

60



UNIVERSIDAD NACIONAL AUTONOMA DE MÉXICO

FACULTAD DE INGENIERIA

**ALGORITMOS DE CONTROL
DITAL DIRECTO**

T E S I S

PARA OBTENER EL GRADO DE :

DOCTOR EN INGENIERIA

P R E S E N T A

GUINEA TRIGO JESÚS ALEJANDRO

MÉXICO

**TESIS CON
FALLA DE ORIGEN**



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

TESIS CON FALLA DE ORIGEN

375

01149

60

T E M A :

Algoritmos de Control
Digital Directo.

Subtema.

Descripción e implementación por
medio de microprocesadores

ano? 72

Jesús Alejandro Guinea Trigo.

TESIS CON
FALLA DE ORIGEN

1.-	Introducción	1.
2.-	El trabajo del Procesador. Analisis.....	
	- Programa Adquisitor.....	3.
	- Algoritmos PI, PID, de Posición y veloci-- dad.....	4.
	- Configuraciones Serie y Paralelo.....	
3.-	Controlador Adaptado al Proceso. Diseño.....	8.
	- Función de Transferencia de Pulso.....	
	- Diseño del Controlador.....	
	- Algoritmo Dead beat, Respuesta para regula ción y servos. (1 er. orden).....	10.
	- Comportamiento de la variable manipulada... 12.	
	- Algoritmo para modelos de 2º orden	
	- Algoritmo con la acción K.libre de oscila-- ciones.....	
	- Algoritmo para 1 er orden con retraso.....	
4.-	Algoritmo de Dahlin.....	
	- Algoritmo para un proceso de 1 er. orden con tiempo muerto.....	14.
	- Algoritmo para 2ºord n con retraso.....	19.
	- Algoritmo de Dahlin implementado con modos convencionales.....	
5.-	Metodo de Kalman.....	
	- Obtención del algoritmo del controlador....	
	- Algoritmo para 1 er. orden con retraso.....	21.
	- Algoritmo para 2 orden con retraso, imple- mentado con modos convencionales.....	23.

6.-	Algoritmo De estabilidad en tiempo finito FTS con retroalimentación de variables de estado..	
	- Algoritmo para 1 er. orden con retraso.....	25.
	- Algoritmo incremental para 1 er. orden con retraso.....	26.
	- Algoritmo para 2 orden.....	27.
	- Algoritmo evitando retro de variable de es- tado (1 er. orden con retraso).....	
7.-	Algoritmo Observador.....	29.
8.-	Algoritmo Predictor.....	30.
	- Proceso de 1 er. orden	
9.-	Control Coalimentado.....	34.
	- Acción Suma y Multiplicación.....	
10.▼	Control en Cascada.....	35.
	- Ejemplo de Aplicación DDC , para retro/ali- mentación de un cambiador de calor en li- nea.....	35.
	- Implementación DDC y diagrama de flujo.....	

ANEXO II.

- A Diagramas de flujo simplificados de los algoritmos presentados. . .
- B Uso de un microprocesador para resolver el algoritmo Fl.
- C Consideraciones de tiempo en la implementación de los algoritmos.

1.- Introducción.

La aplicación de los procesadores electrónicos digitales al control de procesos se ha implementado con buenos resultados en la última década.

El incentivo fundamental para el uso de control digital es el ahorro que aporta al suplantarse a gran número de lazos analógicos, cosa que es discutible debido a gastos de programación y de sistemas de seguridad.

Por otro lado desde el aspecto del control, el procesador puede emular las tareas de los sistemas convencionales, y es capaz de realizar algoritmos de optimización cuya complejidad sería difícil de lograr con elementos analógicos.

Las formas de adaptar un procesador a los procesos continuos se ha realizado a varios niveles desde Almacenamiento de datos, pasando por el control Supervisorio hasta una configuración jerárquica que integra desde trabajos de control directo, supervisión y manejo de datos.

Los procesadores que realizan el DDC están en contacto a primer término con las variables del proceso. En los lazos de medición y de control.

Los procesadores mini's han sido puestos en los lazos DDC haciendo posible las técnicas de Multiplex para controlar gran número de variables físicas y actualmente el advenimiento de los microprocesadores con sus cualidades de tamaño y costo ofrece a los inge-

nieros de sistemas un dispositivo hoc para la creación de es
taciones de control local independientes.

Las facilidades de programación, interconexión y confiabilidad -
abren como nunca antes el horizonte del control Digital Directo.

Los sistemas quedan integrados por: una interface de Medición, acondi-
cionamiento y codificación, Procesador-controlador e interface a
elementos finales de control.

El comportamiento y la precisión de los elementos del sistema se
proyectan en el funcionamiento óptimo del mismo, de modo que se de-
be atender a los errores, no linealidades y demás fenómenos propios
de los elementos.

v.g. el error medio cuadrático de cuantificación en un conversor
analógico digital que está dado por: Ref 1.

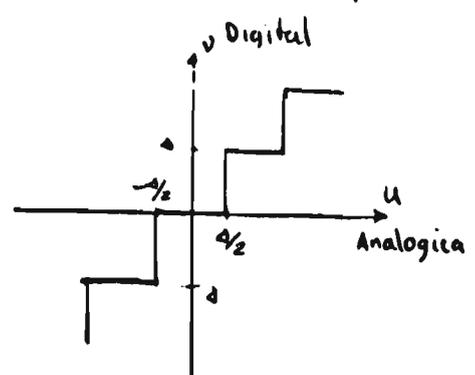


Fig. 1 Función de Transferencia del convertidor A/D

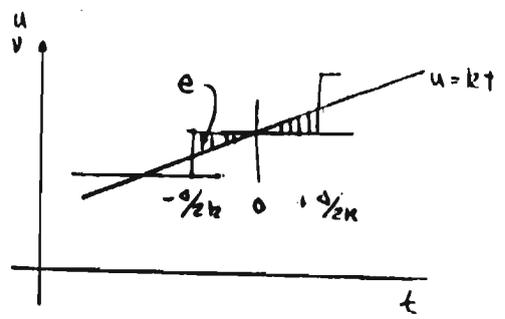


Fig. 2 Error de cuantización para una señal rampa.

Este error puede despreciarse para una codificación de 8 bits de-
bido a que es mucho menor que el provocado por los elementos de -
medición (1% del rango).

$$\bar{e}^2 = \frac{1}{\Delta/k} \int_{-1/2k}^{1/2k} kt^2 dt \approx \frac{\Delta^2}{12} \quad (o)$$

2. El trabajo del Procesador. ANALISIS

El procesador se encarga de realizar cálculos basados en un programa específico que contiene un algoritmo o tarea a seguir.

La tarea más pesada, que utiliza mayor cantidad de memoria y más tiempo de máquina es el programa adquisitor que contiene rutinas de entrada y salida para el manejo de las variables del campo y rutinas para detección de alarmas y de diagnóstico.

Los cálculos que efectúa el procesador los hace eventualmente bajo "interrupciones" del reloj de tiempo real atendiendo con más frecuencia a las variables rápidas como flujo y menos frecuentemente a aquellas como temperatura y composición. Ref. 4

El procesador recibe los datos de los puntos de medición que han sido muestreados y codificados propiamente (A/D). En base al estado de estos y por medio de un algoritmo, obtendrá cuál es la acción que se debe tomar misma que entregará por sus puertos de salida a un elemento conversor quién recibiendo datos discretos, los transforma a continuos para aplicarlos al proceso en la manipulación de una o varias de sus variables. fig 3

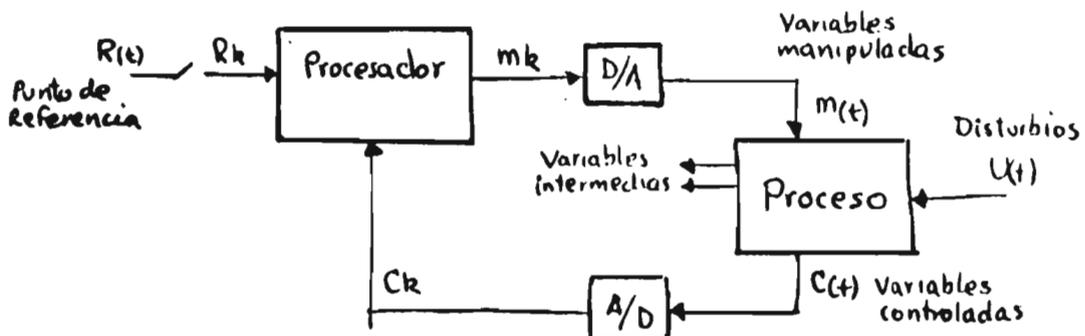


Fig 3 Presentación general del sistema

La idea del control de una planta se resume en el mantener un estado del proceso, identificado por los valores que presentan sus características físicas, en un punto estático o con un comportamiento v.g. La obtención de un contenido de carbono en la producción de acero.

La obtención de una señal de frecuencia constante en un oscilador.

Las características del proceso pueden cambiar debido a parámetros internos o externos; sin embargo, el sistema de control tiene por objeto el hacer que esos cambios no afecten la operación deseada.

Así el controlador debe percatarse cuando el sistema esta fuera de orden, obteniendo una señal de error e a partir de la cual calculará cuál debe ser acción a tomar comandando la variable M que manipulará al proceso.

El trabajo del controlador se puede expresar como una función de transferencia. v.g. PI, PID. que debido a que relaciona variables discretas puede expresarse en ecuaciones de diferencias como:

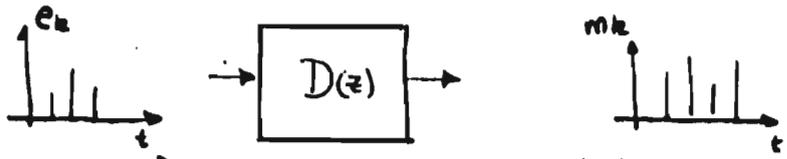


Fig 4 Compensador Digital - Controlador

A. Algoritmo PID (Posición-Paralelo)

$$m_k = K_c \left[e_k + \frac{T}{\delta_i} \sum_{j=1}^k e_j + \frac{\delta_d}{T} (e_k - e_{k-1}) \right] \quad 1$$

$$m_k = K_c' \left[e_k + \frac{T}{\delta_i} \sum_{j=1}^k \frac{e_j - e_{j-1}}{2} + \frac{\delta_d}{T} (e_k - e_{k-1}) \right] \quad 2$$

se muestra una integración o suma finita rectangular en la ecuación 1 y una suma trapezoidal en la 2

Estas ecuaciones representan a un algoritmo PID de "posición" pues el cálculo de m_k depende sólo del error. Su implementación es paralelo fig 5

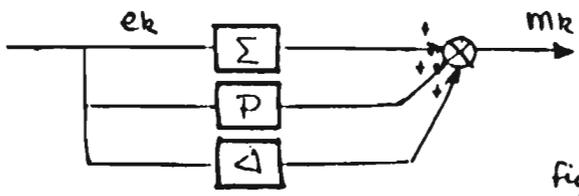


Fig 5 Controlador PI Paralelo

Se puede lograr un algoritmo de velocidad incremental donde sólo se obtenga la variación de manipulación que debe hacerse sobre la ya existente. Esto permite un inicio del algoritmo sin necesidad de un valor de referencia M_R .

La acción derivada es indeseable cuando hay cambios bruscos en el set point o en la física del proceso. En el primer caso del algoritmo puede alterarse dejando a la acción derivada actuar en función de la variable de salida únicamente o eliminarse totalmente.

Algoritmo PI paralelo de velocidad fig 5', 5''

$$\Delta m_k = K_c (e_k - e_{k-1}) + K_i e_k$$

donde $K_i = \frac{T}{\delta_i} K_c$

su representación en transformación z es

$$\frac{M(z)}{E(z)} = D(z) = K_c \frac{z - (1 - \frac{T}{\delta_i})}{z - 1}$$

Algoritmo PID paralelo sin saltos derivados fig 6, 6'

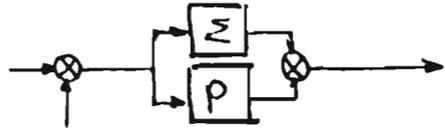
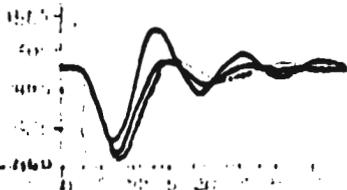


Fig 5' Algoritmo PI Paralelo.

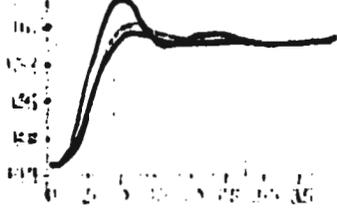
Fig 5''

ALGORITMO PI

• Sintonización para cambios en la carga.



