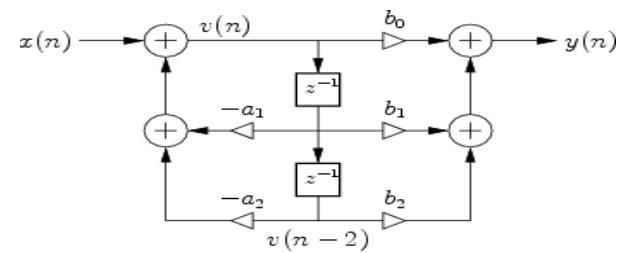


Control Automático

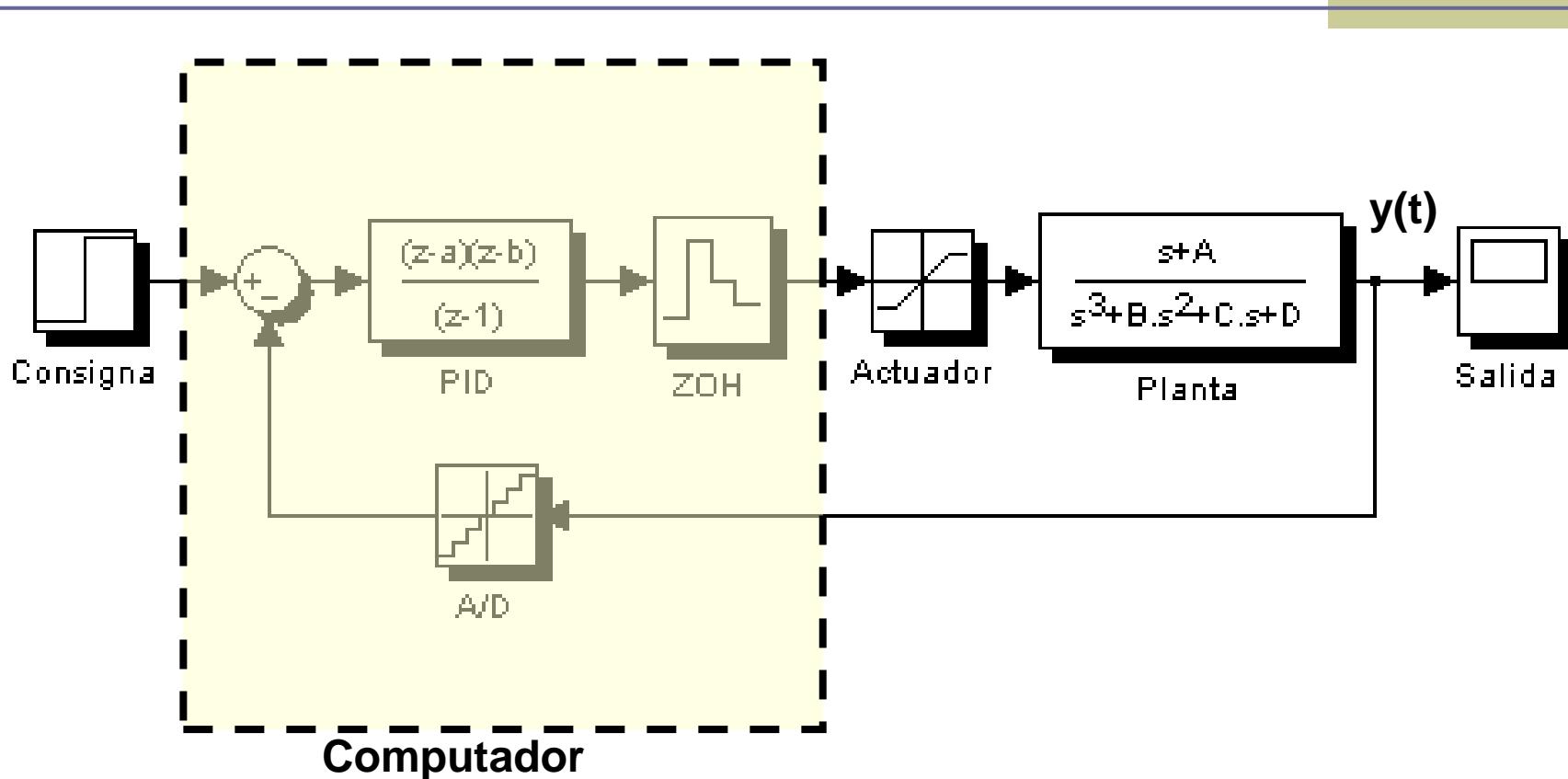
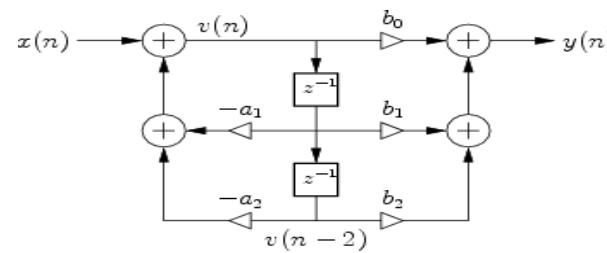
Implementación de reguladores digitales

Contenido

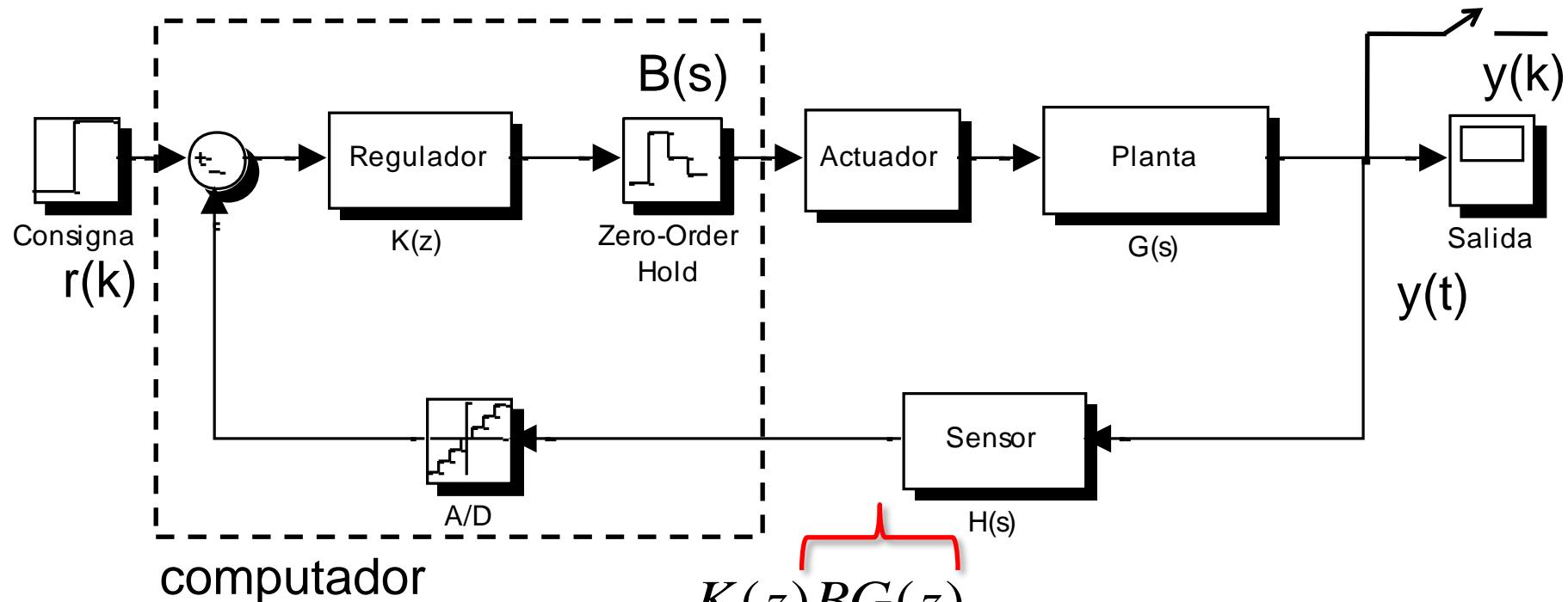
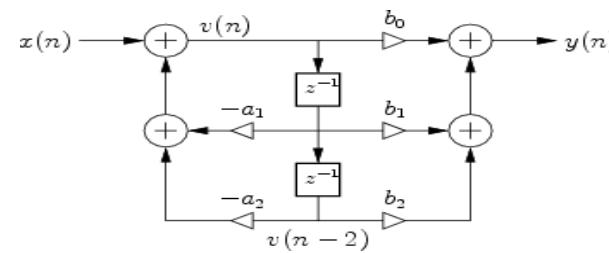


- Formas de implementación de reguladores digitales (filtros digitales). Ecuación de diferencias
 - Formas directas I y II
 - Forma transpuesta
 - Forma paralelo
- Implementación de la ecuación de diferencias en computadores digitales

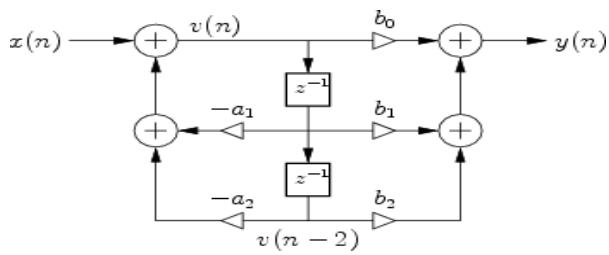
Esquema de un control digital



Sistema híbrido de control

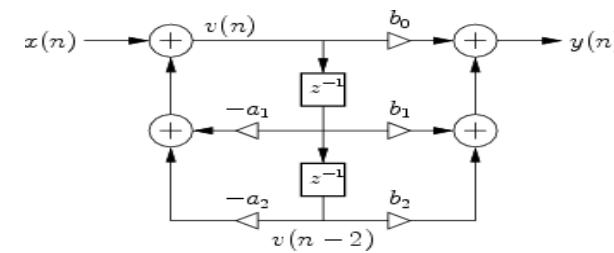


$$T(z) = \frac{K(z)BG(z)}{1 + K(z)BGH(z)}$$

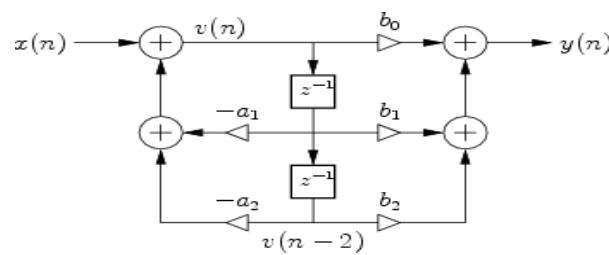


Ecuación de diferencias a
partir de la función de
transferencia

Formas de implementación



- Forma directa I (DF-I)
- Forma directa II o estándar (FCC)
- Estándar transpuesta (FCO)
- Paralelo
- Escalera o rejilla



Forma directa I, DF-I

Se parte de la función de transferencia

$$G(z) = \frac{M(z)}{E(z)} = \frac{N(z)}{D(z)} = \frac{b_q z^q + b_{q-1} z^{q-1} + \cdots + b_1 z + b_0}{a_n z^n + a_{n-1} z^{n-1} + \cdots + a_1 z + a_0}$$

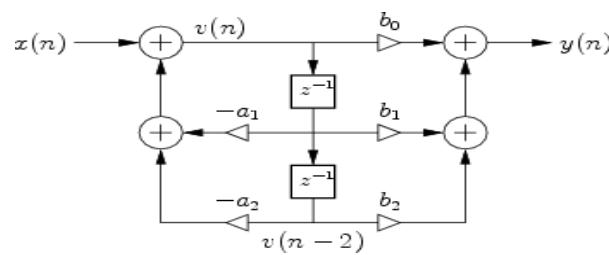
Después de reescribir y dividir entre z^n

$$\sum_{i=0}^n a_i z^{i-n} M(z) = \sum_{i=0}^q b_i z^{i-n} E(z)$$

Se obtiene la transformada z inversa

$$\sum_{i=0}^n a_i \cdot m(k+i-n) = \sum_{i=0}^q b_i \cdot e(k+i-n)$$

En algunos textos
los subíndices se
numeran al revés



Forma directa I (2)

Se despeja el término $m(k)$

$$m(k) = -\sum_{i=0}^{n-1} \left(\frac{a_i}{a_n} \cdot m(k+i-n) \right) + \sum_{i=0}^q \left(\frac{b_i}{a_n} \cdot e(k+i-n) \right)$$

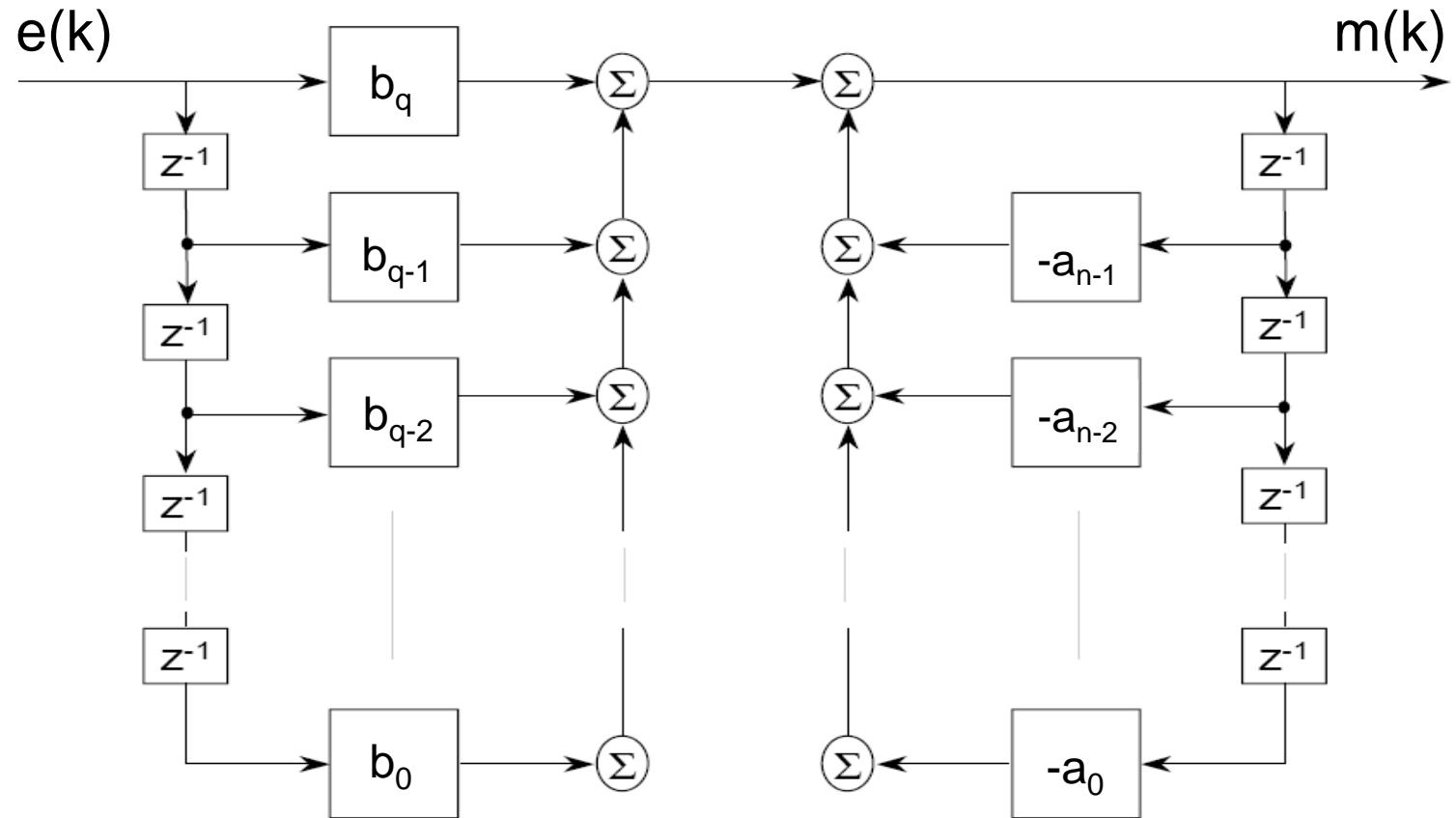
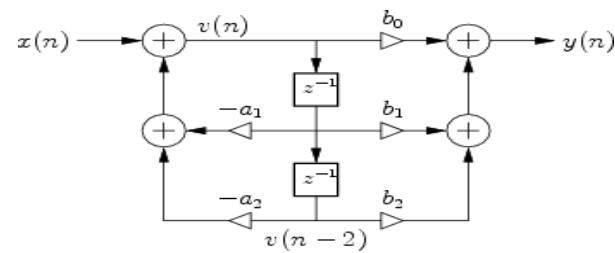
Escribiendo en forma desarrollada

$$m(k) = -\frac{a_0}{a_n} m(k-n) - \frac{a_1}{a_n} m(k+1-n) - \cdots - \frac{a_{n-1}}{a_n} m(k-1) +$$

