$  de la integral que entonces está aproximada por la sumatoria es:

$$Z\left\{\sum_{\ell=0}^k \Delta(\ell)\right\} \approx \left(\frac{z}{z-1}\right)\Delta(z)$$

la función de transferencia del controlador PI es:

$$G_{cPI}(z) = K_p \left[ 1 + K_i T_s \left( \frac{z}{z-1} \right) \right]$$

para que queden evidentes los polos y ceros del controlador PI, describimos la función de transferencia como una función racional:

$$G_{cPI}(z) = K_p \left[ \frac{(1 + K_i T_s)z - 1}{z - 1} \right]$$

de esta manera vemos que  $G_{cPI}(z)$  tiene un polo en  $z = 1$  y un cero en  $z = \frac{1}{1 + K_i T_s}$  que se encuentra a la izquierda del polo. Usando este controlador, nos aseguramos que la respuesta al escalón tenga error en régimen permanente nulo.

## 2) Control proporcional integrativo derivativo

Con el controlador PI, esperamos que nuestro sistema de control satisfaga las especificaciones de sobrepico, tiempo de establecimiento y error en régimen permanente. Pero si se desea un menor tiempo de crecimiento sin incrementar el sobrepico, es necesario incluir un control derivativo. La aproximación en tiempo discreto del término derivativo agregado al controlador PI, el nuevo controlador PID está dado por:

$$m(k) = K_p \left[ \Delta(k) + K_i T_s \sum_{\ell=0}^k \Delta(\ell) + K_d \left( \frac{\Delta(k) - \Delta(k-1)}{T_s} \right) \right]$$

donde  $\Delta(k)$  es la entrada del controlador y  $m(k)$  es la salida, la transferencia del PID es  $M(z)/\Delta(z)$  que puede escribirse como:

$$G_{cPID}(z) = K_p \left[ 1 + K_i \left( \frac{T_s z}{z-1} \right) + K_d \left( \frac{z-1}{T_s z} \right) \right]$$

para hacer visibles los polos y ceros del controlador PID, reescribimos la transferencia como una función racional de  $z$ :



$$G_{cPID}(z) = K_p \left[ \frac{(1 + K_i T_s + K_d / T_s) z^2 - (1 + 2K_d / T_s) z + K_d / T_s}{z(z-1)} \right]$$

esta función tiene un polo en  $z=1$  como tiene la función de transferencia del PI, pero además tiene un polo adicional en  $z=0$ . También tiene dos ceros que pueden ser reales o complejos conjugados, dependiendo de los valores que el diseñador seleccione para  $K_d$  y  $K_i$ .

Una vez que quedó establecida la ganancia  $K_i$  del controlador PI, se requerirán más iteraciones para la determinación de las ganancias  $K_p$  y  $K_d$  dado que se deben cumplir dos especificaciones sobrepico y tiempo de crecimiento. Sin embargo puede haber varios valores de esas ganancias que cumplan todas las especificaciones.

### **Ejemplo implementación control de posición de un motor de CC realimentado con potenciómetro**

Es un programa base y deberá ajustarse a la planta y controlador a utilizar

```
main()
{
    Zona definición de variables y constantes.
    Kp, Kd , Ki.
    Inicialización periféricos del microcontrolador. Los
    comentarios en rojo depende del microcontrolador que se use.
    while(1)
    {
        Inicializo timer para asegurarme que todo lo que se realiza
        abajo dure un tiempo constante, ese tiempo define el
        muestreo.
        Adquiero los datos de setpoint (Setp) sea éste desde
        teclado o potenciómetro y Realimentación de posición (Pos)
        desde potenciómetro. Lo adquiero con ADC seteado en 10
        bits. Tener en cuenta que los ADC de los micros son
        UNIPOLARES, admiten solo tensiones positivas.

        // Calculo error
        Error = (Kp * (Setp - Pos)) + (Kd * (Setp - Pos - Der_pos))
        + Ki * Int_pos);

        // Muestra vieja para la derivada
        Der_pos = Setp - Pos;

        // Integral del error
        Int_pos = Int_pos + Setp - Pos;
```

```

// Anti wind-up
if(Int_pos >= 1200)
    Int_pos = 1200;
if(Int_pos <= -1200)
    Int_pos = -1200;

// Determino sentido de giro
if(Error >= 0)
    // avanza paun lao
    sent = 1;
else
{
    // avanza pal otro
    sent = 0;
    // Valor absoluto error
    Error = -1 *Error;
}
// Ajuste saturación
// Que el error absoluto esté entre 0 y 1023
if(Error <= 0)
    Error = 0;

if(Error >= 1023)
    Error = 1023;
Sale Error hacia el Registro de PWM que normalmente son de
10 bits. Uso sent para manejar un bit del puente H así fijo
la dirección para donde debe girar el motor.
}
}

```