• Sintonización para cambios en la referencia



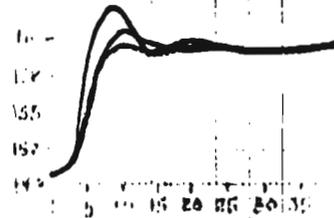
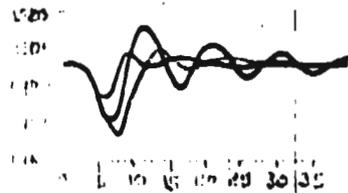
— Sintonización óptima en ITAE

- - - tiempo muerto equivalente $\theta+T$

... " " " $\theta+T/2$

Fig 6'

ALGORITMO PID



$$5) \quad m_k = m_{k-1} + K_c (C_k - C_{k-1}) + K_i (C_{k-1} - r_k) + K_D (C_k - 2C_{k-1} + C_{k-2})$$

$$6) \quad D(z) = K_c \frac{(1+K_D)z^2 - (1+2K_D-K_i)z + K_D}{z(z-1)} \quad 6$$

donde $K_D = K_c \delta_0 / T$

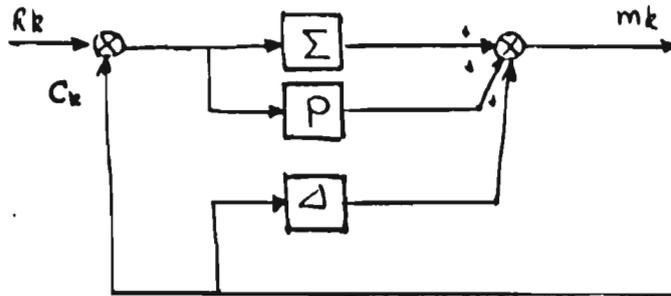


Fig 6 Algoritmo PID paralelo.

En sistemas ruidosos el término de diferencia puede mejorarse definiendo una diferencia a 3 puntos por :

$$7) \quad \frac{\Delta C}{\Delta t} = \frac{C_n - C_{n-3} + 3C_{n-1} - 3C_{n-2}}{6T}$$

El algoritmo DDC puede ser programado para aumentar su eficiencia v.g frente a problemas de ruido en altas frecuencias la acción derivada se puede limitar compensando con una red que sature su acción. (Ref 22 pag 44, Ref. 7)

También mediante acotamiento de la acción integral puede preverse la saturación del elemento final de control lográndose así evitar sobretiro en la respuesta del sistema.

C Algoritmos PID Serie (12)

En el uso de algoritmos en DDC el tipo de serie,

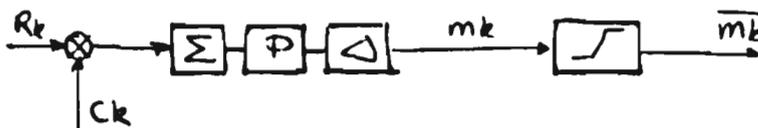
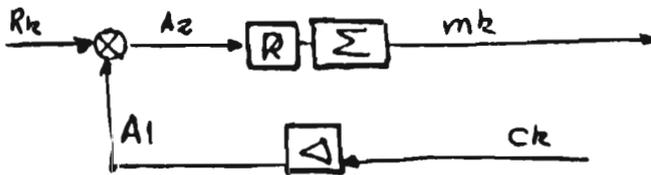


Fig 7. Algoritmo PID Serie

es recomendado debido a que el control que proporciona es idéntico al realizado por la contraparte analógica y es preferido por la gente que esta acostumbrada a aquel.

En la figura se muestra el diagrama de bloques del sistema con un limitador de movimiento. Sacando la acción "diferencia" del lazo, este queda: fig 8



Algoritmo con la acción diferencia en la salida
fig 8

La programación del algoritmo se presenta a continuación:

8)

$$\begin{aligned}
 A_{1k} &= C_k + K_0 (C_k - C_{k-1}) \\
 A_{2k} &= A_{1k} - R \\
 m_k &= m_{k-1} + K_i A_{2k} + K_i (K_i - 1) A_{2k-1} \\
 \bar{m}_k &= e^{-\tau K_L} (\bar{m}_{k-1} - m_k) + m_k
 \end{aligned}$$

3.- Controlador adaptado al Proceso. DISEÑO

Los controladores citados logran un comando con buenas características las cuales pueden observarse tanto en el dominio del tiempo como de la frecuencia. Sin embargo los parámetros contenidos en el Algoritmo tienen que ver con el proceso de que se trate, debiéndose así sintonizarlos para obtener resultados óptimos. Los métodos de sintonización de los parámetros Kp, KI, Kc, KL etc. se basan en consideraciones empíricas obtenidas de la respuesta en frecuencia y tiempo del sistema y de principios de un buen control en las salidas del mismo.

Así el algoritmo de control se obtiene en función de la transferencia del proceso $HG(z)$ y de la respuesta deseada $\frac{C(z)}{R(z)}$ frente a cambios en el Set point ó $\frac{C(z)}{U(z)}$ para alteraciones en la carga o disturbios.

La función de transferencia de pulso para el par Retensor-Proceso es:

$$9) \quad HG(z) = \mathcal{Z} \left[\mathcal{Z}^{-1} G(s) H(s) \right]$$

donde:

H(s) función de transferencia del Retensor

G(s) función de transferencia del Proceso

HG(z) Transferencia discreta del par Retensor-Proceso

por regla de Mason la respuesta del sistema en función del punto de referencia R(z) y de los disturbios U(s) es:

$$10) \quad C(z) = \frac{HG(z) D(z)}{1 + HG(z) D(z)} R(z) + \frac{U(z)}{1 + HG(z) D(z)}$$

*Nota: G(s) puede ser representada en la mayoría de los casos por modelos de 1° y 2° orden con retraso.

$UG(z)$ es la transferencia a pulso del par Disturbios-Proceso.

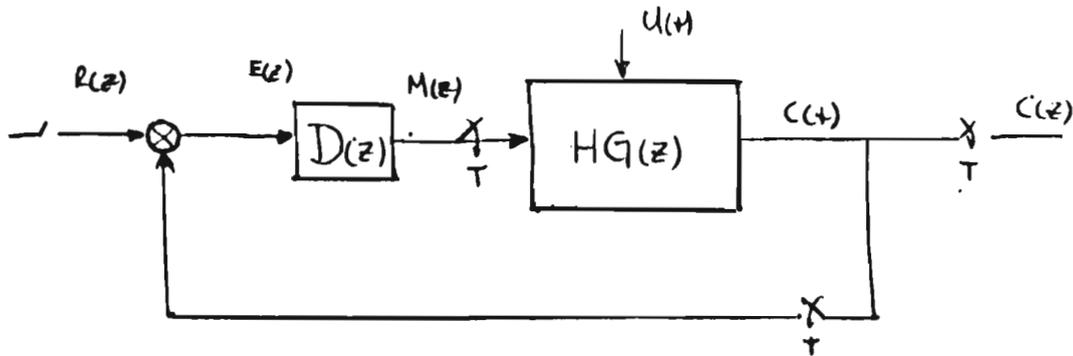


Fig 9
Lazo de Control Digital Típico de un Proceso.

Si consideramos $U(s) = 0$ podemos llegar a la fórmula para obtener el algoritmo del controlador $D(z)$:

$$11) \quad D(z) = \frac{1}{HG(z)} \frac{C(z)/R(z)}{1 - C(z)/R(z)}$$

cuya forma canónica es:

$$12) \quad D(z) = \frac{b_0 + b_1 z^{-1} + \dots + b_n z^{-n}}{1 + a_1 z^{-1} + \dots + a_n z^{-n}}$$

La transferencia de malla cerrada $C(z)/R(z)$ nos indica la forma y calidad del control que deseamos. Esto nos lleva a definir índices de calidad en función de minimizar las desviaciones respecto al comportamiento deseado.

v.g. Se puede desear una respuesta a escalón sobre amortiguada con una razón de decrecimiento de un cuarto, lo cual nos dará el valor de K_c en función del periodo de muestreo T .

Para un diseño de respuesta a disturbios en la ecuación (10) $R_d = 0$ tenemos:

$$13) \quad C(z) = \frac{UG(z)}{1 + HG(z) D(z)}$$

$$14) \quad D(z) = \frac{UG(z)/C(z) - 1}{HG(z)}$$

Nótese que para el diseño en función de U se debe especificar el comportamiento $C(z)$ respecto a cada tipo de disturbio.

El periodo de muestreo T esta relacionado con varios parámetros, a saber:

- La mayor constante de tiempo del proceso
- La capacidad del sistema de muestreo
- Tiempo muerto del proceso.

Los criterios usados para el cálculo del periodo T son muy variados y dependen del tipo de control y de proceso que se trate (2)

$$15) \quad \text{v.g.} \quad \frac{T}{\tau} = - \ln \frac{1}{2}$$

El algoritmo de control es realizable sólo si el tiempo muerto del proceso no excede a T .

D. Algoritmo Dead beat.

En el control digital las respuestas se expresan válidas en los tiempos de muestreo. Este algoritmo se representa por una salida que se estabiliza en un número $N + 1$ de tiempos de muestreo. Para cambios en el punto de referencia tenemos:

$$16) \quad \frac{C(z)}{R(z)} = z^{-(N+1)}$$

$$17) \quad D(z) = \frac{1}{HG(z)} \frac{z^{-(N+1)}}{1 - z^{-(N+1)}}$$

Si $\frac{C(z)}{R(z)} = z^{-1}$ y $HG(z)$ es un proceso con un polo α y un retensor

de orden cero tenemos: $G(s) = \frac{1}{s+1}$

$$18) \quad HG(z) = \frac{1-\alpha}{z-\alpha} \quad \alpha = e^{-T/\tau}$$

$$19) \quad \Rightarrow \quad \boxed{D(z) = \frac{z-\alpha}{(1-\alpha)(z-1)}}$$

este controlador $D(z)$ se puede implementar con un control PI.

En las referencias se presentan ejemplos de diseño para una planta modelada por $G(s) = \frac{1}{s+1}$, para cambios en la carga del tipo escalón, es decir un diseño ante disturbios. Ref 8

20)
$$U(z) = \frac{z(1-\alpha)}{(z-1)(z-\alpha)}$$

21)
$$D(z) = \frac{(1+\alpha)z - \alpha}{(1-\alpha)(z-1)}$$

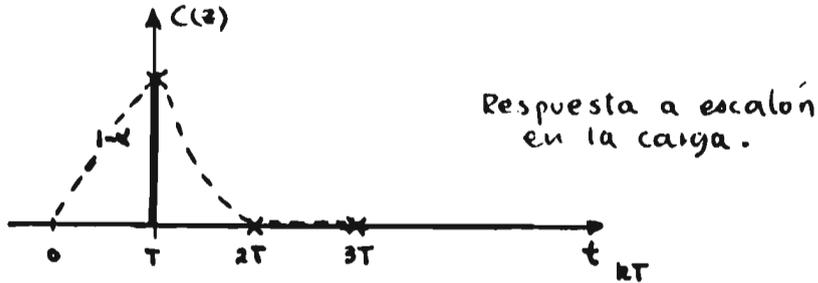


fig 9

Debido a que el controlador tarda ^{como} máximo, un periodo en percatarse del disturbio. La respuesta óptima deseada será.

22)
$$C(z) = (1-\alpha) z^{-1}$$

Estos algoritmos prueban ser eficientes también en su respuesta ante los cambios para los que no han sido diseñados, muestra de esto es que el diseño de la ecuación 21 se comporta bien ante un cambio en el set point. Ref 8

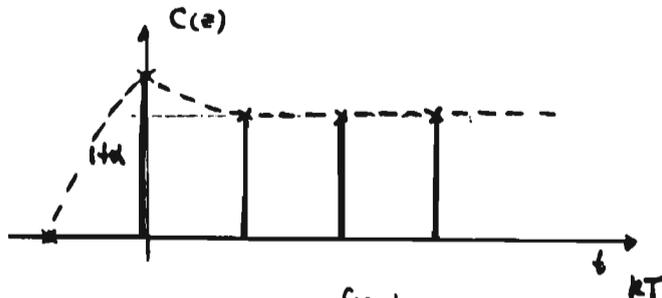


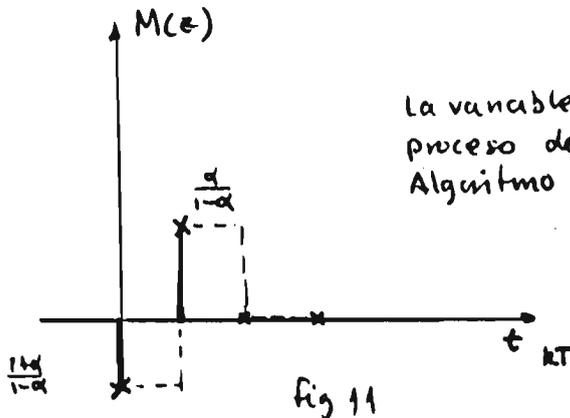
fig 10

Respuesta a escalón en $R(t)$

Hasta ahora el diseño del controlador $D(z)$ ha sido para cumplir una respuesta en el proceso, sin embargo esto puede crear un comportamiento brusco en la variable manipulada. Observémos:

$M(z)$ en el diseño de la ecuación 20 :

$$23 \quad M(z) = -\frac{1+\alpha}{1-\alpha} + \frac{\alpha}{1-\alpha} z^{-1}$$



La variable manipulada en un proceso de 1^{er} orden Algoritmo Dead beat.