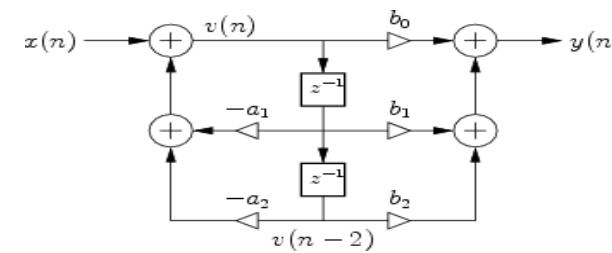
Usualmente $a_n = 1$

$$+\frac{b_0}{a_n} e(k-n) + \frac{b_1}{a_n} e(k+1-n) + \cdots + \frac{b_q}{a_n} e(k+q-n)$$

Forma directa I (3)

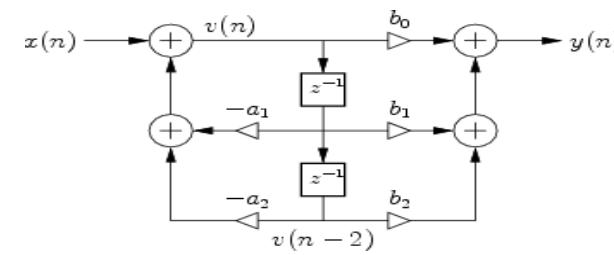


Forma directa I (4)



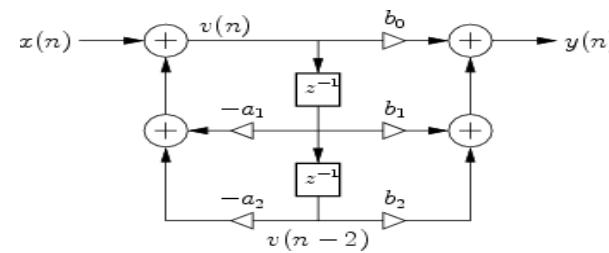
- Puede ser considerada como una sección de filtro de ceros seguida de una sección de filtro de polos.
- Una propiedad importante y valiosa de la estructura DF-I, debida a que únicamente existe un punto de suma en el filtro y a que la aritmética de punto fijo, p.e. complemento a dos, simplemente se desborda del número más positivo al más negativo y viceversa; **siempre que el resultado final se encuentre dentro del rango, no existe desbordamiento interno del filtro**, aun si existe desbordamiento de los resultados intermedios de la suma.
- Se requiere el doble de los retardos necesarios (memoria). La estructura DF-I no es canónica con respecto al retardo. En general es posible implementar un filtro de orden n con solamente n retardos (DF-II)

Forma directa I (5)



- En todas las formas directas, aquellas en las cuales sus coeficientes están dados por los coeficientes de la función de transferencia, los polos y ceros del filtro son muy **sensibles a los errores de redondeo de los coeficientes**. Esto es especialmente problemático para filtros de órdenes altos.
- Para minimizar la sensibilidad, es mejor utilizar implementaciones serie o paralelo de orden dos, o la DF-II transpuesta.

Ejemplo 1: Forma directa I



Compensador

$$G(z) = \frac{M(z)}{E(z)} = k_c \frac{(z + c)}{(z + a)(z + b)} = \frac{k_c z + k_c c}{z^2 + (a + b)z + ab}$$

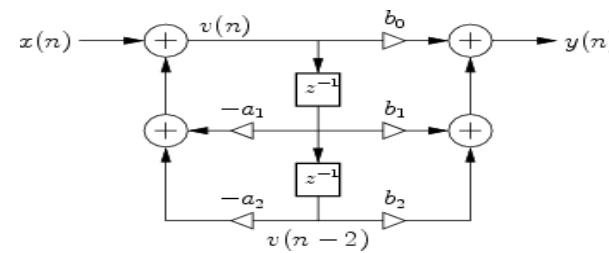
Se divide entre z^2 (z^n) y se separa

$$M(z) + (a + b)z^{-1}M(z) + ab \cdot z^{-2}M(z) = k_c \cdot z^{-1}E(z) + k_c \cdot c \cdot z^{-2}E(z)$$

Transformando al dominio del tiempo discreto y despejando $m(k)$

$$m(k) = -(a + b)m(k - 1) - ab \cdot m(k - 2) + k_c \cdot e(k - 1) + k_c \cdot c \cdot e(k - 2)$$

Ejemplo 2: forma directa I



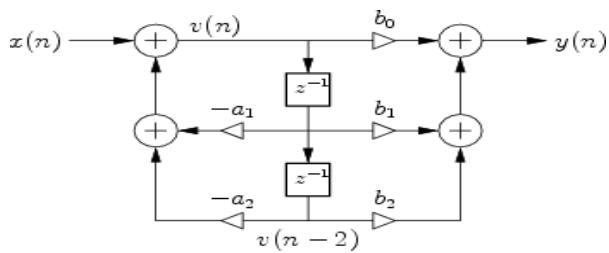
Encuentre la implementación DF-I, para el controlador $K(z)$

$$K(z) = \frac{z(z - 0.97)}{z^2 - 0.099z + 0.058}; T = 0.05s$$

Transformando al dominio del tiempo discreto y despejando $m(k)$

$$M(z) - 0.099z^{-1}M(z) + 0.058z^{-2}M(z) = E(z) - 0.97z^{-1}E(z)$$

$$m(k) = 0.099m(k-1) - 0.058m(k-2) + e(k) - 0.97e(k-1)$$



Forma directa II o estándar

Se multiplica la función de transferencia por $H(z)/H(z)$

$$\frac{Y(z)}{U(z)} = \frac{[b_q z^{q-n} + b_{q-1} z^{q-1-n} + \dots + b_1 z^{1-n} + b_0 z^{-n}]}{[1 + a_{n-1} z^{-1} + a_{n-2} z^{-2} + \dots + a_1 z^{1-n} + a_0 z^{-n}]} \cdot \frac{H(z)}{H(z)}$$

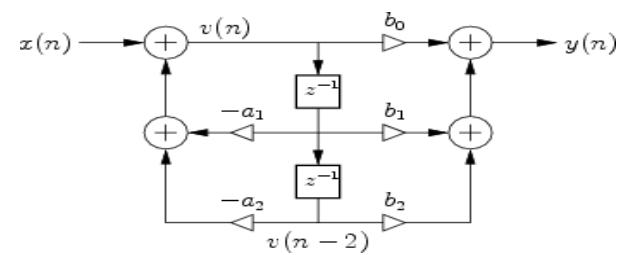
Separando el numerador y el denominador en dos ecuaciones

$$Y(z) = [b_q z^{q-n} + b_{q-1} z^{q-1-n} + \dots + b_1 z^{1-n} + b_0 z^{-n}] \cdot H(z)$$

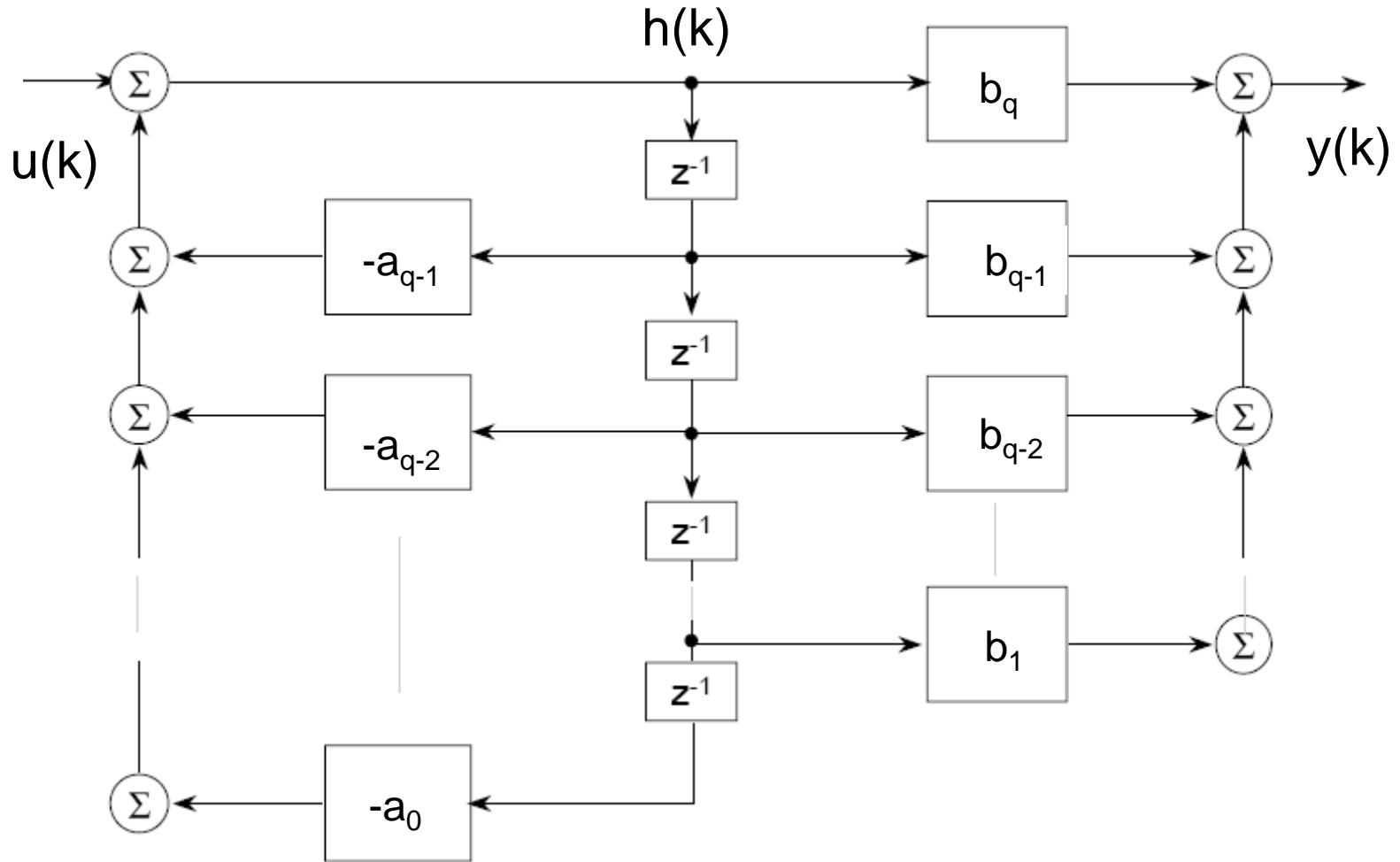
$$U(z) = [1 + a_{n-1} z^{-1} + a_{n-2} z^{-2} + \dots + a_1 z^{1-n} + a_0 z^{-n}] \cdot H(z)$$

$$h(k) = u(k) - a_{n-1} h(k-1) - a_{n-2} h(k-2) - \dots - a_1 h(k+1-n) - a_0 h(k-n)$$

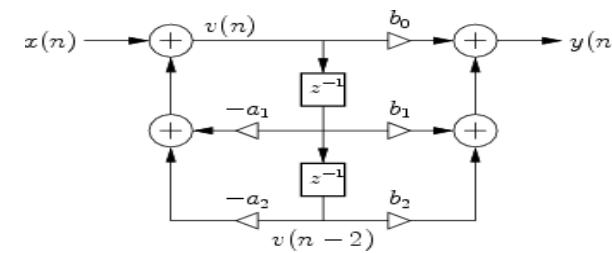
$$y(k) = b_q h(k+q-n) + b_{q-1} h(k+q-1-n) + \dots + b_1 h(k+1-n) + b_0 h(k-n)$$



Forma directa II o estándar (2)

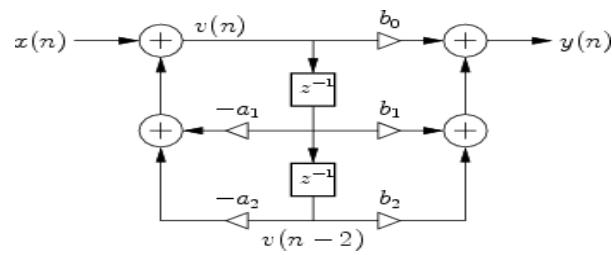


Forma directa II o estándar (3)

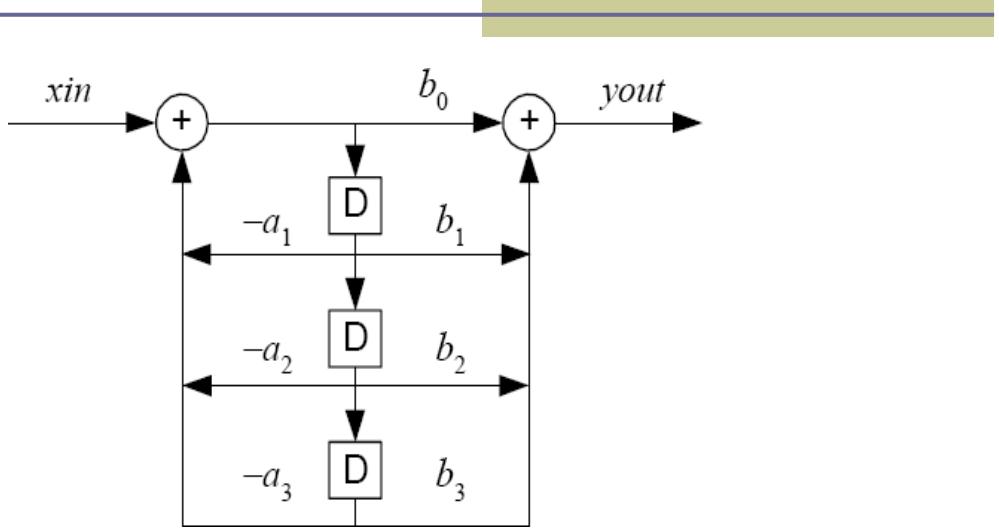


- Puede ser vista como una sección filtro de polos seguida de una sección de filtro de ceros.
- Es canónica respecto al retardo. Esto es porque los elementos de retardo asociados con las secciones de filtro de polos y ceros, son compartidos.
- Puede ocurrir un desborde interno (*overflow*) en la entrada de la línea de retardo, a diferencia de la implementación DF-I.
- Como en todas las estructuras de filtros de forma directa, los polos y ceros son sensibles a errores de redondeo en los coeficientes, especialmente en funciones de transferencia de gran orden. Sensibilidades menores son obtenidas descomponiendo en secciones de orden bajo (segundo orden) o usando estructuras en escalera (*lattice filter*)

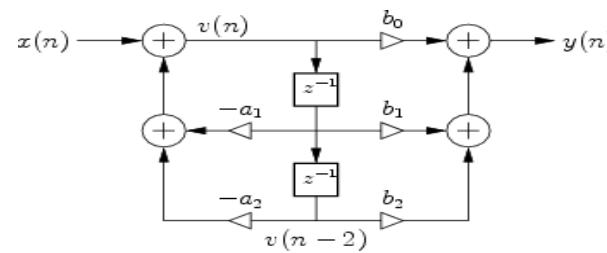
Forma directa II transpuesta



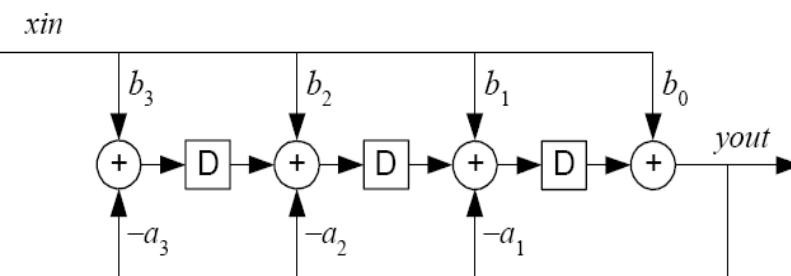
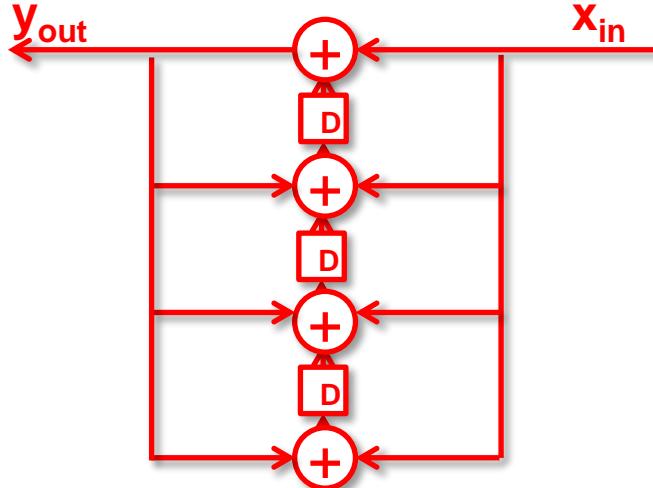
- Para un sistema SISO, realice los acomodos de cambiar los puntos de ramificación de señal a sumadores y viceversa y simplemente invierta la dirección de todos los caminos de señal.



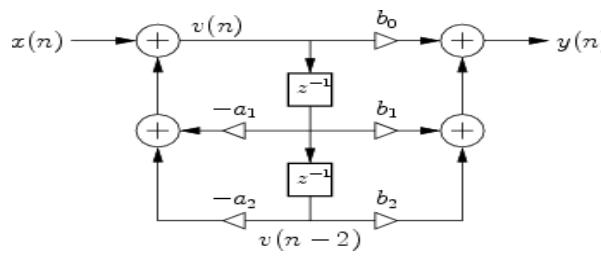
Forma directa II transpuesta



- Para un sistema SISO, realice los acomodos de cambiar los puntos de ramificación de señal a sumadores y viceversa y simplemente invierta la dirección de todos los caminos de señal.
- Después de esta operación, la entrada, usualmente a la izquierda, se encuentra a la derecha. Para renormalizar, todo el esquema se refleja horizontalmente.



Ejemplo 3: Forma directa II y D-II transpuesta

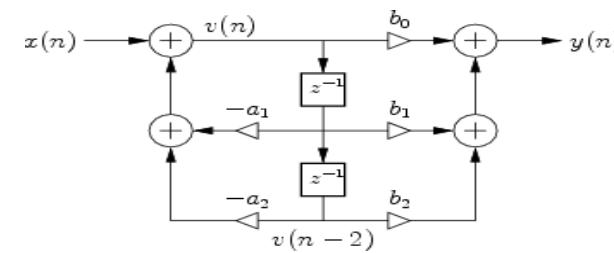


- Dibuje la estructura y coloque los valores adecuados a ésta, para implementar, en la forma DF_II transpuesta, el regulador $K(z)$ mostrado.

$$K(z) = \frac{0.019 (z + 0.9737)}{(z^2 - 1.885z + 0.9231)}, T = 10ms$$

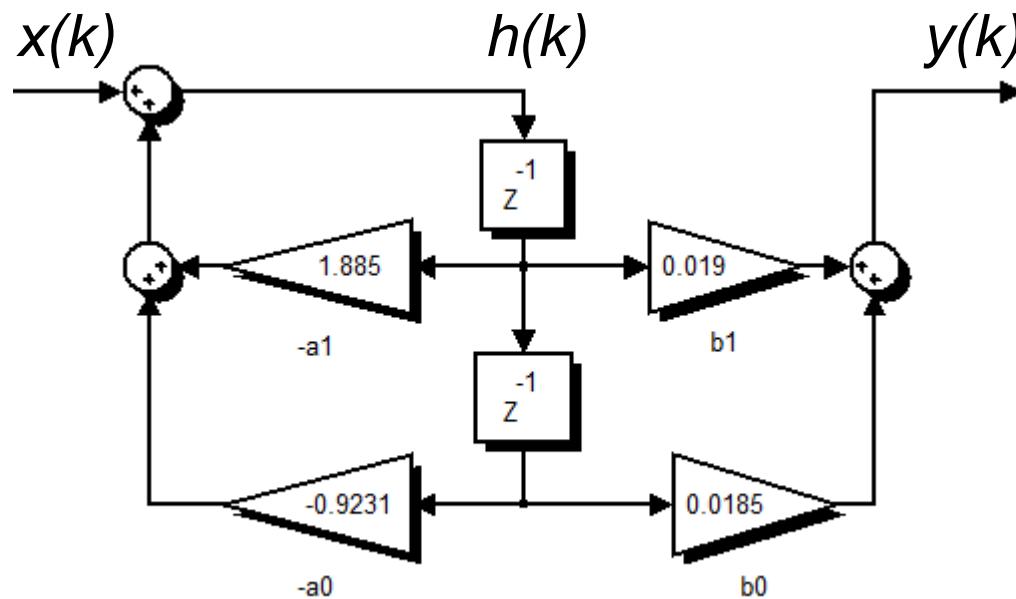
$$K(z) = \frac{Y(z)}{X(z)} = \frac{0.019z^{-1} + 0.0185z^{-2}}{1 - 1.885z^{-1} + 0.9231z^{-2}} \cdot \frac{H(z)}{H(z)}$$

Ejemplo 3: Forma directa II

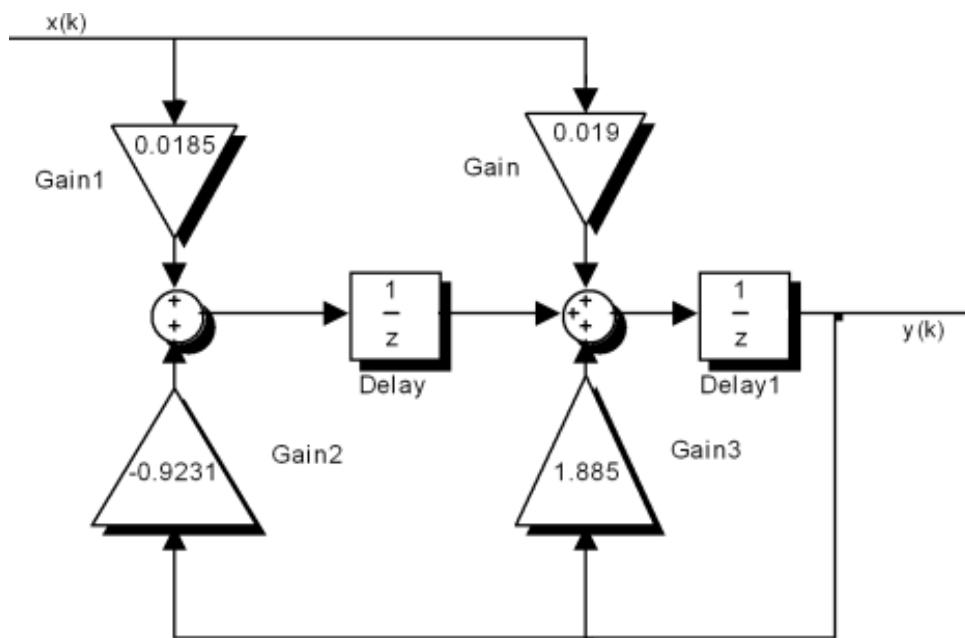
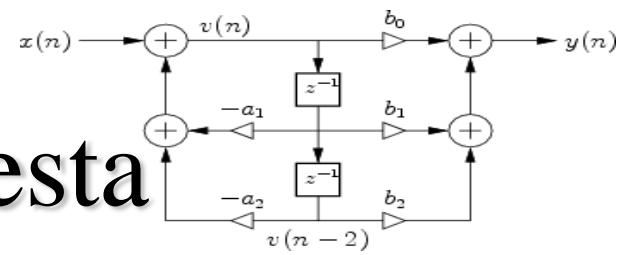


$$y(k) = 0.019h(k-1) + 0.0185h(k-2)$$

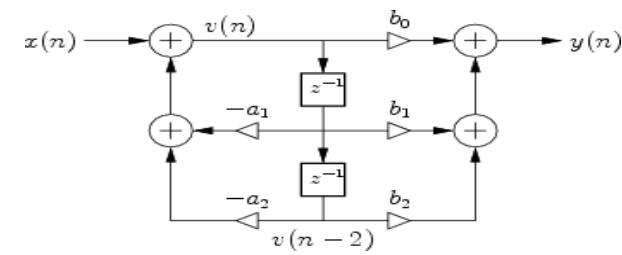
$$h(k) = x(k) + 1.885h(k-1) - 0.9231h(k-2)$$



Ejemplo 3: Forma directa II transpuesta

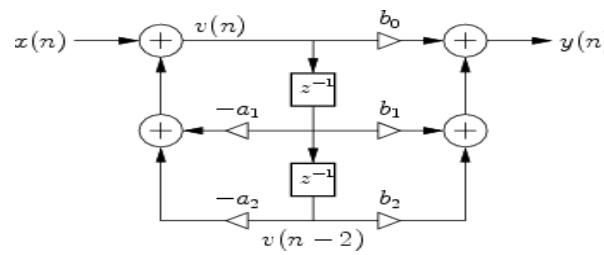


Paralelo

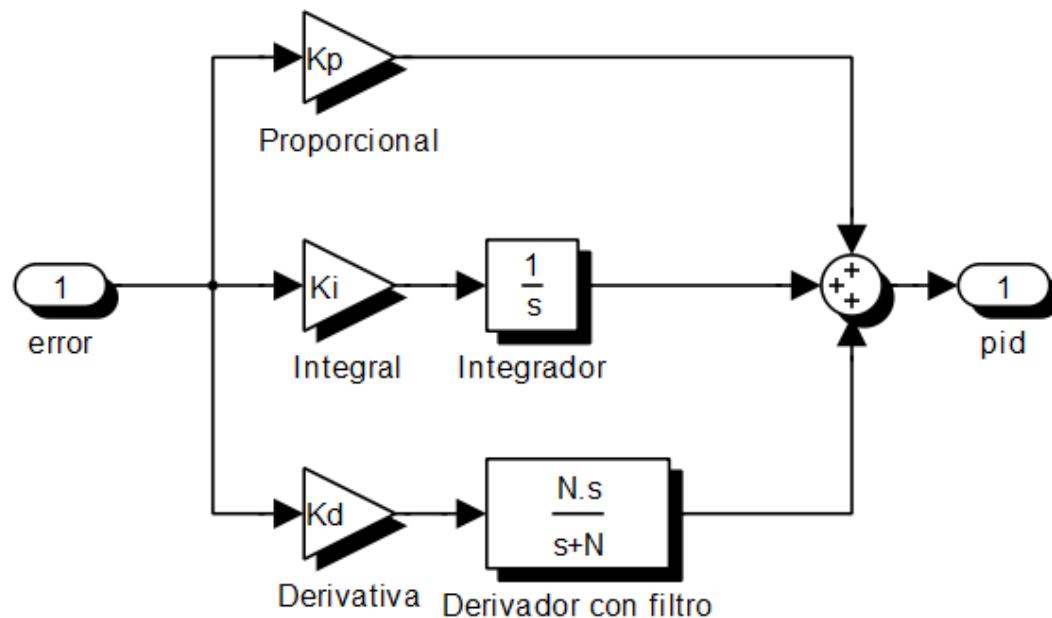


- La forma paralelo se obtiene separando la función de transferencia en fracciones parciales
- Cada fracción parcial se implementa y luego se suma con las otras fracciones
- Puede ser usada para implementar un PID *antiwindup* por saturación del término integral o por seguimiento integral

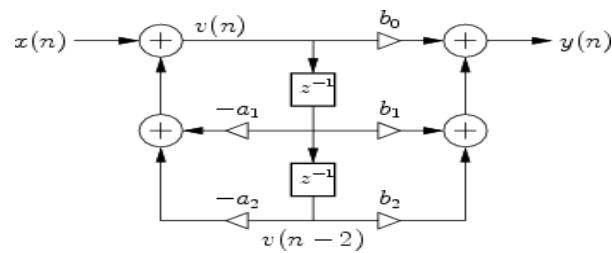
PID real en paralelo



- El PID se implementa como la suma de sus tres términos
- Usualmente se colocan límites al integrador (*antiwindup*)
- Para que el derivador sea propio y disminuir el ruido se agrega un filtro