Controlador en un proceso de 2^o orden

Tenemos: 24 $H(z) = \frac{(1-\alpha-\alpha T/\delta)z - \alpha(1-\alpha-T/\delta)}{(z-\alpha)^2}$

para FTS $\frac{C_m}{R(s)} = z^{-1}$

$$25 \quad D(z) = \frac{(z-\alpha)^2}{(z-1)[(1-\alpha-\alpha T/\delta)z - \alpha(1-\alpha-T/\delta)]}$$

éste es un algoritmo que no sigue una forma PID, es en éstos, donde la gran flexibilidad de DDC se explyta.

En particular la solución de ec.22 produce oscilaciones en la variable manipulada; sin embargo un rediseño del FTSC de la forma dada puede eliminar el problema. Propongamos una respuesta más suave y calculemos el controlador para obtenerla.

$$26 \quad C(z) = \gamma z^{-1} + z^{-2} + \dots$$

$$27 \quad R(z) = \frac{z}{z-1}$$

quedando el controlador como.

$$28) \quad D(z) = \frac{(z-\alpha)^2}{(z-1)[(1-\alpha)^2 z - \alpha(1-\alpha-T/b)]}$$

Para un proceso de 1er. orden con retraso tenemos (para $L = NT$)

$$29) \quad H(z) = \frac{z^{-N} (1-\alpha)}{(z-\alpha)}$$

siendo $\frac{C(z)}{R(z)} = z^{-(N+1)}$

$$30) \quad D(z) = \frac{1}{1-\alpha} \frac{1-\alpha z^{-1}}{1-\alpha z^{-(N+1)}}$$

La variable manipulada

$$M(z) = \frac{1}{1-\alpha} - \frac{\alpha}{1-\alpha} z^{-1} \quad 31)$$

$D(z)$ cancela un polo de la planta en este algoritmo, de modo que se pueden presentar modos incontrolables. Ref 3
La respuesta usando estos algoritmos puede verse en la fig 12

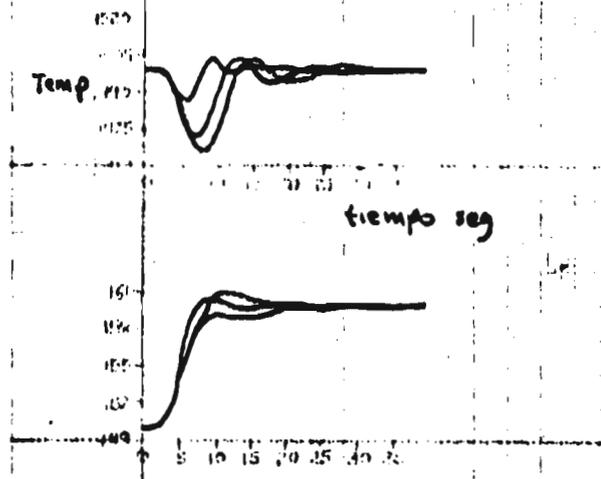
4.- Método de Dahlin. E

El algoritmo FTS para convergir en un intervalo de muestreo no logra resultados en la mayoría de los procesos industriales. El problema está en la elección de C/R para FTS. El método de Dahlin propone que el sistema completo C/R responde como un modelo de 1er orden con retraso

$$32) \quad C(s) = \frac{e^{-\theta s}}{\lambda s + 1} R(s)$$

Fig 12

Respuesta usando Dead beat



- Algoritmo Dead beat
- Algoritmo PID optimo
- Algoritmo PI optimo

donde λ es la constante de tiempo de Dahlin.

Podemos escribir:

$$33 \quad \frac{C(z)}{R(z)} = \frac{(1 - e^{-T/\lambda}) z^{-(N+1)}}{(1 - e^{-T/\lambda} z^{-1})}$$

de donde :

$$34 \quad D(z) = \frac{1 - (1 - \beta) z^{-(N+1)}}{H_G(z) [(1 - \beta z^{-1}) - (1 - \beta) z^{-(N+1)}]} \quad \beta = e^{-T/\lambda}$$

La elección de λ se hace sobre la sintonización del sistema fig 13

Ref 2

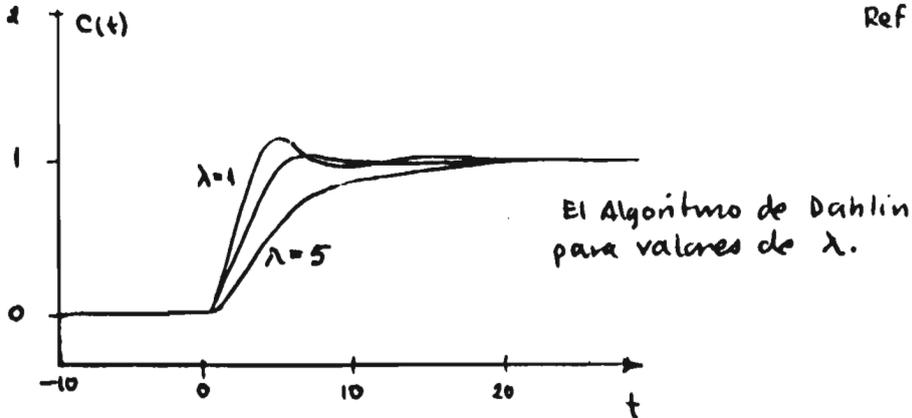


fig 13

Con el uso del algoritmo de Dahlin, el problema de restringir el periodo de muestreo T a valores mayores que el tiempo muerto del sistema, queda solventado.

Ej. Dahlin en un proceso con tiempo muerto 1º orden

$$35 \quad G(s) = \frac{k e^{-Ls}}{s + 1}$$

de ecuación 33 y para H_1 retensor de orden cero

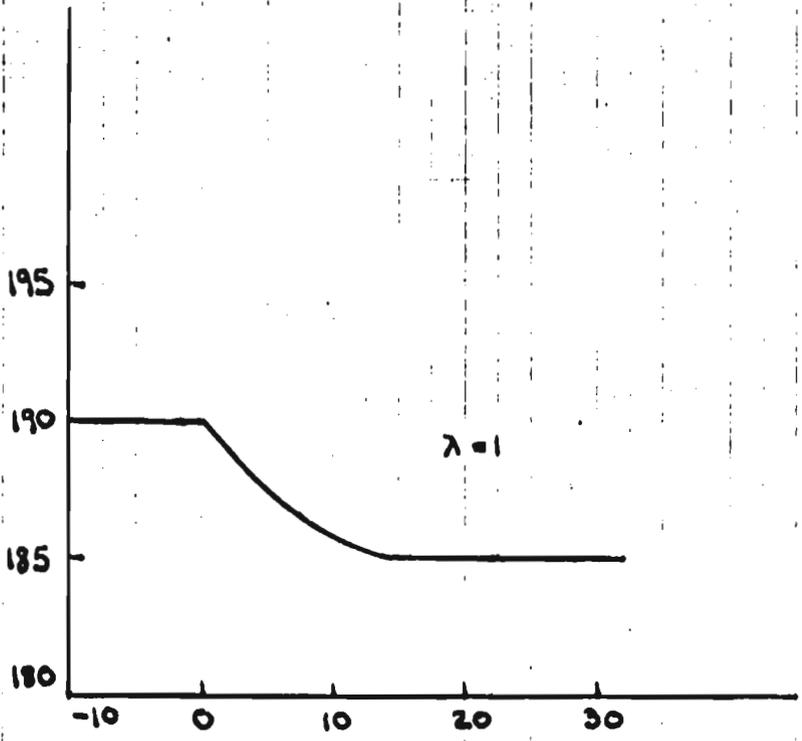
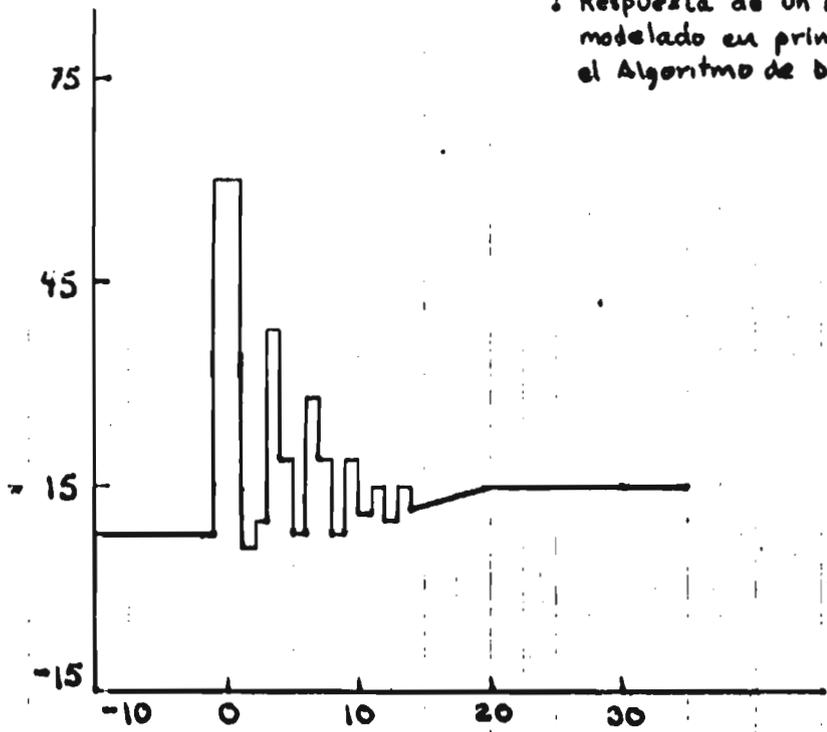
$$36 \quad H_G(z) = \frac{k z^{-N} (1 - \alpha)}{z - \alpha}$$

obteniéndose al sustituir en la ecuación 34

$$37 \quad D(z) = \frac{(1 - \beta) (1 - \beta z^{-1})}{k [1 - \beta z^{-1} - (1 - \beta) z^{-(N+1)}] (1 - \alpha)}$$

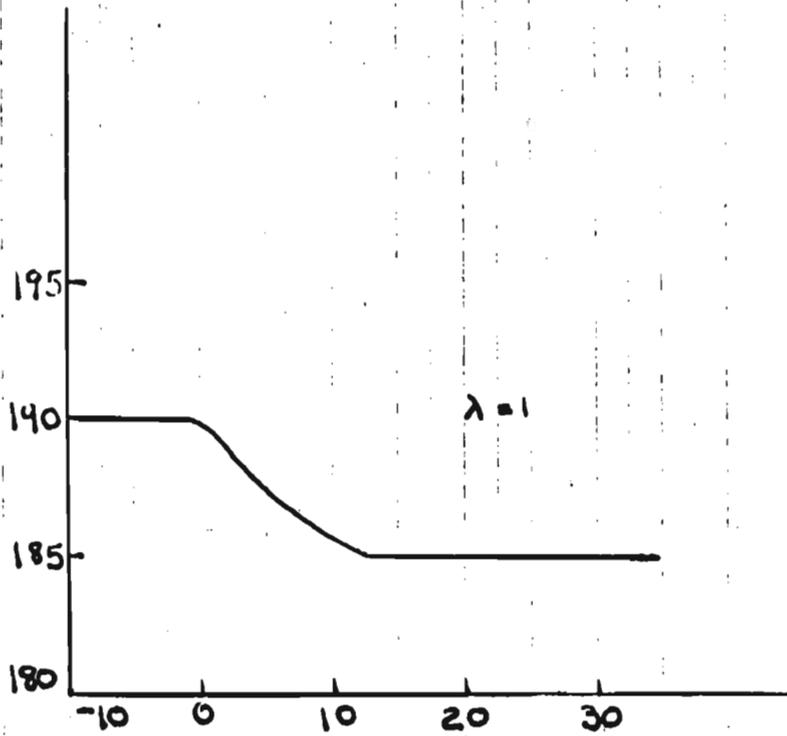
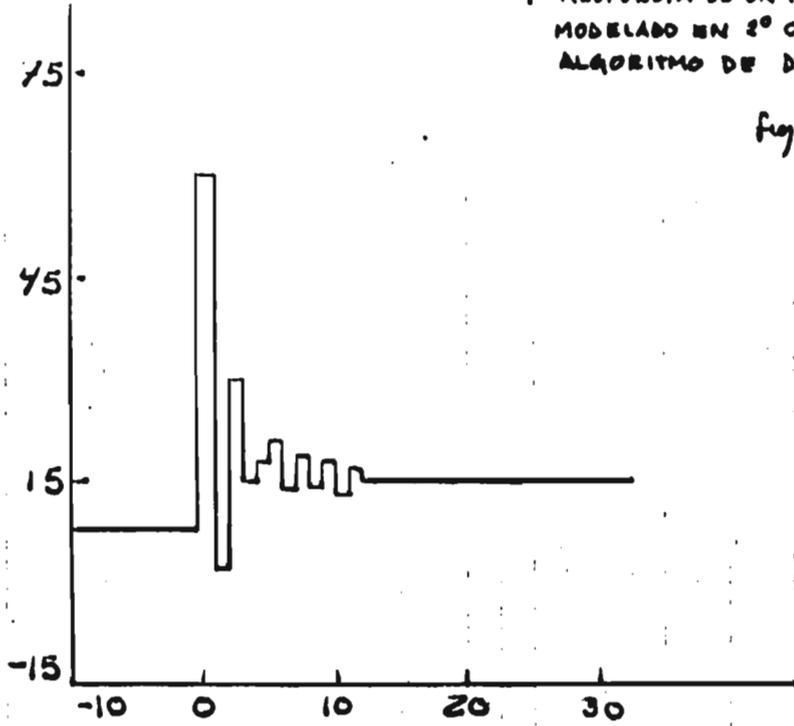
Fig 14