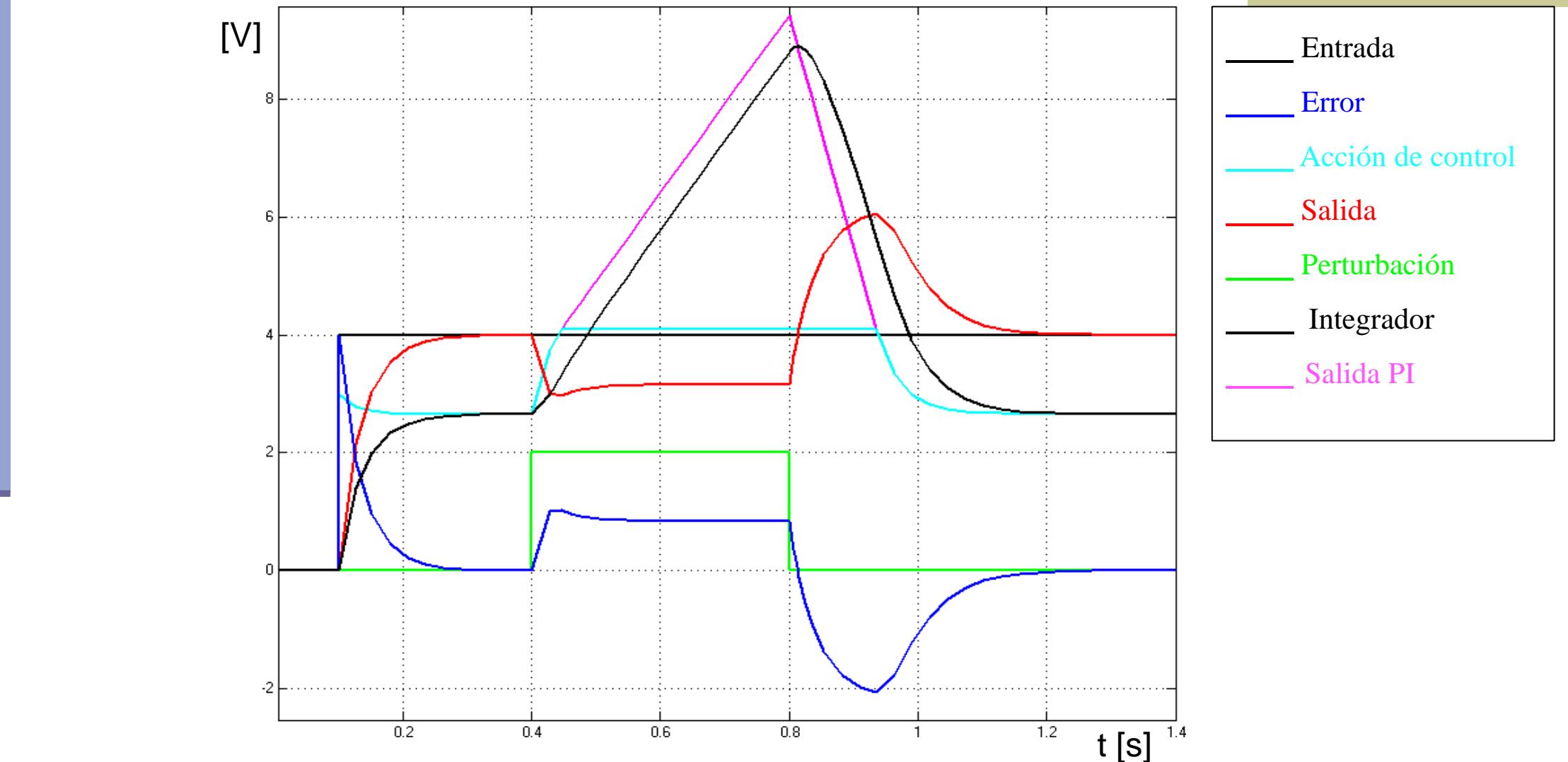
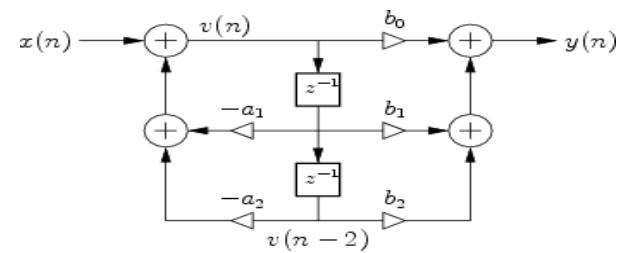


¿Qué es el *windup*?

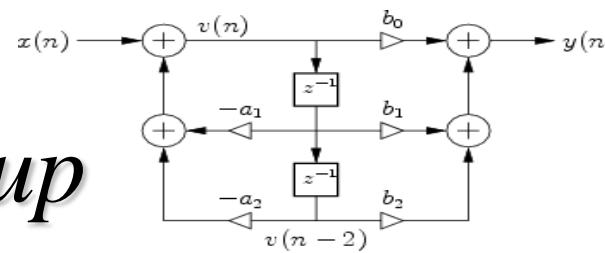


- Es la acumulación de un gran valor en la sumatoria o integral del error, debido a:
 1. Saturación en los actuadores
 2. Un error muy grande por
 - Un cambio muy grande en la consigna
 - Un error sostenido

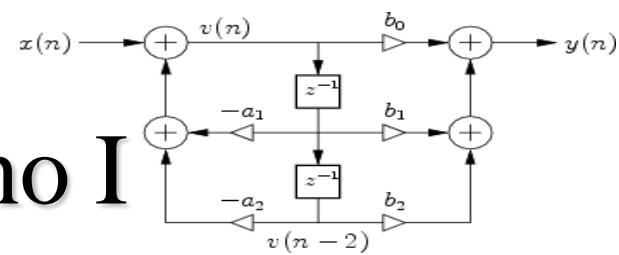
¿Qué es el *windup*?



Metodología de diseño del PID *antiwindup*

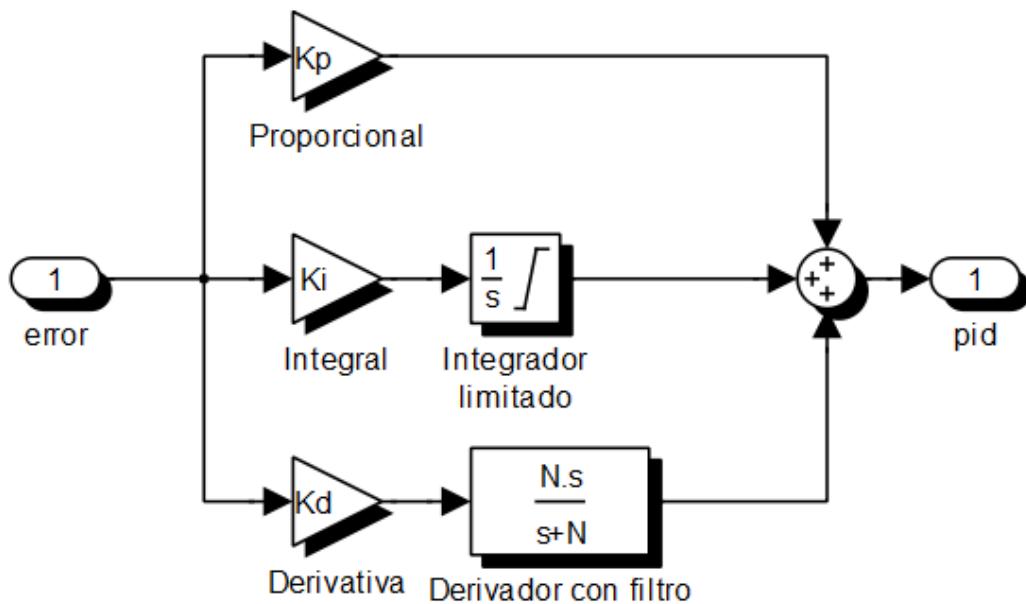


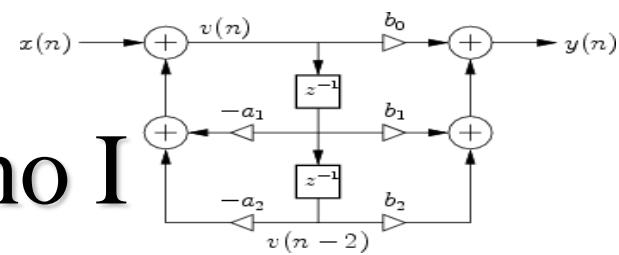
1. Diseñar el PID ideal
2. Definir los límites de los actuadores
3. Agregar al PID ideal la compensación *antiwindup* cuando se satura el actuador
 - a) Saturar el término integral
 - b) Suspender temporalmente la integral (seguimiento integral)



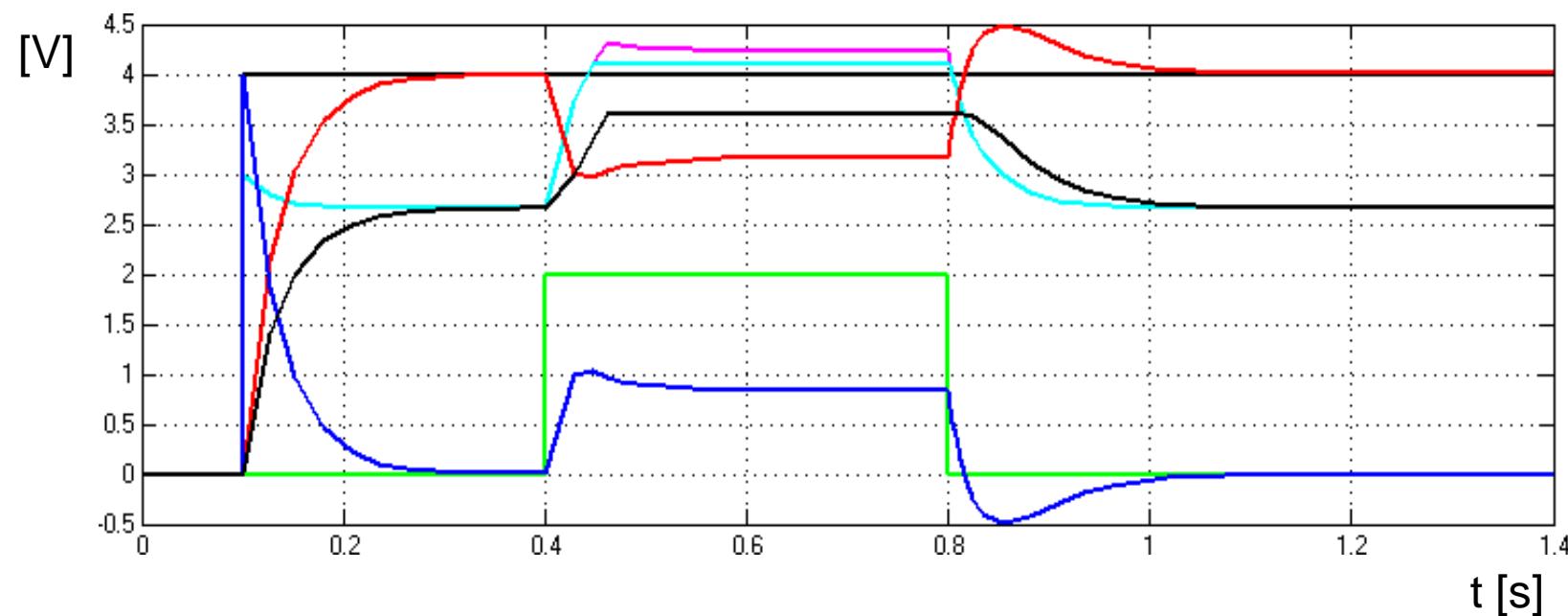
PID *antiwindup* por limitación del término I

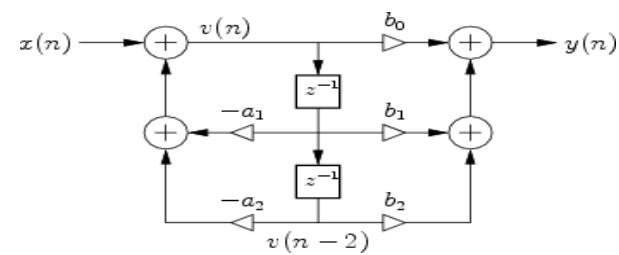
- El PID es no lineal e invariante
- El diseñador impone los límites usando su experiencia e intuición
- Los límites son fijos para un actuador
- Fuera del rango permitido se cancela la acción I





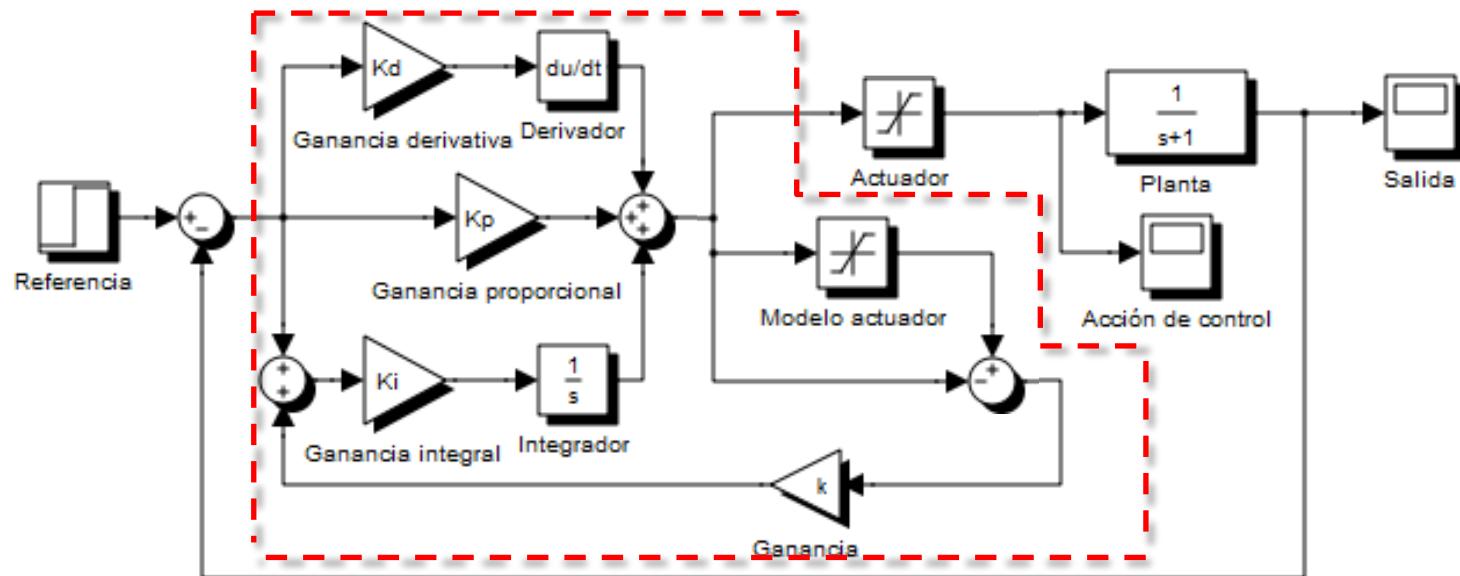
PID *antiwindup* por limitación del término I