• Respuesta de un Reactor Exotérmico modelado en primer orden, usando el Algoritmo de Dahlin. $T = 1$



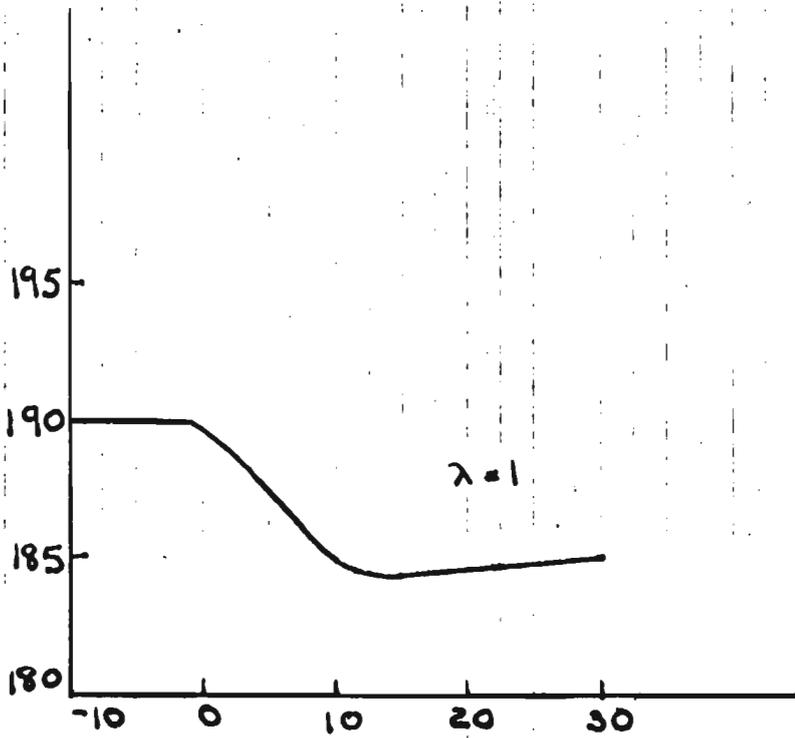
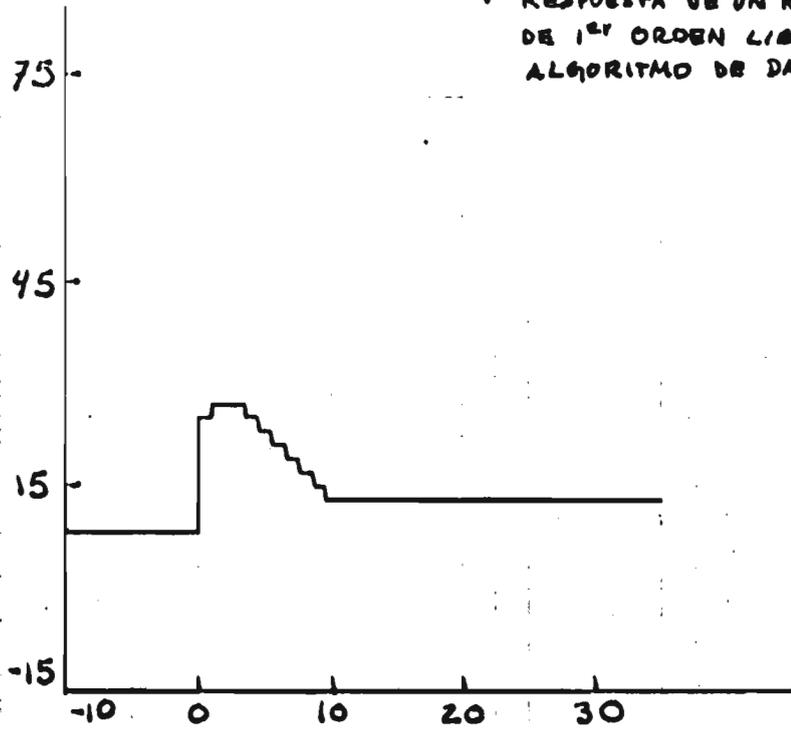
RESPUESTA DE UN REACTOR EXOTERMICO
MODELADO EN 2º ORDEN USANDO EL
ALGORITMO DE DAHLIN.

Fig 15



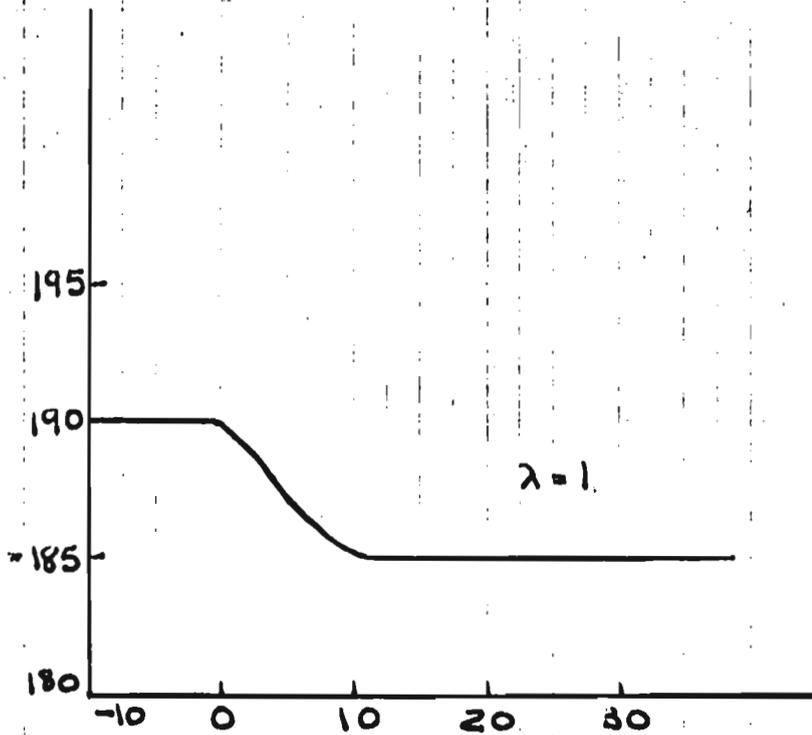
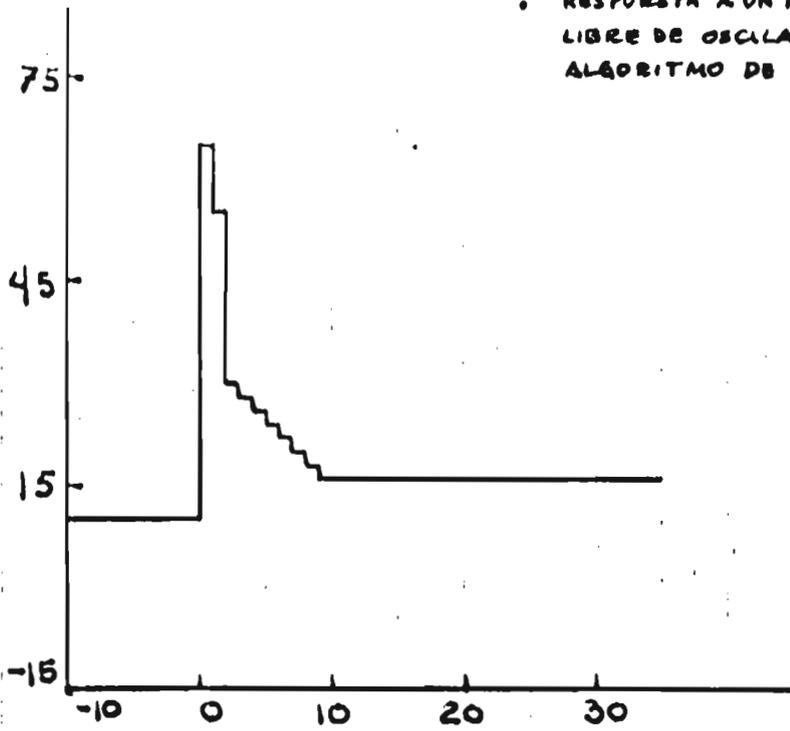
• RESPUESTA DE UN REACTOR PARA MODELO DE 1^{er} ORDEN LIBRE DE OSCILACIONES ALGORITMO DE DAHLIN.

fig 16



• RESPUESTA A UN MODELO DE 2º ORDEN
LIBRE DE OSCILACIONES,
ALGORITMO DE DAWLIN.

Fig 17



para un proceso representado por una ecuación de segundo orden con retraso tenemos :

$$38 \quad G(s) = \frac{K e^{-Ls}}{(s_1s+1)(s_2s+1)}$$

$$39 \quad H(z) = \frac{K (C_3 + C_4 z^{-1}) z^{-(N+1)}}{(1-\alpha_1 z^{-1})(1-\alpha_2 z^{-1})}$$

y el compensador queda.

$$40 \quad D(z) = \frac{(1-\beta)(1-\alpha_1 z^{-1})(1-\alpha_2 z^{-1})}{K [1-\beta z^{-1} - (1-\beta)z^{-(N+1)}] (C_3 + C_4 z^{-1})}$$

donde

$$C_3 = \alpha_1 \alpha_2 + \frac{\beta_1 \alpha_1 - \beta_2 \alpha_2}{\beta_2 - \beta_1}$$

$$C_4 = (1-\alpha_1)(1-\alpha_2)$$

Si los algoritmos 37 y 40 tienen polos cercanos a $z = -1$ habrá oscilaciones en la variable manipulada. Si el tiempo de muestreo es pequeño respecto a la constante de tiempo del sistema pueden removerse seguramente dichos polos reduciendo a los algoritmos PI y PID respectivamente, siendo por las ecuaciones 2, 37 figs: 14 a 17

$$41 \quad K_c = \frac{1-\beta}{K (\alpha^{-1}-1) [1+N(1-\beta)]}$$

$$42 \quad T_i = \frac{T}{\alpha-1}$$

y para 3, 40

$$43 \quad K_c = \frac{(\alpha_1^{-1} + \alpha_2^{-1} - 2)(1-\beta)}{K (\alpha_1^{-1}-1)(\alpha_2^{-1}-1) [1+N(1-\beta)]}$$

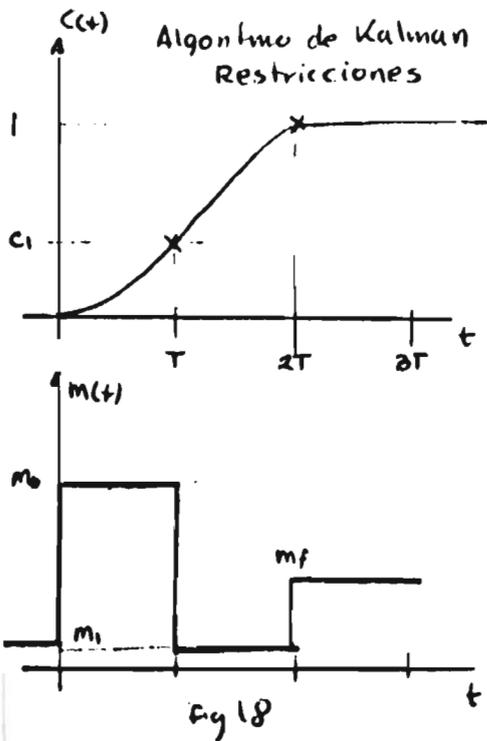
$$44 \quad T_i = \frac{T (\alpha_1^{-1} + \alpha_2^{-1} - 2)}{(\alpha_1^{-1}-1)(\alpha_2^{-1}-1)}$$

$$T_D = \frac{T}{\alpha_1^{-1} + \alpha_2^{-1} - 2}$$

5.- Método Kalman: F

El algoritmo de Kalman es un método de tiempo de estabilidad mínimo. En lugar de especificar C/R este método restringe el comportamiento de las variables manipuladas y las variables controladas.

v.g. Se desea que la respuesta C se estabilice en dos periodos de muestreo asumiendo que la variable M puede tomar solo dos valores y estabilizarse fig. 18



$$45 \quad C(z) = \sum_{n=0}^{\infty} c_n z^{-n}$$

$$46 \quad M(z) = \sum_{n=0}^{\infty} m_n z^{-n}$$

$$m_2 = m_3 = m_f = \dots$$

las funciones de transferencia para estas funciones se pueden encontrar: Para $R_{(z)}$ = escalón

$$47 \quad \frac{C(z)}{R(z)} = (1-z^{-1}) [c_1 z^{-1} + \dots] \\ = c_1 z^{-1} + (1-c_1) z^{-2} \\ = p_1 z^{-1} + p_2 z^{-2} = P(z)$$

del mismo modo :

$$48 \quad \frac{M(z)}{R(z)} = q_0 + q_1 z^{-1} + q_2 z^{-2} = Q(z)$$

podemos escribir el proceso como :

$$49 \quad HG_{(z)} = \frac{P(z)}{Q(z)}$$

de modo que P y Q quedan identificadas y el controlador se expresa directamente en función de los coeficientes del proceso.

$$50 \quad D_{(z)} = \frac{Q(z)}{1-P(z)}$$

Se demuestra que la planta $HG_{(z)}$ debe ser del orden del polinomio Q, y el número de cambios permitidos en M será igual al del proceso en particular.

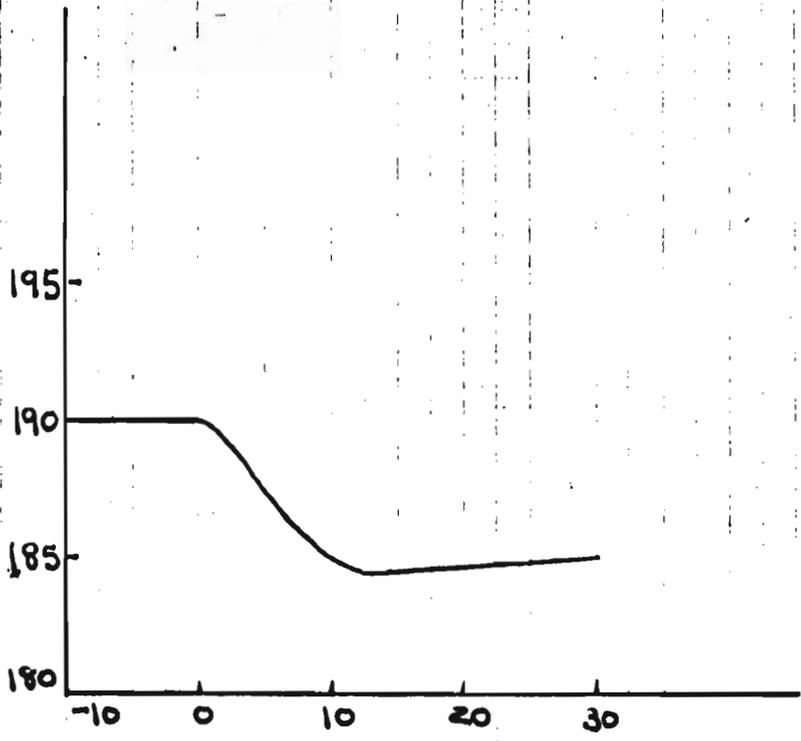
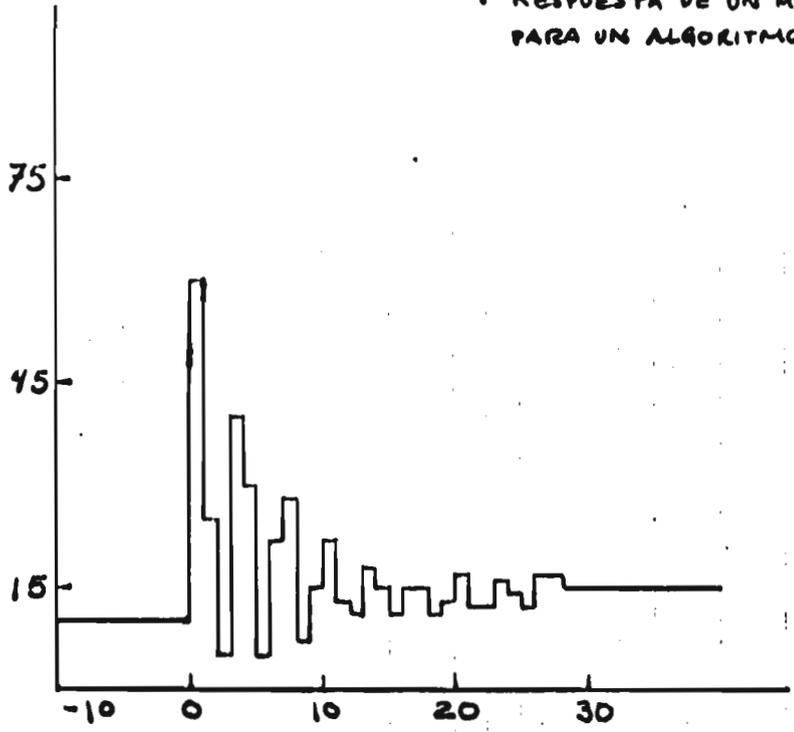
Debe notarse que aunque el proceso tenga un retraso el cálculo de D(z) es exactamente igual.

Para una planta de 1er orden y retraso se obtiene

$$51 \quad HG(z) = \frac{k (c_1 + c_2 z^{-1}) z^{-(N+1)}}{1 - \alpha z^{-1}} = \frac{P(z)}{Q(z)}$$

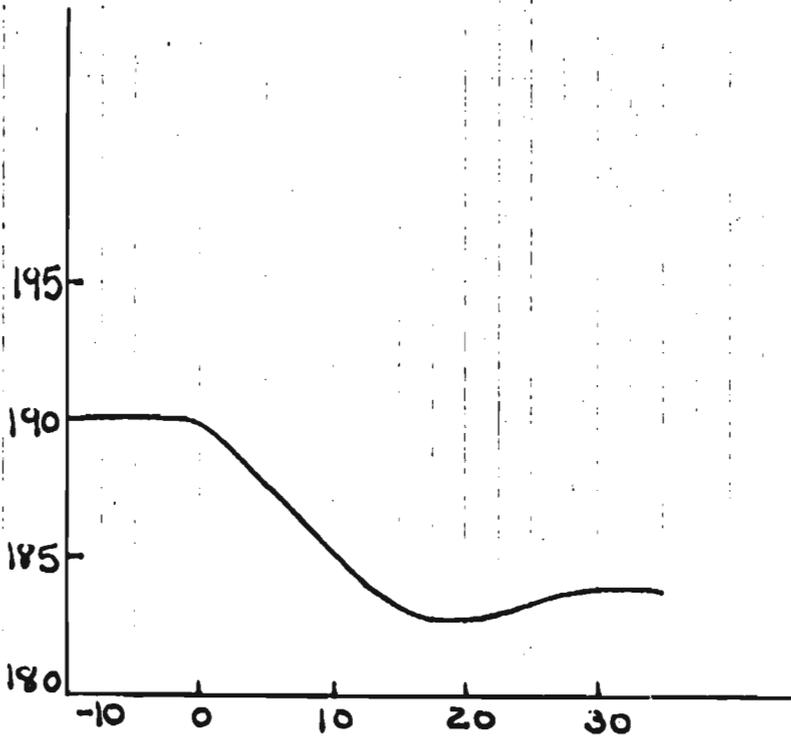
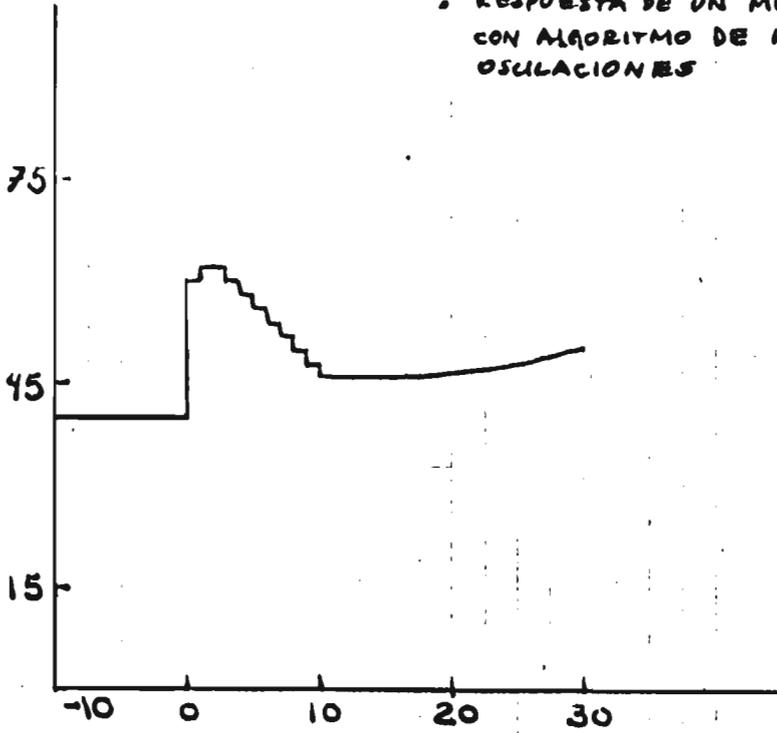
Fig 19

• RESPUESTA DE UN MODELO DE 2º ORDEN PARA UN ALGORITMO DE KALMAN



fy 20

• RESPUESTA DE UN MODELO DE 2° ORDEN
CON ALGORITMO DE KALMAN LIBRE DE
OSCILACIONES



trabajando con esta expresión se obtiene

$$52 \quad P(z) = \frac{C_1 z^{-(N+1)}}{C_1 + C_2} + \frac{C_2 z^{-N}}{C_1 + C_2}$$

$$53 \quad Q(z) = \frac{1}{K(C_1 + C_2)} - \frac{\alpha}{K(C_1 + C_2)} z^{-1}$$

obteniendo:

$$54 \quad D(z) = \frac{1 - \alpha z^{-1}}{K(C_1 + C_2) \left[1 - \frac{C_1 z^{-(N+1)}}{C_1 + C_2} - \frac{C_2 z^{-N}}{C_1 + C_2} \right]}$$

La respuesta del sistema y la variable manipulada se muestran. fig. 19, 20

Las oscilaciones en M pueden eliminarse quedando un algoritmo implementado con PI :

$$55 \quad D(z) = \frac{1}{K(\alpha_1 - 1)(N + z - X)} \left(1 + \frac{\alpha_1 - 1}{1 - z^{-1}} \right)$$

$$X = \frac{C_1}{C_1 + C_2}$$

igualmente para el modelo de segundo orden con retraso nos lleva a un algoritmo PID

$$56 \quad D(z) = \frac{(1 - \alpha_1 z^{-1})(1 - \alpha_2 z^{-1})}{K(C_3 + C_4)(z + y)(1 - z^{-1})}$$

$$y = \frac{C_4}{C_3 + C_4}$$

$$K_c = \frac{\alpha_1 + \alpha_2 - 2}{K(\alpha_1 - 1)(\alpha_2 - 1)(2 + N - X)}$$

$$T_i = T \frac{\alpha_1 + \alpha_2 - 2}{(\alpha_1 - 1)(\alpha_2 - 1)}$$

$$T_d = \frac{T}{\alpha_1 + \alpha_2 - 2}$$

TABLA I

Nomenclatura.

a = (L-NT)/T parámetro de transformada z modificada

C(z) = variable controlada

C₁, C₂, C₃, C₄ constantes

D (z)	Función de transferencia del controlador
E (z)	Señal de error
G (z)	Función de transferencia del proceso continuo
H (z)	Función de transferencia del retensor
HG (z)	FT de pulso del par retensor-proceso
K	Ganancia de estado estable del proceso
Kc, Ki, Kd	Parametros PID del algoritmo de control
L	Tiempo muerto del proceso
M (z)	Variable manipulada
R (z)	Variable punto de trabajo
T	Periodo de muestreo
Ti	Tiempo de integración
Td	Tiempo de derivación
α, β	Constantes $\alpha_j = e^{-V/s_j}$ $\beta = e^{-V/\lambda}$
λ	Constante de tiempo del algoritmo de Dahlin
Θ	Tiempo de la función de malla cerrada
τ_1, τ_2, τ_3	Constantes de tiempo de proceso.