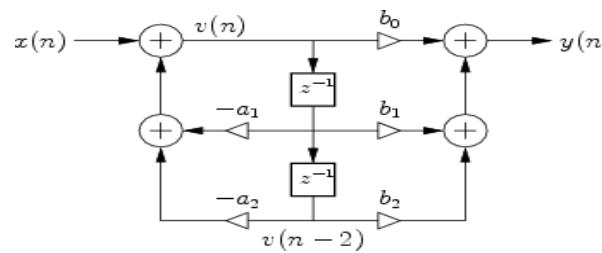


PID *antiwindup* de seguimiento integral

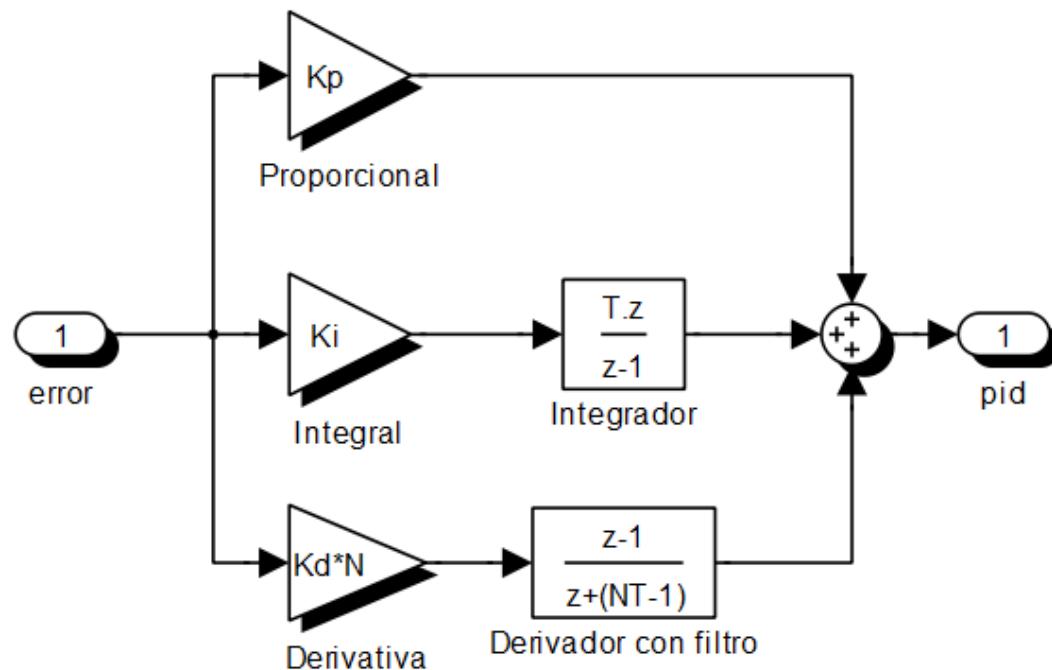
- PID lineal y variante
- Se agrega realimentación dentro del PID
- Al existir saturación se modifica la salida $v(t)$ del PID para que sea igual a $u(t)$, la acción de control sobre la planta
- La realimentación solamente actúa cuando hay saturación

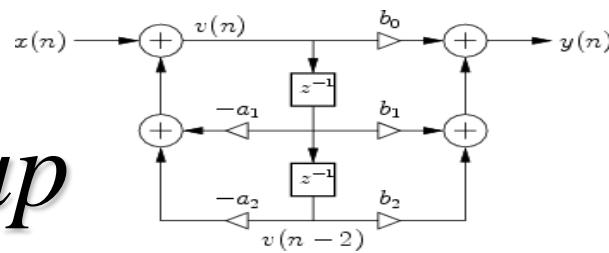


PID real en paralelo en tiempo discreto



- El PID se implementa como la suma de sus tres términos transformando cada término por separado
- El tiempo de muestreo T debe ser el adecuado para la respuesta deseada de la planta.
- $1/s$ puede ser transformado por los métodos: *forward Euler*, $(T/(z-1))$; *backward Euler*, $(T.z/(z-1))$ o *Trapezoidal / Tustin* $(T/2 \cdot (z+1)/(z-1))$.





Ejemplo: PID real paralelo *antiwindup*

- Se implementará un PID real creado como compensador de adelanto-PI en paralelo con *antiwindup*

$$PID(z) = \frac{M(z)}{E(z)} = \frac{(z-0.8)(z-0.95)}{(z-1)(z-0.4)}, T=1$$

- Por fracciones parciales mejor se procesa $\frac{PID(z)}{z}$

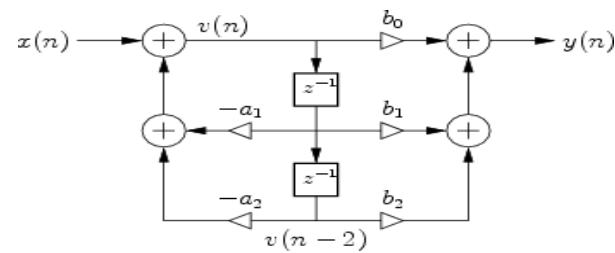
$$\frac{PID(z)}{z} = \frac{R_1}{z} + \frac{R_2}{(z-0.4)} + \frac{R_3}{(z-1)}, \quad T=1$$

$$PID(z) = 1.9 - \frac{0.9167}{(z-0.4)} z + \frac{0.016667}{(z-1)} z, T=1$$

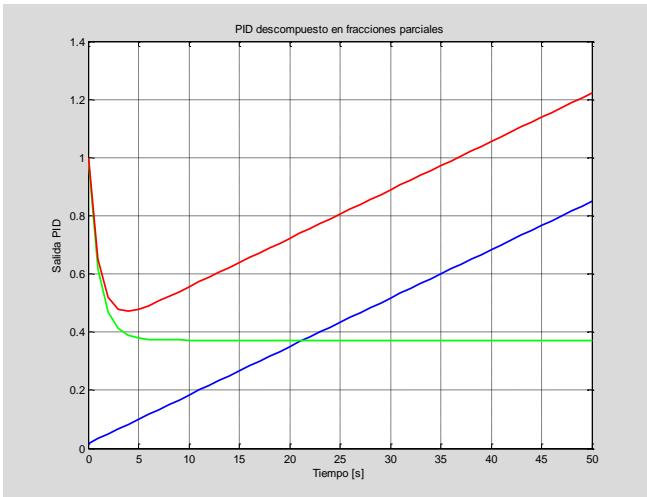
- O también expresado en forma “PD” + I

$$PID(z) = \frac{0.98333 (z-0.7729)}{(z-0.4)} + \frac{0.016667}{(z-1)} z, T=1$$

Ejemplo: PID *antiwindup*

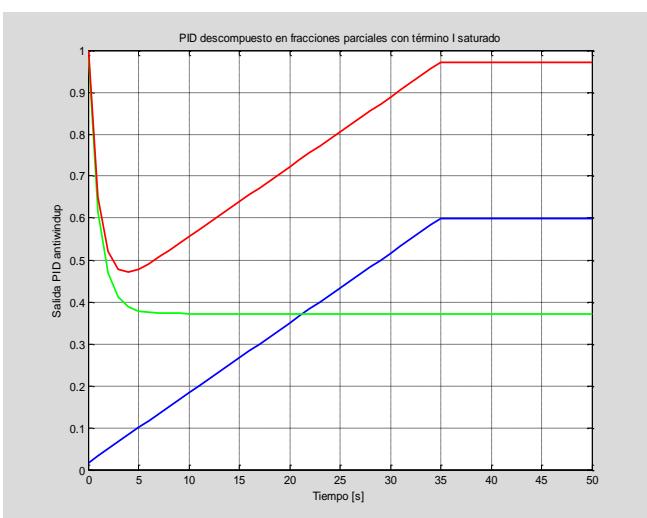


■ PID original

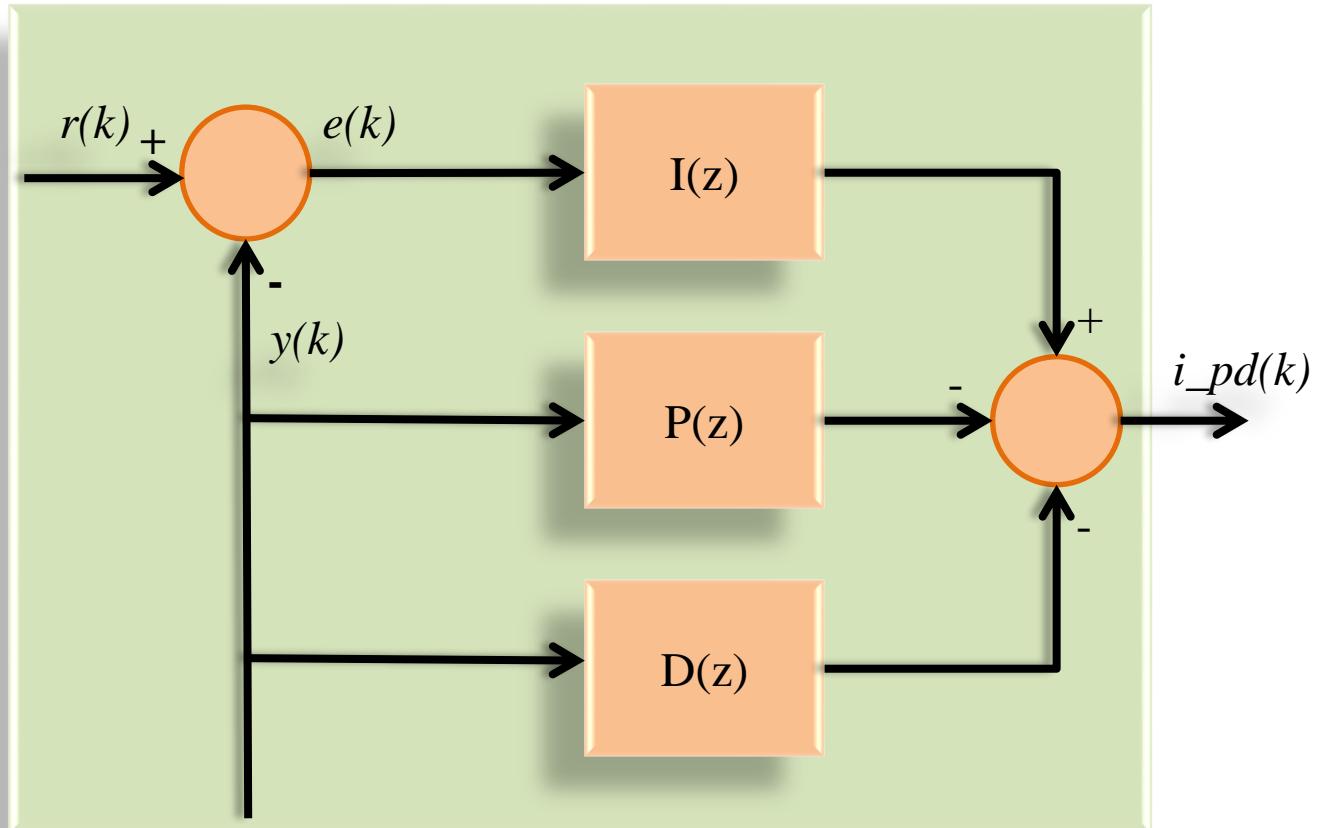
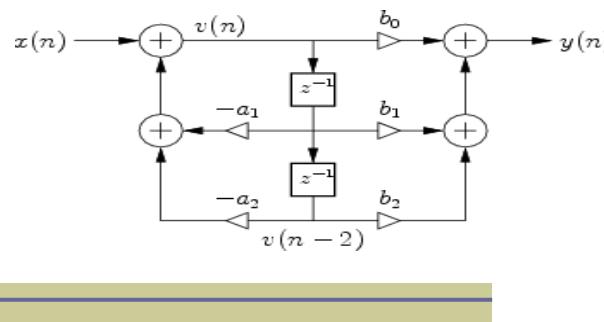


■ PID con el término integral saturado en el valor arbitrario 0.6.

$$PID(z) = 1.9 - \frac{0.9167 z}{(z-0.4)} + \frac{0.016667 z}{(z-1)}, T = 1$$



Ejemplo de implementación I_PD paralelo



PID paralelo I_PD

Ejemplo de implementación I_PD paralelo

```
CY_ISR(Timer5ms_Interrupt)
{
    /* Inicia la interrupción cada 5ms */

    /* Lee la posición actual en cuentas con 8 bits, 128 representa el cero */
    POT = ADC_GetResult8();

    /* Ajusta la variable al rango de -2V a +2V */
    yk = POT-128;

    /* Determina el error e inicia el cálculo del regulador I_PD */
    ek = REFERENCIA - yk;

    /* Calcula la porción integral */
    ik += KI*ek;

    /* Porción derivada actuando sobre realimentación de la posición y el valor anterior de dk, d(k-1) */
    dk = KD*(yk - yk1) + (KD2*dk)/256;

    /* Acción total de control I_PD */
    acc = ik -KP*yk -dk;

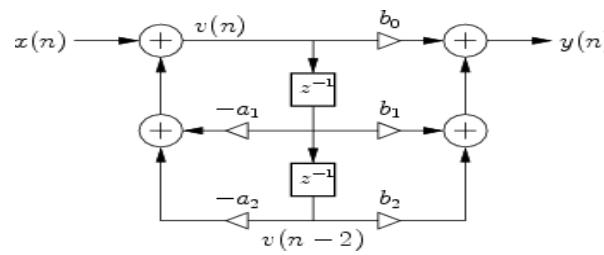
    /* Escala, limita el dac */
    if(acc < 0) vdac = 0;
    else if (acc > 32000) vdac = 125; /* 2V */
        else vdac = (uint8) (acc >> 8);

    /* Actualiza el VDAC */
    VDAC_SetValue(vdac);

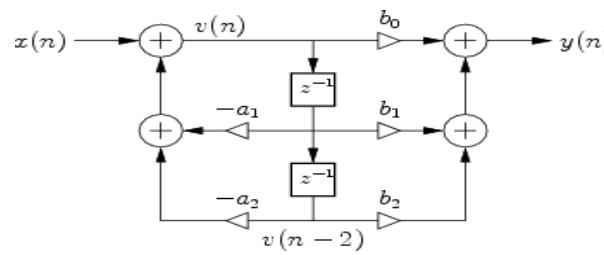
    /* Guarda la posición para la siguiente vez */
    yk1 = yk;

    /* Satura el término integral */
    if (ik > MAXINTEGRAL)
        ik = MAXINTEGRAL;
    else
        if (ik < -MAXINTEGRAL) ik = -MAXINTEGRAL;

    /* '#END' */
}
```



Ejemplo de implementación I_PD paralelo



```
CY_ISR(Timer5ms_Interrupt)
{
    /* Inicia la interrupción cada 5ms */

    /* Lee la posición actual en cuentas con 8 bits, 128 representa el
    POT = ADC_GetResult8();

    /* Ajusta la variable al rango de -2V a +2V */
    yk = POT-128;

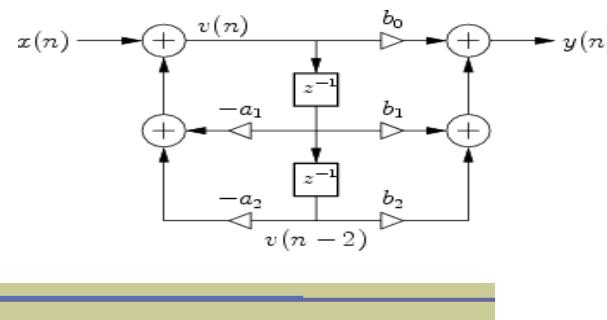
    /* Determina el error e inicia el cálculo del regulador I_PD */
    ek = REFERENCIA - yk;

    /* Calcula la porción integral */
    ik += KI*ek;

    /* Porción derivada actuando sobre realimentación de la posición y */
    dk = KD*(yk - yk1) + (KD2*dk)/256;

    /* Acción total de control I_PD */
    acc = ik - KP*yk - dk;
```

Ejemplo de implementación I_PD paralelo



```
/* Acción total de control I_PD */
acc = ik -KP*yk -dk;

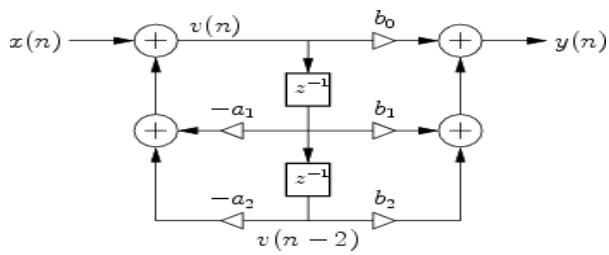
/* Escala, limita el dac */
if(acc < 0) vdac = 0;
else if (acc > 32000) vdac = 125; /* 2V */
    else vdac = (uint8) (acc >> 8);

/* Actualiza el VDAC */
VDAC_SetValue(vdac);

/* Guarda la posicion para la siguiente vez */
yk1 = yk;

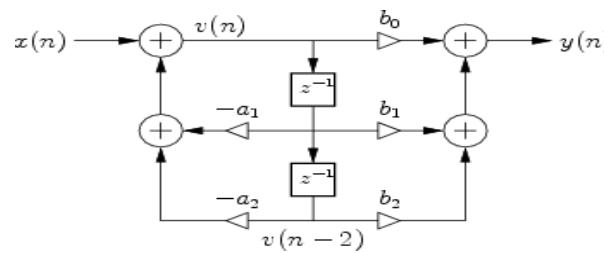
/* Satura el término integral */
if (ik > MAXINTEGRAL)
    ik = MAXINTEGRAL;
else
    if (ik < -MAXINTEGRAL) ik = -MAXINTEGRAL;

/* `#END` */
}
```



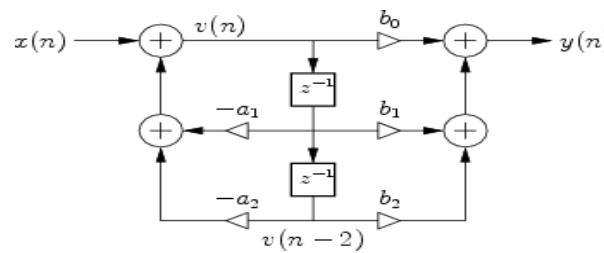
Implementación de la ecuación de diferencias en un computador digital

Implementación: Problemas → ruido

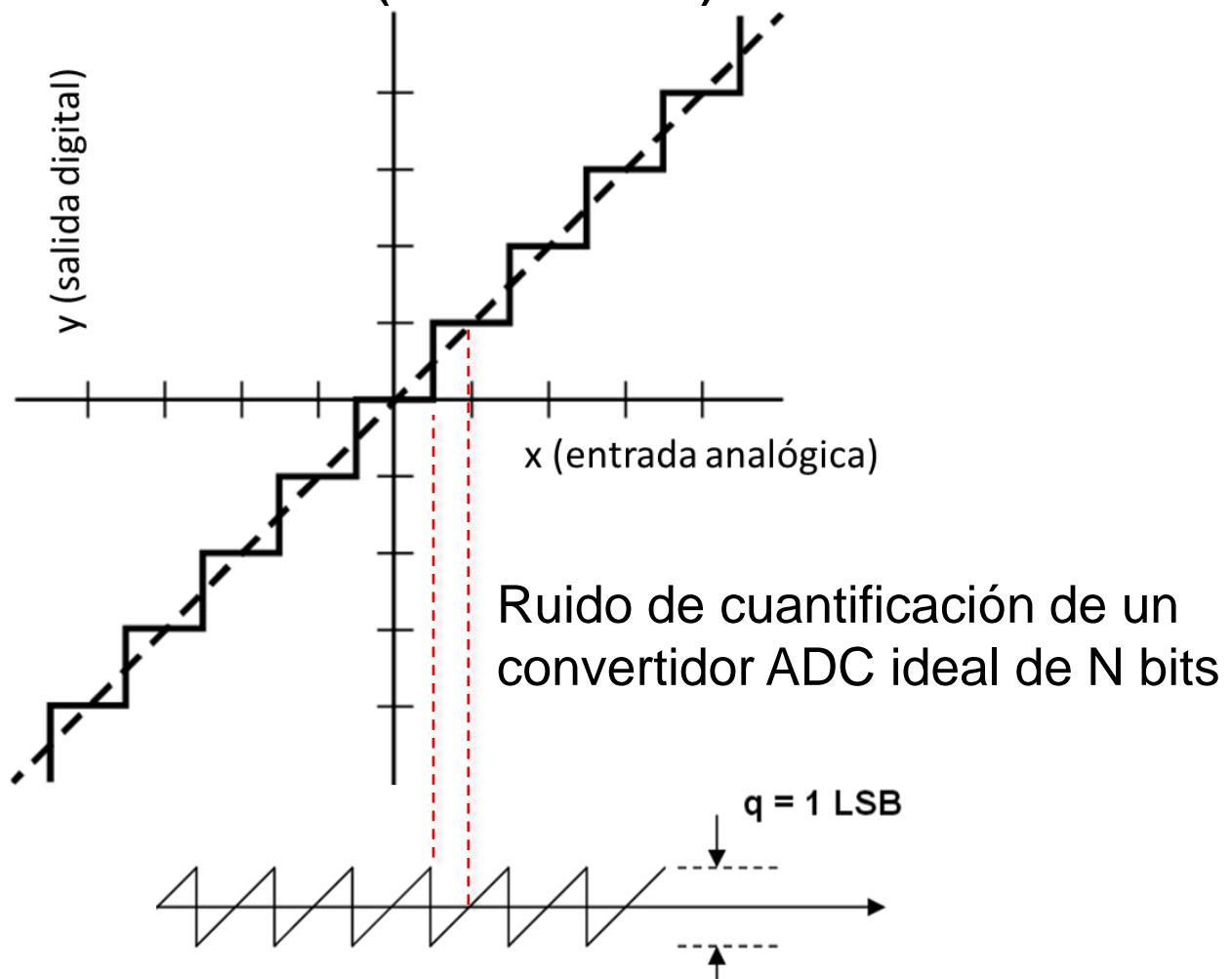


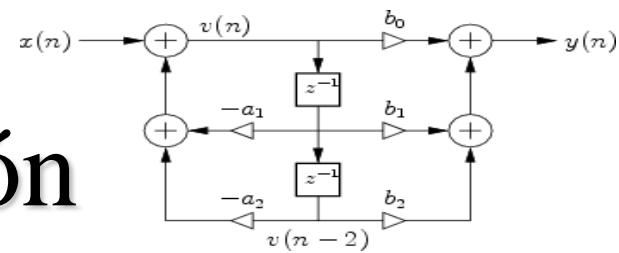
- El ruido o error de redondeo y el desbordamiento (*overflow*) debidos al ancho de la palabra limitado en las operaciones de cálculo en aritmética entera.
- El ruido en la medición y el proceso (planta, realimentación, sensores)
- El ruido de cuantificación

Implementación: Problemas → ruido



■ El ruido de cuantificación (muestreo)



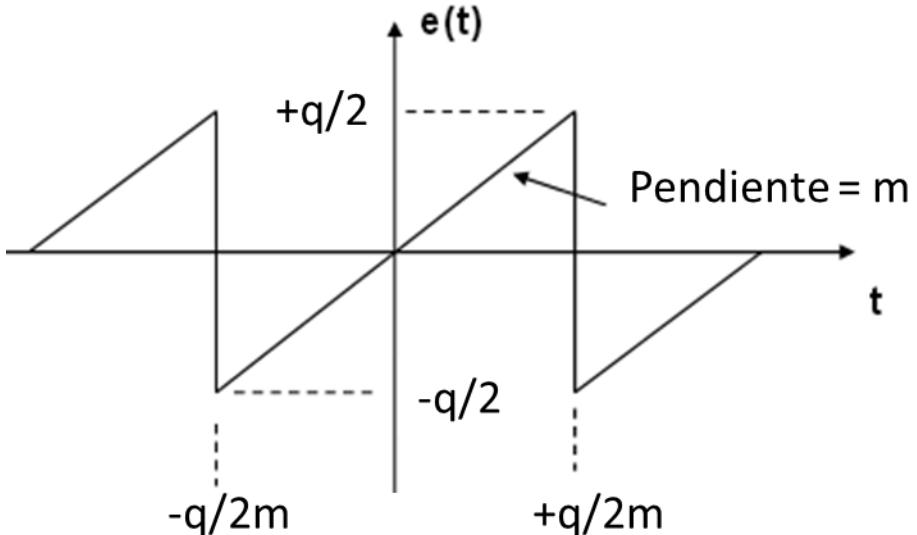


Modelo para el ruido de cuantificación

Error en función del tiempo

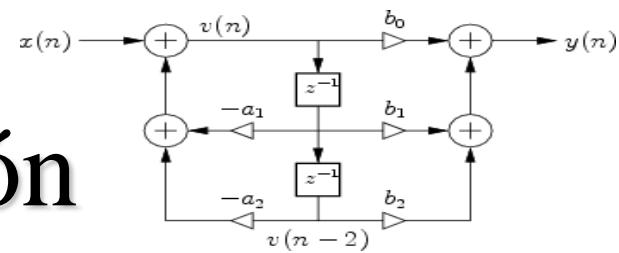
Ecuación para la onda
diente de sierra

$$e(t) = mt, -\frac{q}{2m} < t < \frac{q}{2m}$$



Valor cuadrático medio de $e(t)$

$$\overline{e^2(t)} = \frac{m}{q} \int_{-q/2m}^{+q/2m} (mt)^2 dt = \frac{q^2}{12}$$



Modelo para el ruido de cuantificación

El valor RMS del error de cuantificación

$$\text{Ruido de cuantificación RMS} = \sqrt{e^2(t)} = \frac{q}{\sqrt{12}}$$

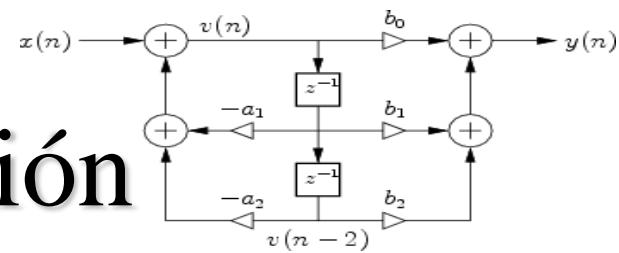
La distribución es aprox. Gaussiana desde 0 hasta $f_s/2$

Se supone una señal sinusoidal de escala completa ($q = \text{LSB}$)

$$v(t) = \frac{q2^N}{2} \operatorname{seno}(2\pi ft)$$

Cuyo valor RMS es

$$v(t)_{rms} = \frac{q2^N}{2} * \frac{1}{\sqrt{2}}$$



Relación señal a ruido de cuantificación

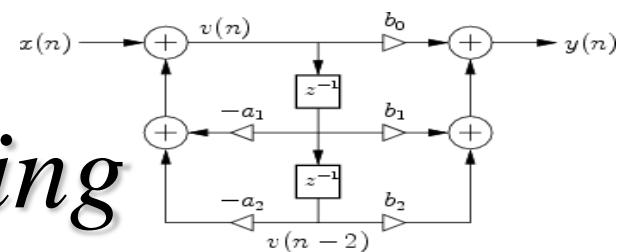
La relación señal a ruido de un convertidor ADC de N bits es:

$$S/R = 20 \log_{10} \left(\frac{\text{Valor RMS de la entrada a escala completa}}{\text{valor RMS del ruido de cuantificación}} \right)$$

$$\frac{S}{R} = 20 \log_{10} \left(\frac{\frac{q2^N}{2\sqrt{2}}}{\frac{q}{\sqrt{12}}} \right) = 20 \log_{10} 2^N + 20 \log_{10} \sqrt{\frac{3}{2}}$$

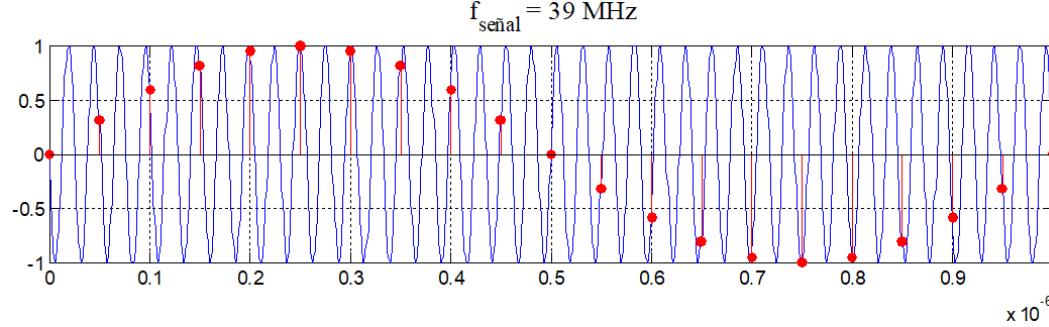
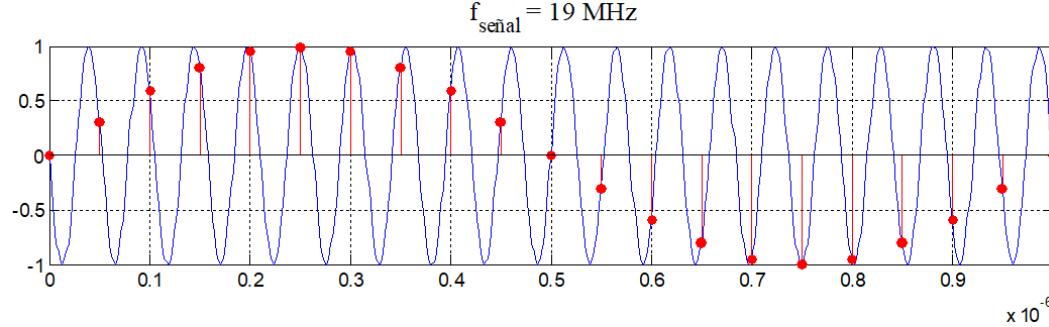
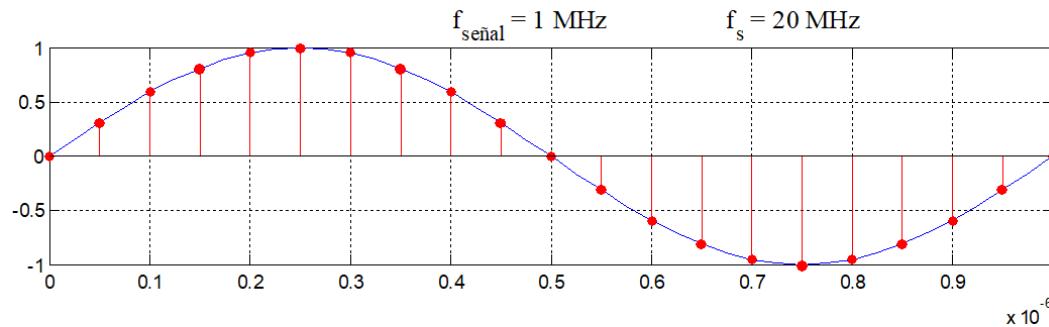
$$\frac{S}{R} = 6.02N + 1.76 \text{ [dB]}$$