6.- Algoritmo FTS con retroalimentación de variables de estado. Ref 1,4

El uso del algebra lineal y su facilidad de manejo en procesadores digitales ofrece amplias ventajas al uso de variable de estado en la representación de sistemas.

Sea el proceso:

$$x_{k+1} = P x_k + q u_k$$

Se trata de llevar al sistema de un estado X_0 a otro X_f por medio de la manipulación discreta U_k .

Retrolimentación de Variables de Estado.

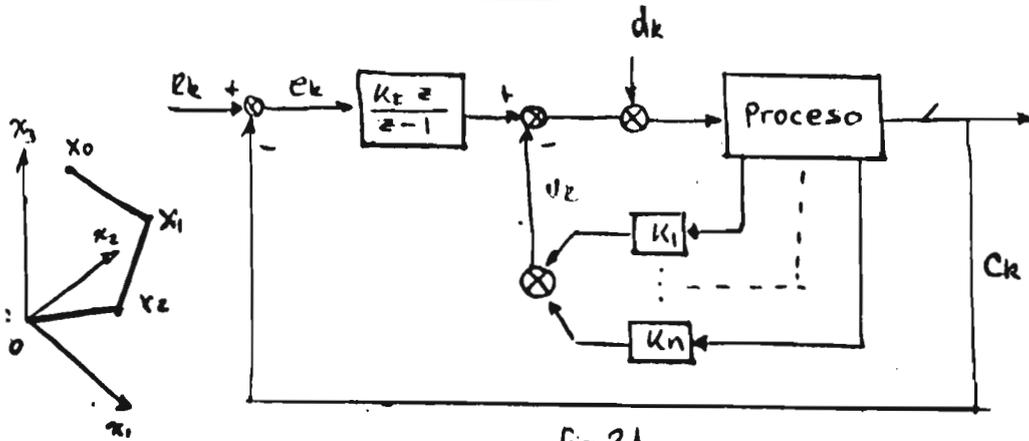


Fig 21

El controlador $D(z)$ para este propósito es

$$u_k = -D(z) x_k$$

para obtener $D(z)$ se calcula el valor de los estados
Obteniendose:

$$x_k = 0 = \left\{ P^{-n} \begin{bmatrix} p^{n-1} q, & \dots, & p q, & q \end{bmatrix} \begin{bmatrix} D \\ D(P-qD) \\ \vdots \\ D(P-qD)^{n-1} \end{bmatrix} \right\}$$

El primer renglón de la matriz final es el controlador $D(z)$.

El Algoritmo en variable de estado para un proceso de 1 er. orden con retraso es:

$$G(s) = \frac{e^{-LT}}{s+1}$$

$$HG(z) = \frac{1}{z^n} \frac{1-\alpha}{z-\alpha}$$

Sea $L=T$ entonces $N=1$ y
 expresado en variable de estado



$$62 \quad \begin{bmatrix} x_{1k+1} \\ x_{2k+1} \end{bmatrix} = \begin{pmatrix} 0 & 0 \\ 1 & \alpha \end{pmatrix} \begin{bmatrix} x_{1k} \\ x_{2k} \end{bmatrix} + \begin{bmatrix} 1-\alpha \\ 0 \end{bmatrix} U_k$$

se obtiene

$$63 \quad \boxed{D(z) = \left[\frac{\alpha}{1-\alpha}, \frac{\alpha^2}{1-\alpha} \right]}$$

Este algoritmo tiene error de offset y requiere que se realimenten las variables de estado que no siempre son obtenibles. No obstante, el primer problema puede resolverse usando un algoritmo de velocidad.

Con un elemento actuador, integrador tenemos:

$$64 \quad HV_f(z) = \frac{z}{z-1} \frac{1-\alpha}{z(z-\alpha)}$$

Obteniendose

$$65 \quad \boxed{D(z) = \left[\frac{1+\alpha}{1-\alpha}, \frac{\alpha^2}{1-\alpha} \right]}$$

La realimentación de variable de estado se puede evitar haciendo $n-1$ mediciones de la variable de salida para un sistema de orden n quedando el controlador expresado por:

$$66 \quad -D(z) = \frac{U(z)}{Y(z)} \quad \begin{array}{l} U(z) \text{ Acción} \\ Y(z) \text{ Medición} \end{array}$$

Para la medición escalar "y" tenemos:

$$67 \quad y_k = C X_k, \quad C = [c_1, c_2, \dots, c_n]$$

v.g. Para un sistema de 2º orden

$$68 \quad \begin{aligned} y_0 &= C X_0 \\ y_1 &= C X_1 = c P x_0 + c q u_0 \end{aligned}$$

$$69 \quad \begin{bmatrix} y_0 \\ y_1 \end{bmatrix} = \begin{bmatrix} c \\ c P \end{bmatrix} X_0 + \begin{bmatrix} 0 \\ c q \end{bmatrix} u_0$$

de donde se puede identificar al estado inicial X_0 . El algoritmo de control queda

$$70 \quad u_1 = -D(z) P \underbrace{\begin{bmatrix} c \\ c P \end{bmatrix}^{-1} \left\{ \begin{bmatrix} y_0 \\ y_1 \end{bmatrix} - \begin{bmatrix} 0 \\ c q \end{bmatrix} u_0 \right\}}_{X_0} - D_{(z)}^q u_0$$

v.g. Sea un sistema de primer orden con retraso igual al periodo de muestreo y la medición X_2

$$71 \quad \begin{aligned} y &= X_2 \\ C &= [1, 0] \end{aligned}$$

$$\bar{y} = C \bar{x}$$

el algoritmo D se determino en la ecuación ⁶⁵ y de P en la ecuación ⁶⁴ tenemos

$$72 \quad P = \begin{bmatrix} 1 & 0 \\ 1 & \alpha \end{bmatrix}$$

$$73 \quad u_1 = - \begin{bmatrix} \frac{1+\alpha}{1-\alpha} & \frac{\alpha^2}{1-\alpha} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & \alpha \end{bmatrix} \underbrace{\begin{bmatrix} y_1 - \alpha y_0 \\ y_0 \end{bmatrix}}_{X_0} - (1+\alpha) u_0$$

en transformada Z tenemos

$$74 \quad - D(z) = \frac{U(z)}{Y(z)} = \frac{1}{1-\alpha} \frac{\alpha(1+\alpha) - (\alpha^2 + \alpha + 1)z}{z + (1+\alpha)}$$

la función del sistema es:

$$75 \quad \frac{C(z)}{R(z)} = \frac{[\alpha(1+\alpha)H]z - \alpha(1+\alpha)}{z^3}$$

misma que nos presenta offset.

7.- Algoritmo Observador

Este algoritmo trabaja sobre un modelo de la planta, (implementado en software) el cual identifica el comportamiento del proceso y presenta variables de estado para ser usadas en la retroalimentación de variables de estado.

Si el proceso es variable en el tiempo el observador puede realizar algoritmos simples para adaptarse a aquél. Ref 4.

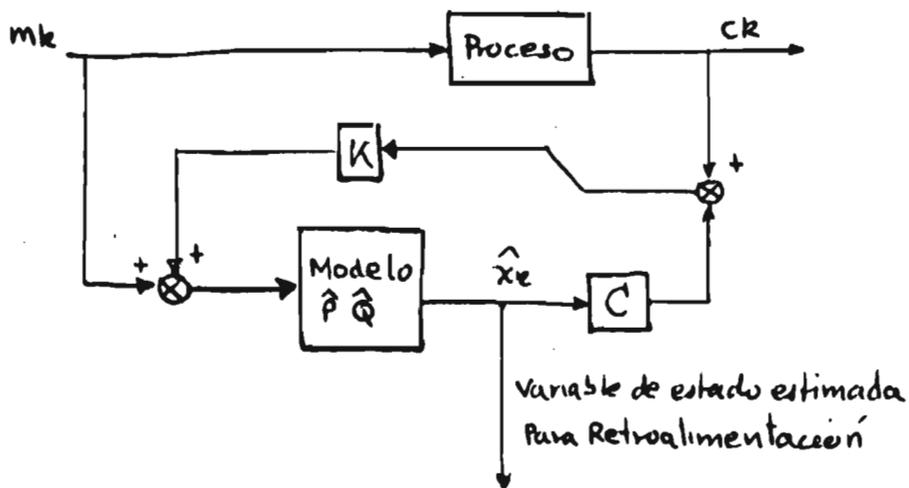


fig 22

8. Algoritmo Predictor. G

Este algoritmo se obtiene de tratar al retensor de orden cero como un retraso puro de magnitud T/θ . En principio se trata de predecir por medio de un modelo cuál será la salida en el próximo intervalo de muestreo.

$$76 \quad C(s) = (m(s) + u(s)) \frac{K e^{-\theta s}}{\theta s + 1}$$

La ley de control será:

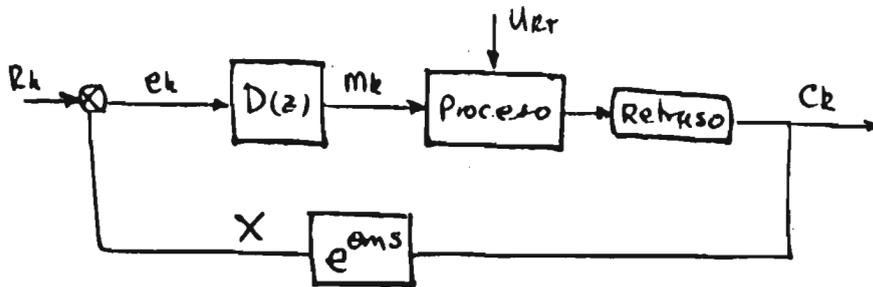
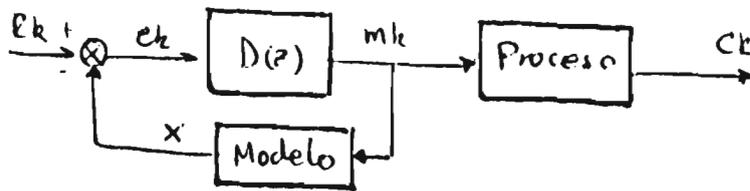
$$77 \quad m_{kr} = K_c \left(R_{kr} - C_{kr+\theta+T/2} \right) - u_{kr}$$

$C_{kr+\theta+T/2}$ · salida predicha en:

$$t = \theta + \frac{T}{2}$$

u_{kr} Terminó para tomar en cuenta los
disturbios.

Este algoritmo elimina los retrasos de transporte asociados con el par retensor-proceso



predicador
fy 23

El predictor utiliza el valor actual de la salida C_0 y los valores de M desde Θ_m unidades de tiempo para estimar la respuesta que habra Θ_m unidades en el futuro.

Si la planta es de 1 er orden $\frac{K}{s+1}$ el predictor resuelve:

78

$$\tau \frac{dc}{dt} + C = m \quad C(t_0) = C_0$$

cuya solución es

$$79 \quad C(t_0 + \Theta_m) = C_0 \Theta^{-\tau/\tau} + K \int_{t_0}^{t_0 + \Theta_m} U(\lambda - \Theta_m) e^{-\tau(\lambda - \Theta_m)/\tau} d\lambda$$

la salida del predictor es

$$80 \quad X_{kT + \Theta_m} = K(1-A')(m_{kT} + U_{kT}) + A'K(1-C')(m_{kT-2} + U_{kT}) + A'C' C_{kT}$$

81

$$m_{kT} = C_1 R_{kT} - C_2 C_{kT} - C_3 (m_{kT-1} + U_{kT}) - U_{kT}$$

TABLA 2

Parametros usados en los algoritmos.

$$A = 1 - e^{-(r-l)/\tau}$$

$$A' = e^{-T/2\tau}$$

$$B = \left[e^{-T/\tau} - e^{-(T-L)/\tau} \right]$$

$$C' = e^{-L/\tau}$$

$$C1 = \frac{K_c K + 1}{K [1 + K_c K (1 - A')]}$$

$$C2 = \frac{K_c A' C'}{1 + K_c K (1 - A')}$$

$$C3 = \frac{K_c K A' (1 - C')}{1 + K_c K (1 - A)}$$

Los valores K_c y K_i del predictor pueden calcularse para la respuesta C/R deseada para un proceso de primer orden

Si se quiere un FTS se obtienen :

$$91^{\circ} \quad K_c = \frac{A}{K(1-A)}$$

$$K_i = \frac{0,3}{\tau K (1-\beta)}$$

0.3 es valor empirico

La respuesta de este algoritmo para cambios en el set point es semejante teniendo el predictor mayores pendientes y mas sobretiro

En su respuesta a disturbios el predictor da mejores resultados que el control PI óptimo pero esta aun debajo de la calidad del PID

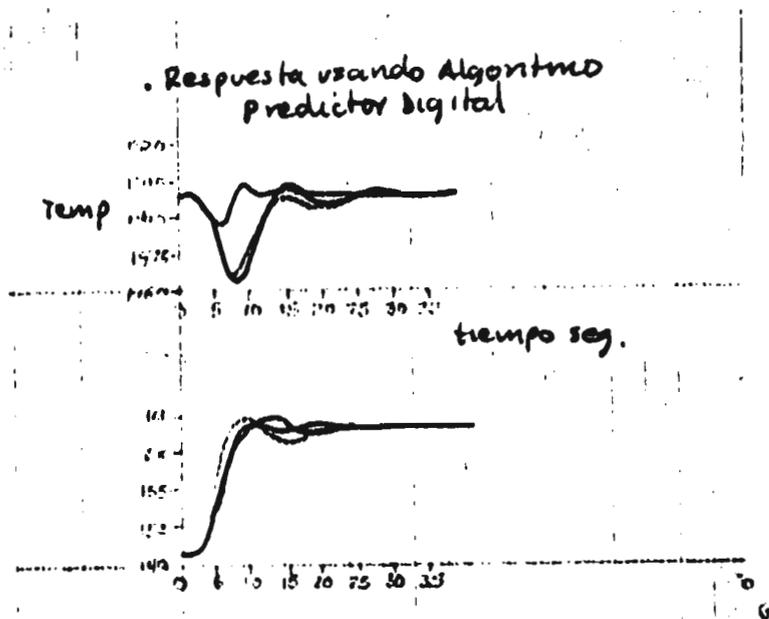
Sin embargo el predictor es mejor que los algoritmos PI, PID sintonizados en forma estandar y ofrece un mejor control en procesos donde el retraso es significativo.

El término de disturbio se calcula como

$$82 \quad u_{kr} = u_{kr-1} + K_i (C_{kr-1} - X_{kr}) T$$

y la salida del modelo en el momento del muestreo es :

$$83 \quad X_{kr} = K(1 - \beta'/c')(m_{kr-1} + u_{kr-1}) + \beta'/c'(1 - c')(m_{kr-2} + u_{kr-2})$$



- Algoritmo PI óptimo
- " " PID óptimo
- Algoritmo predictor

* Rise time

Control Co-alimentado.

El control coalimentado se aplica en problemas de regulación realizando mediciones de los disturbios y actuando en contraposición - por medio de la variable manipulada, adelantándose a la acción de aquellos.

La acción del control coalimentado puede ser sumada a la salida del controlador convencional.

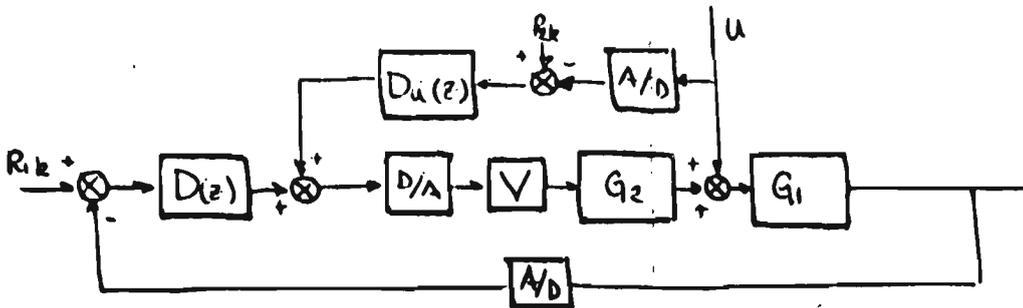


Diagrama del Control Coalimentado por suma.
Fig 24

En otra configuración se puede realizar una coalimentación multiplicativa (ver ref.10).

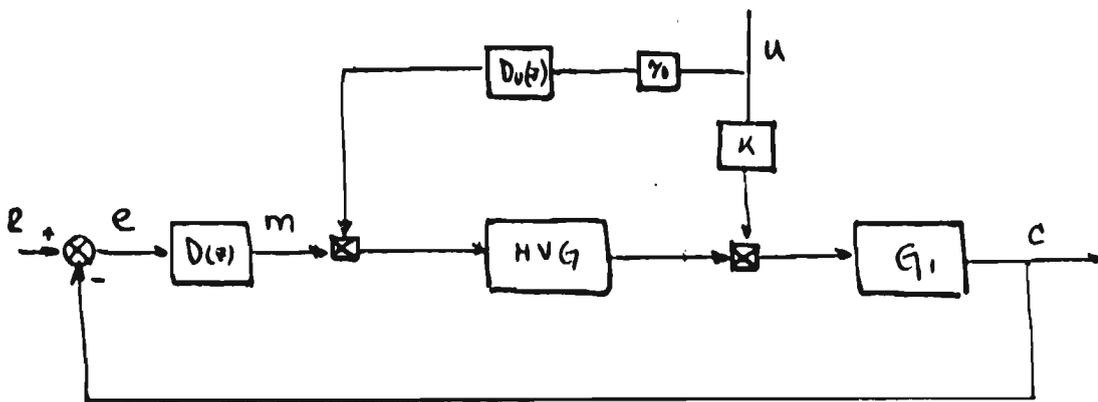


Fig 25
Diagrama del Control coalimentado multiplicativo

9. Algoritmos para configuraciones cascada.

26

La configuración cascada figura, contiene dos lazos uno interno o esclavo y uno externo o máestro.

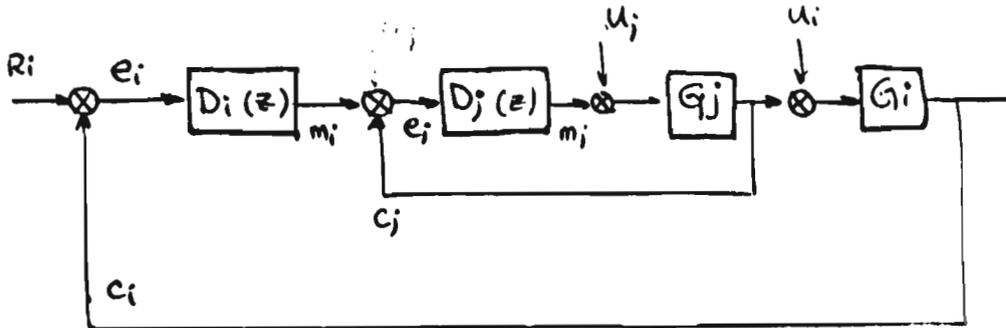


Fig 26
Diagrama de control en cascada

$$84 \quad R_{j,k} = R_{j,k-1} + m_j e$$

Se procede entonces a efectuar el diseño del controlador esclavo y despues a partir de la relación entre Ci y el punto de operación Rj se diseña Di controlador maestro, siguiendo las tecnicas antes citadas.

Como un ejemplo de aplicación de DDC con configuraciones tanto de coalimentación como de cascada tenemos el control de temperatura de un calentador en línea con doble alimentación. Ref 10 fig 28

Diagrama de bloques : pag. siguiente.

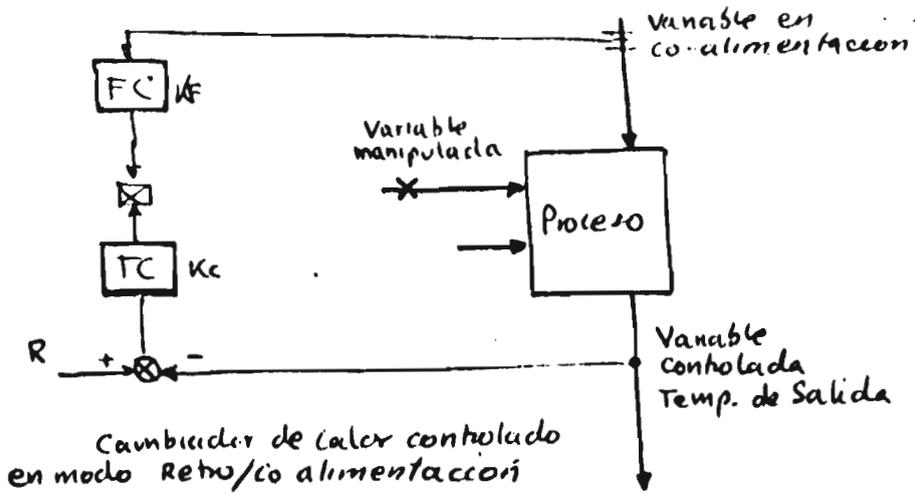


fig 27

La transferencia de controlador de temperatura es (algoritmo de velocidad PID):

$$85 \quad \Delta M_{FB} = \frac{100}{PB} \left[(e_k - e_{k-1}) + K_i e_k + K_d (e_k - 2e_{k-1} + e_{k-2}) \right]$$

bajo una suma finita continua representada por el simbolo \sum este algoritmo se transforma en posicional Mfb. El controlador de coalimentación esta dado por

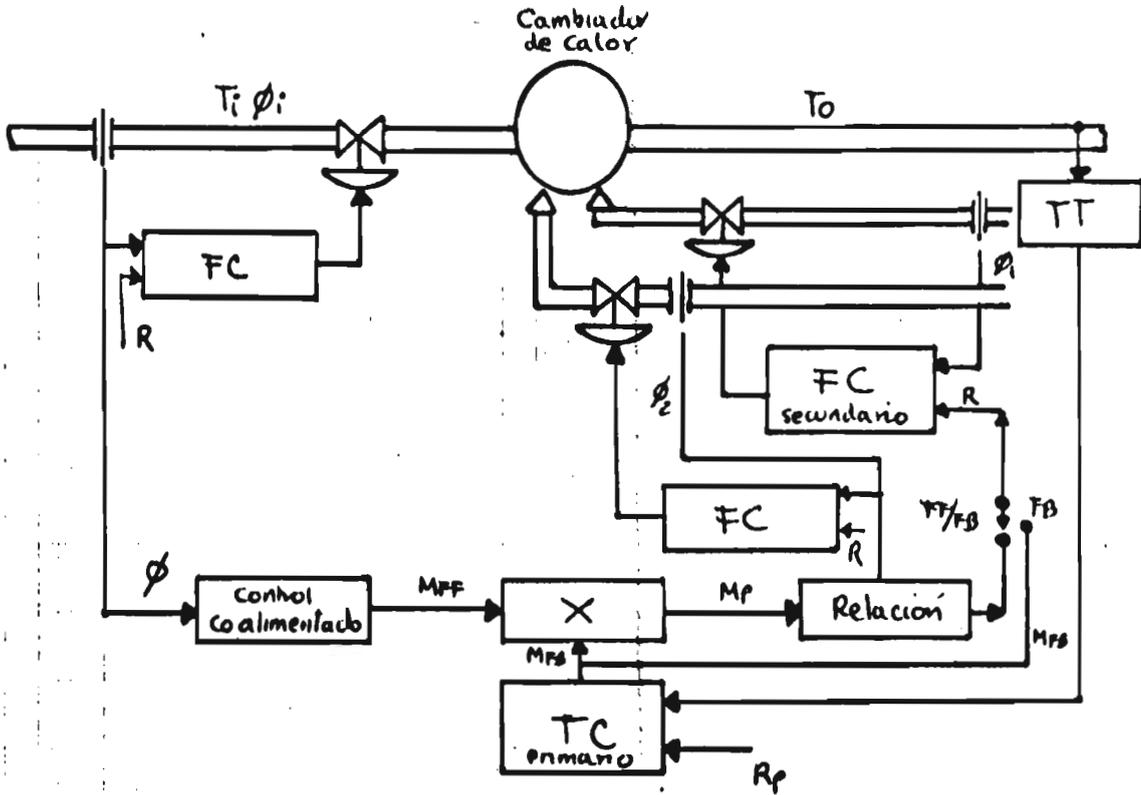
$$86 \quad M_{FF} = \frac{T_1}{T_2} (u_k - z_k) + z_k$$

donde z_k es una variable de retraso muda que se usa como estimación o predicción.

$$87 \quad z_{k+1} = \frac{T}{T_2} (u_k - z_k) + z_k$$

Diagrama completo del Intercambiador de calor y el sistema de control

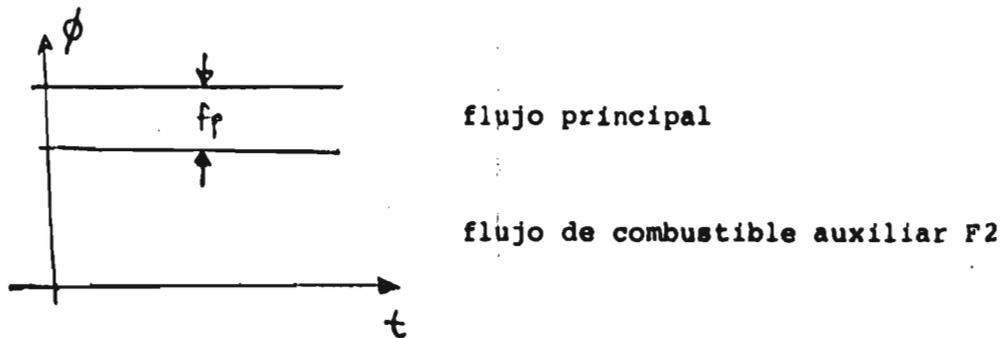
Fig 28



para T1 Constante de adelanto en MFF

T2 Constante de atraso en MFF.