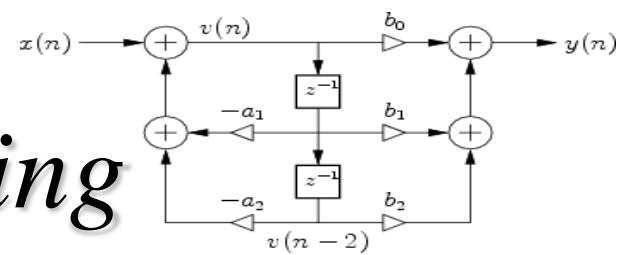
NOTA: Se supone que el ruido de cuantificación no está correlacionado con la señal de entrada



Implementación: Problemas → *aliasing*

Solapamiento (*aliasing*) en el dominio del tiempo (muestreo)

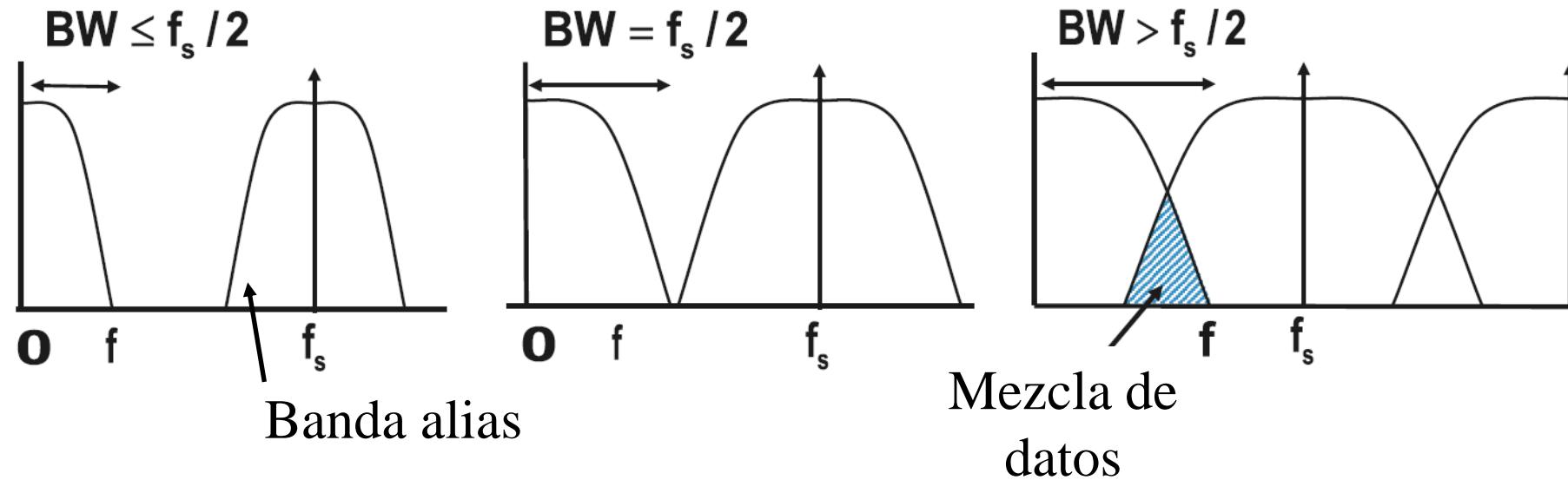




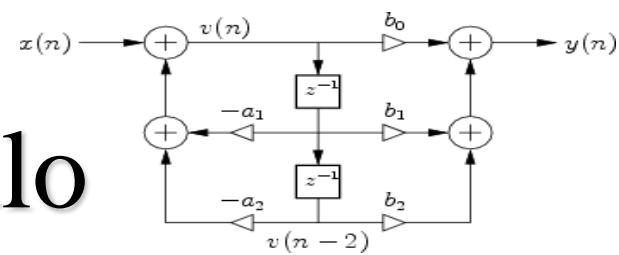
Implementación: Problemas → *aliasing*

Solapamiento (*aliasing*) en el dominio de la frecuencia (muestreo)

Solapamiento de los espectros al aumentar el BW

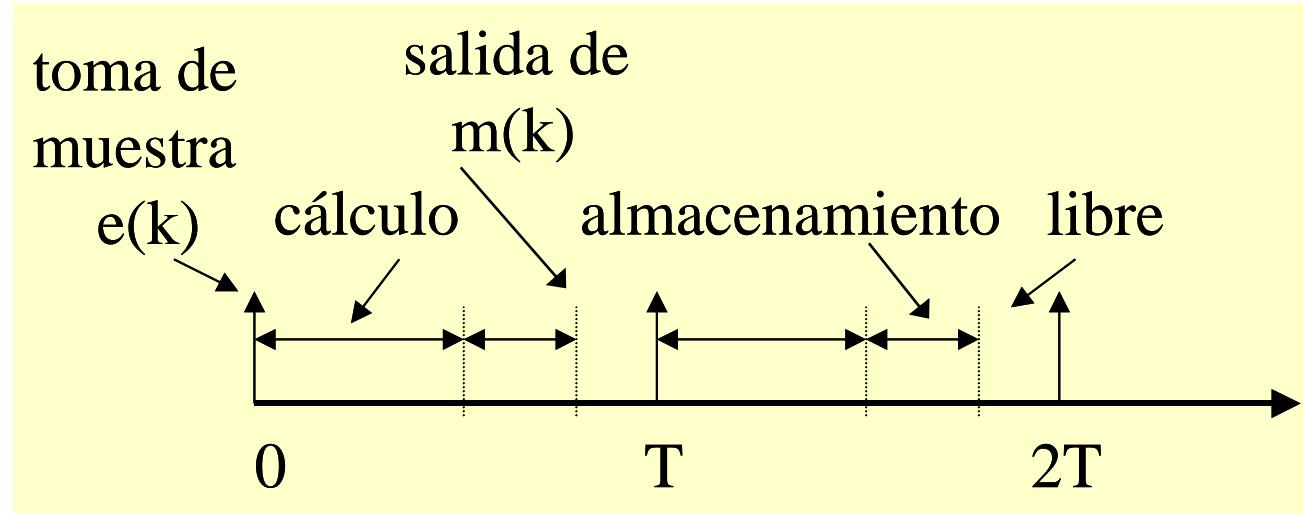


El solapamiento aparece cuando el ancho de banda es mayor que la mitad de la frecuencia de muestreo

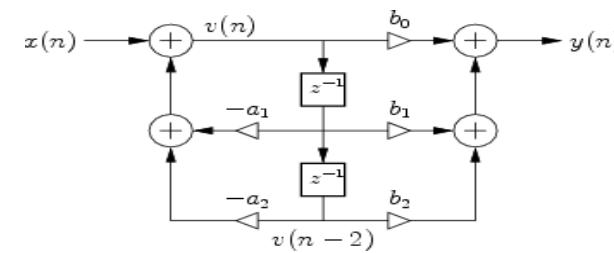


Implementación: Problemas → cálculo

Potencia de cálculo insuficiente como para lograr que el tiempo requerido para el cálculo desde la toma de la muestra $e(k)$ hasta la salida de $m(k)$ sea menor a $\frac{1}{10}$ del periodo T .



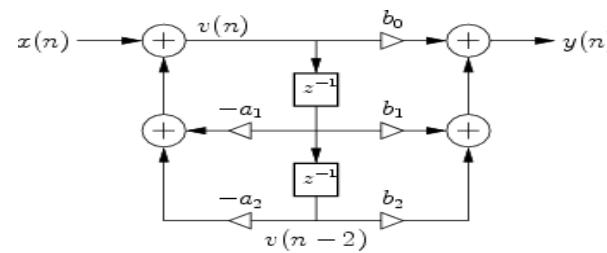
Soluciones (1)



■ Al ruido de redondeo y *overflow*

- Realizar los cálculos intermedios con una exactitud numérica mayor a la del resultado esperado y redondear hasta el final para disminuir el error de redondeo
- Utilizar implementaciones insensibles al *overflow* intermedio como la DF_I

Soluciones (2)



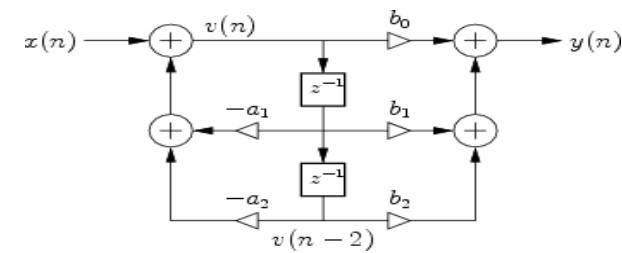
Al ruido en la medición

- Filtro promediador. Tomar varias muestras de la realimentación en rápida secuencia y promediarlas para obtener el valor $y(k)$.
- Filtro de media móvil. Promediar las últimas N muestras de $x(k)$ tomadas. Tiene una frecuencia de corte menor y disminuye el ruido, eso sí, a costa de retrasar la actuación: $retardo = T(N-1)/2$

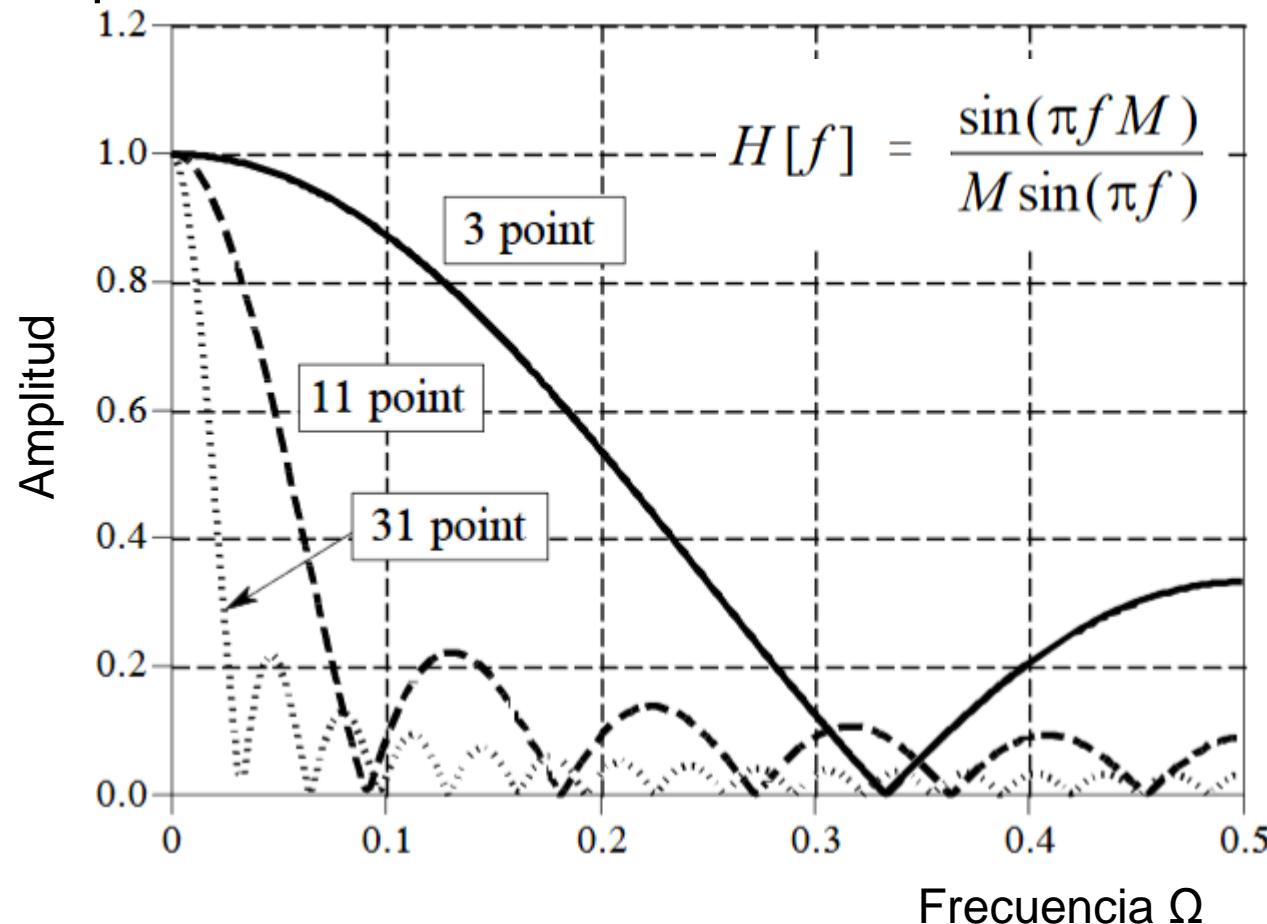
$$H(z) = \frac{1}{N} \sum_{i=0}^{N-1} z^{-i}$$

$$Y(z) = H(z)X(z) \quad y(k) = \frac{1}{N} \sum_{i=0}^{N-1} x(k - i)$$

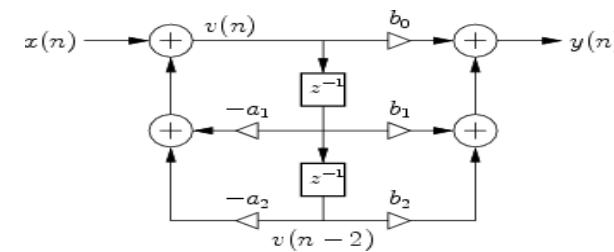
Soluciones (2)



Respuesta de frecuencia del filtro de media móvil



Soluciones (3)

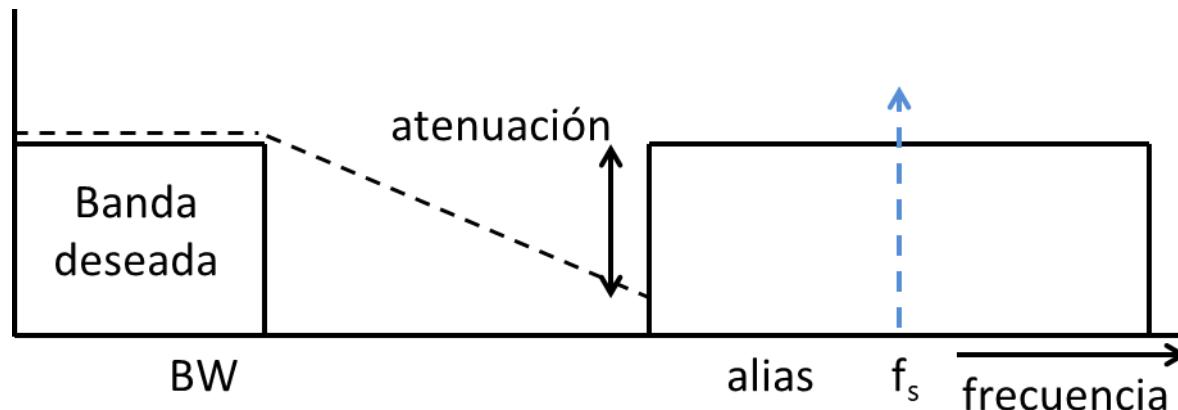


■ Al ruido de cuantificación

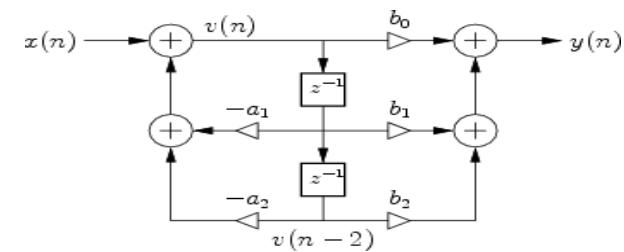
- Aumentar la resolución del convertidor A/D para disminuir el error de cuantificación. $S/N \approx 6\text{dB/bit}$

■ Al solapamiento (*aliasing*)

- Acondicionamiento de la señal con filtrado *antialias*
- Los filtros anti-alias son filtros basabajas de tiempo continuo activos o pasivos con tres regiones: banda de paso, banda de transición, banda de rechazo

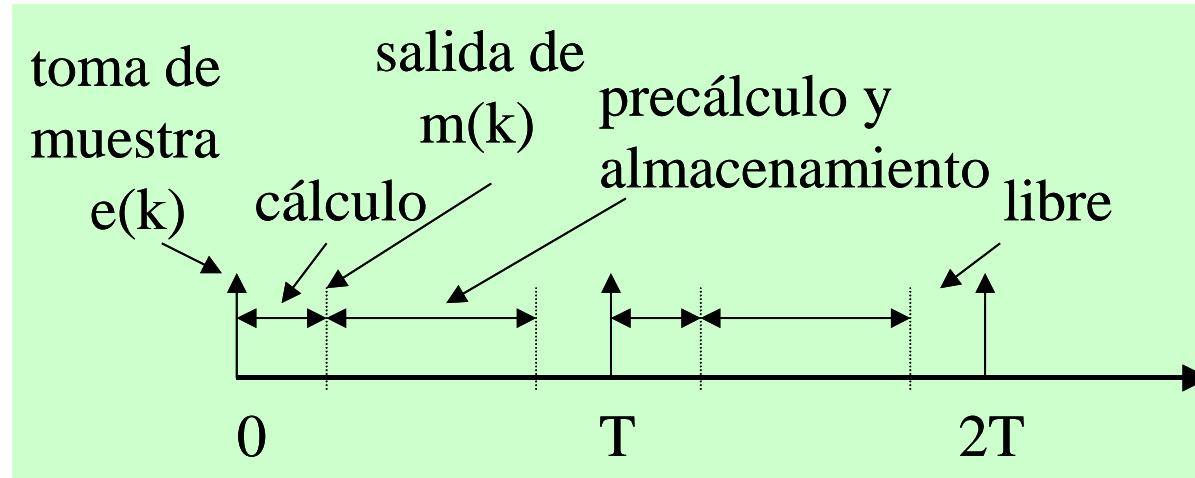


Soluciones (4)

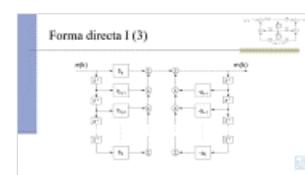


A la potencia de cálculo insuficiente

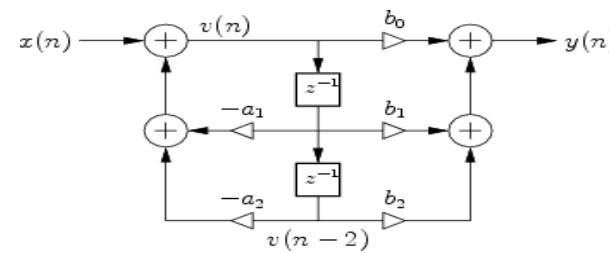
- Utilizar algoritmos más eficientes como el precálculo



- Usar aritmética entera para minimizar el tiempo de cálculo entre la toma de la muestra y la salida
- Utilizar un registro circular en lugar de mover muestras en memoria
- Incorporar el retardo de cálculo dentro del modelo, como un retardo en el sensor, usando la transformada Z modificada.



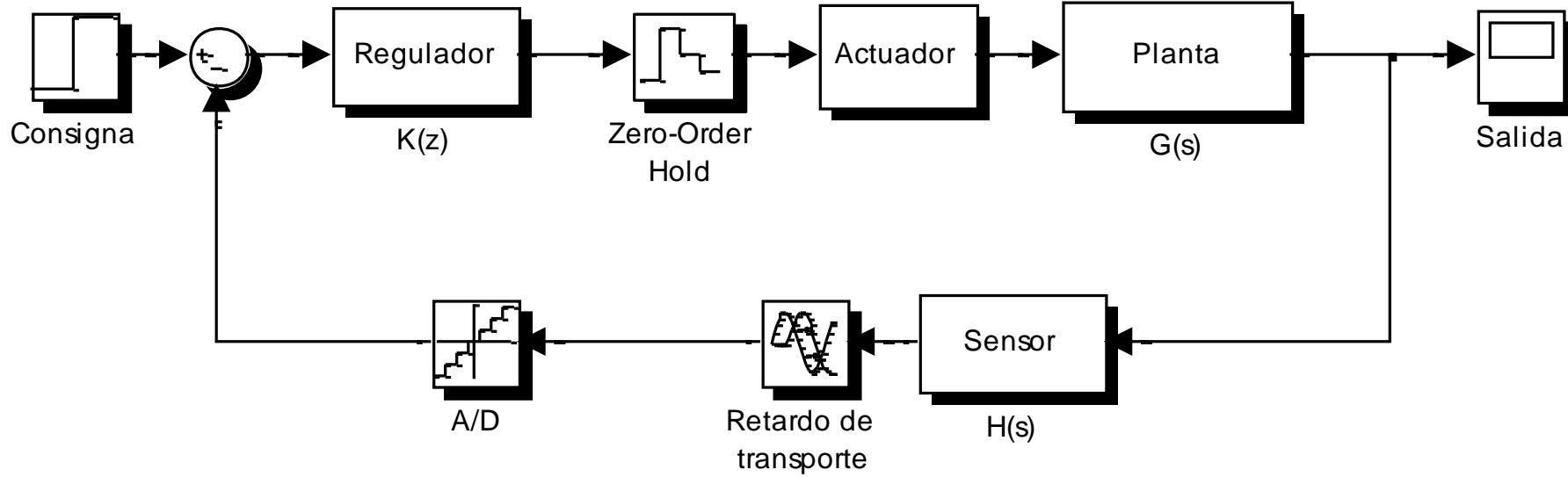
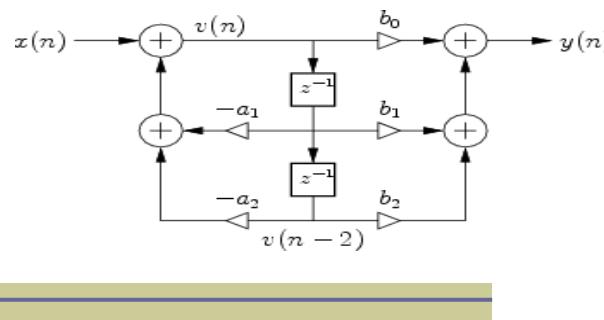
Comparación de algoritmos



```
;SIMPLE
;Cada interrupción del timer
label proc_PID
    in(rk); leo el valor de consigna
    in(yk); leo el valor de la salida
    ek = rk - yk; obtengo el error
    mk = mk_1 + A*ek + B*ek_1
        + C*ek_2;
    mk_1 = mk;
    ek_2 = ek_1;
    ek_1 = ek;
    out(mk);
end;
```

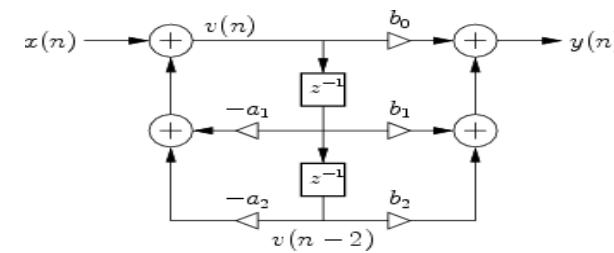
```
;CON PRECÁLCULO
;Cada interrupción del timer
label proc_PID
    in(rk); leo el valor de consigna
    in(yk); leo el valor de la salida
    ek = rk - yk; obtengo el error
    mk = pk + A*ek;
    out(mk);
    mk_1 = mk;
    ek_2 = ek_1;
    ek_1 = ek;
    pk = mk_1+ B*ek_1+ C*ek_2;
end;
```

Sistema de control con retardo para modelar el tiempo de cálculo



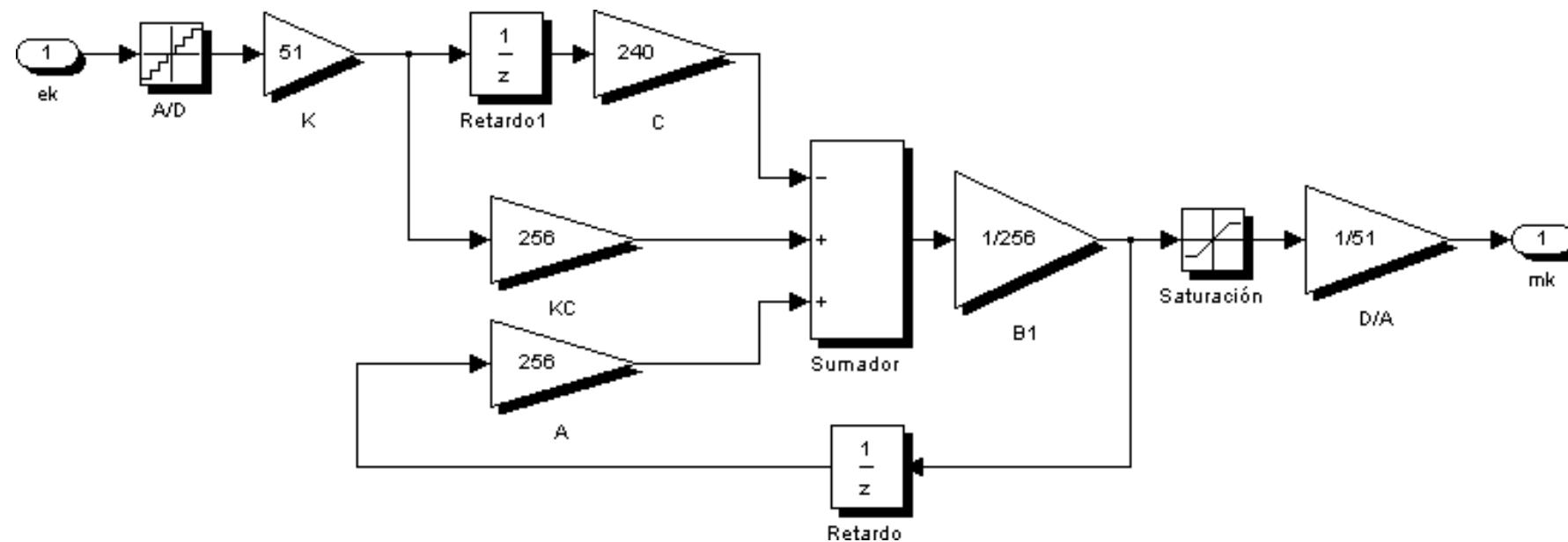
Se utiliza la transformada Z modificada para modelar el tiempo de cálculo γT con $0 < \gamma < 1$

Estructura de una ecuación de diferencias DF-I

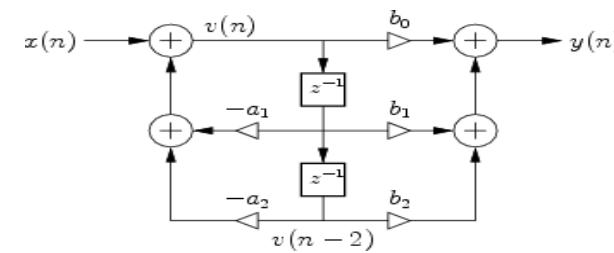


Ecuación de un regulador PI escalado * 256

$$256*m(k) = 256*e(k) + 256*m(k-1) - 240*e(k-1)$$

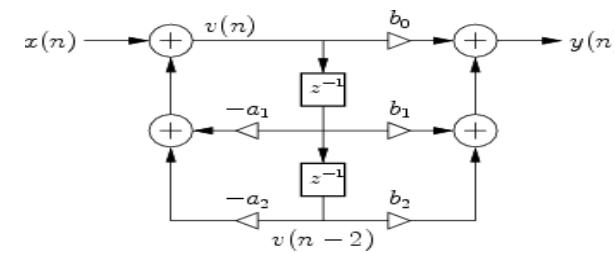


Tareas



- Investigue sobre el acondicionamiento de señales por filtrado de forma analógica y de forma digital.
 - Filtros antisolapamiento
 - Filtros suavizadores
 - En general sobre filtros digitales IIR y FIR
- Investigue sobre formas de representación de números reales con enteros y sus operaciones.
- Investigue sobre transformada Z modificada
- Investigue sobre variaciones al regulador PID (PID 2DoF, I_PD, PI_D, PID, PD con filtro derivativo.)
- Investigue sobre otras formas adicionales de combatir el *wind-up*, cuando se utilizan integradores (I, PI, PID).

Referencias



- Kuo, Benjamin C.. „**Sistemas de Control Automático**“, Ed. 7, Prentice Hall, 1996, México.
- http://www.sec.upm.es/docencia/plan_92/scii/descarga_SCII/pdf/antiwindup.pdf
- Motorola DRM007/D: “BLDC Motor Control Board for Industrial and Appliance Applications” Designer Reference Manual, 2003
- <http://www.ie.itcr.ac.cr/einteriano/control/Laboratorio/5.anti-windup-pdf>
- <http://www.ie.itcr.ac.cr/einteriano/control/Laboratorio/100 Trucos en el Laboratorio de Control Automatico Extracto.pdf>
- [https://en.wikipedia.org/wiki/Q_\(number_format\)](https://en.wikipedia.org/wiki/Q_(number_format))
- http://en.wikipedia.org/wiki/Advanced_Z-transform
- http://en.wikipedia.org/wiki/Anti-aliasing_filter
- <https://www.mathworks.com/help/signal/examples/signal-smoothing.html>
- Proakis John G., Manolakis, Dimitris G.. „**Tratamiento Digital de Señales**“, 3^a Ed., Prentice Hall, 1998, España.