Los términos MFB y MFF son coalimentados según la configuración multiplicativa procediéndose a escalar el resultado usando K_1 y obteniéndose el valor parcial de la variable manipulada M_p la cuál se polariza dependiendo del valor del flujo auxiliar de combustible. El valor de M_p es el set point del controlador esclavo que en este caso es el de flujo principal.



Relación de flujos en el tiempo.

Fig 29

El diagrama de implementación digital se muestra a continuación Fig 30 conjuntamente con un diagrama de flujo de la rutina del controlador. Fig 31

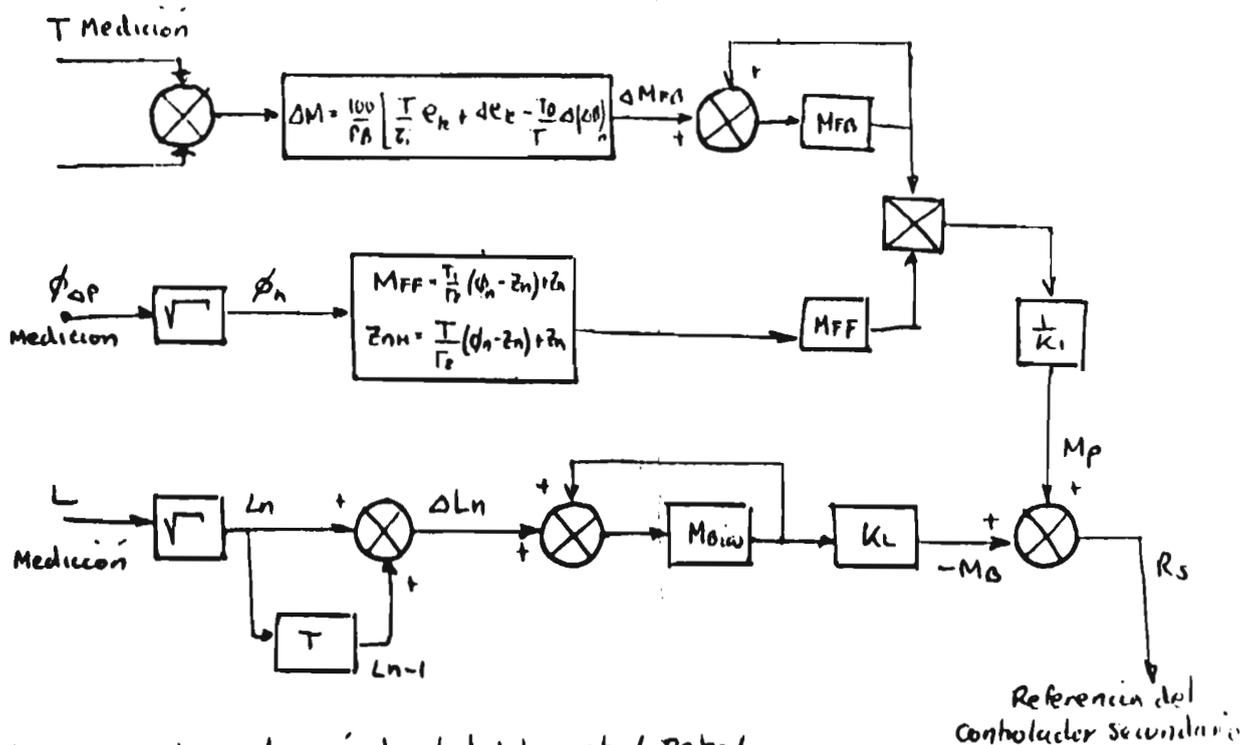


Fig 30. Implementación digital del control Retro/co-Alim.

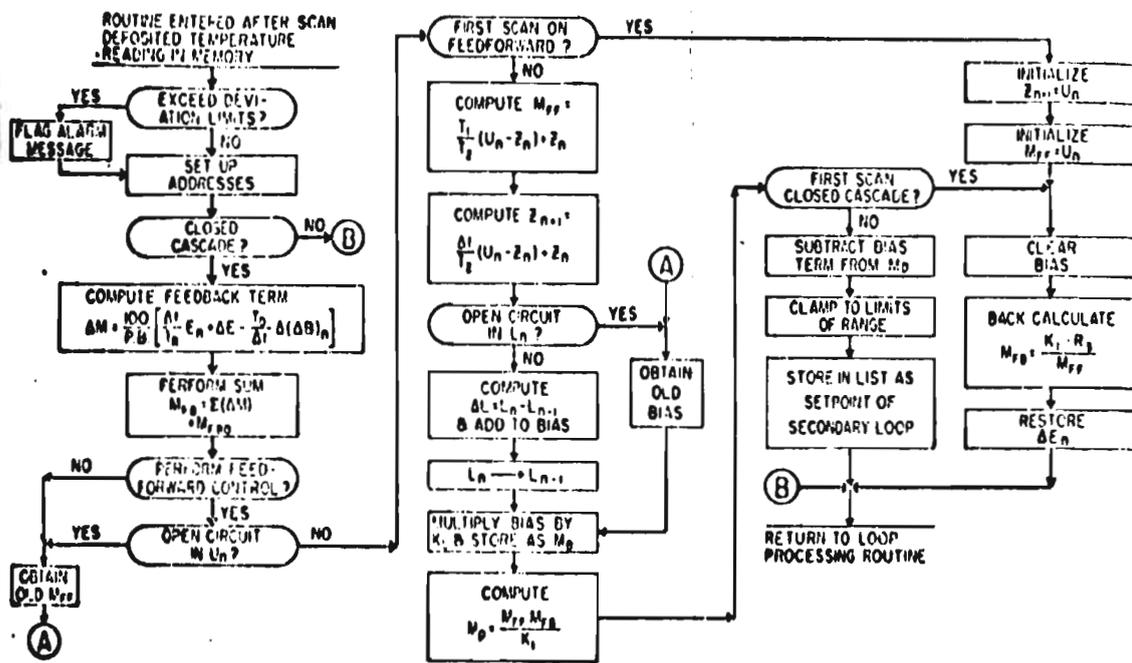


Fig 31 Diagrama de flujo simplificada del control Fig 28

Implementación del controlador con Microprocesadores.

La aplicación de este nuevo dispositivo a los "sistemas de control" es inmediata debido a que los algoritmos discretos que debe resolver el controlador son recursivos y requieren básicamente de operaciones aritméticas sencillas (+, -, x, ÷) y de manejo simple de datos.

El microprocesador, para realizar sus funciones, cuenta con una unidad de control (CPU), unidad de memoria, una unidad lógica y aritmética y unidades de entrada y salida. De modo que tenemos el -- dispositivo idóneo para usarlo en acciones recursivas. Sin embargo, existe aún una diferencia fundamental entre el tipo de variables en el campo y el rango de trabajo del mismo, primeramente las variables de medición generalmente analógicas deben ser normalizadas y acotado su rango (4-20ma condicionador). A continuación se deben de codificar digitalmente de modo que el "micro" las pueda reconocer y entender, o sea la información queda codificada por palabras de una longitud constante (v.g. 8bits).

La manera canónica de identificar la información en cada palabra es relacionar el rango analógico (0-20ma) linealmente con el campo digital (00-FF) teniéndose una cuantificación binaria entera (ver error de cuantificación) Esta primera forma ofrece ventajas desde el punto de vista del procesador. Sin embargo, debido al carácter de las variables físicas, es generalmente imprescindible el uso de tipos de aritmética no entera de punto flotante lo cual requiere identificar a la información como una partición en mantissa

(6 bits) y exponente (2 bits).

Este formato de numeración nos resuelve el problema del manejo de los valores físicos, sin embargo, la precisión del controlador disminuye considerablemente en función del número de dígitos que se asignen a la mantisa. Los problemas tratados se pueden resolver mediante el uso de algoritmos de doble o mayor precisión que realizan la concatenación de varios bytes creando palabras de mayor longitud. En las operaciones del sistema se deben manejar también los signos de las variables, lo cual requiere el uso de numeración apropiada.

Los requerimientos de las operaciones \times , \div deben de cubrirse ya sea por subrutinas programadas o por elementos electrónicos directos creados para este fin.

A pesar de la automatización de estos sistemas eventualmente es necesario indicar a un operador el estado de cierta variable o recibir alguna orden, acto que debe realizarse en el lenguaje del operador. La traducción puede hacerse programada en un sistema independiente reconocido como terminal inteligente. Inclusive es posible el uso de lenguajes interactivos y prácticamente las compañías del ramo han diseñado varios lenguajes especiales para estas aplicaciones. La intercomunicación de la terminal o puesto central con los subsistemas micro-controlados se simplifica mediante el uso de técnicas de "acceso directo" e "interrupciones".

La comunicación del "micro" con el proceso se realiza por sus puertas de entrada y salida y es controlada de manera total por el --

mismo micro quien por medio del programa adquisitor trae los datos de los varios puntos de medición y los almacena en los bloques correspondientes efectuando un procesamiento intermedio (medias, desviaciones, diferencias, alarmas etc) en el momento preciso el procesador da servicio a los datos ejecutando el programa controlador y obtiene la acción de control la cuál es mandada cada período de muestreo al transmisor y controlador, terminando el procesamiento vuelve el CPU a atender el programa adquisitor y el procedimiento se repite. En los sistemas con uso de minis pueden tenerse programas adquirentes muy sofisticados llamados ejecutivos que disminuyen al máximo la intervención del operador.

Los programas ejecutivos, adquirentes y subrutinas de propósito general han sido diseñados por los fabricantes, inclusive es posible también adquirir soporte para programas de control de algoritmos muy generales PI, PID. Sin embargo, la cualidad superior del uso de control DDC es la capacidad de diseño y modificación de los algoritmos para ajustarlos al comportamiento de la planta y así obtener óptimos resultados.

Entonces debemos realizar nuestra programación y ajuste, cosa que se antoja complicada, sin embargo, gracias a las facilidades de lenguajes de alto nivel ó de ensambladores el problema se atenúa. Cuando se prescinde de los anteriores, la programación en lenguaje de máquina y las rutinas de prueba del programa se pueden resolver por medio de un ensamblador cruzado y de un simulador.

Se presenta el caso particular de un controlador para ser usado en un proceso de primer orden sometido a cambios de la carga y en el punto de referencia.

El controlador está representado por

$$D(z) = \frac{b_1 z^{-1}}{1 + a_1 z^{-1}}$$

Debido a la magnitud de los parámetros característicos: a_1 , b_1 , K_I , K_P , K se requiere manejar números en el rango de 10^{-2} hasta 10^2 de modo que se hace uso de doble precisión. (16 bits)

El micro usado fue un KIT AMI 6800 cuyas características pueden encontrarse en las referencias. La programación se hizo en lenguaje de máquina entrando vía teletipo. El programa del controlador entrega el comando u_k a un programa simulador el cual calcula en base al modelo de la planta cuál sería la salida de éste (y_k). Ambos valores se muestran como salida u_k , y_k . Los parámetros del sistema: P , Q , así como la entrada o punto de referencia y la simulación de cambios en la carga pueden indicarse desde la terminal. El programa controlador tuvo un total de 51 instrucciones y un tiempo de ejecución para cada iteración de 120 ciclos de máquina (0.96 ms.) el programa simulador tuvo 54 instrucciones y un tiempo de ejecución de 1.22 ms. Finalmente el programa monitor quién contiene las subrutinas de carga de datos, muestra de títulos y resultados tuvo 128 instrucciones incluyendo 13 macros.

Un factor sumamente importante en el diseño de los algoritmos de

DDC es el tiempo de muestreo T que se puede usar. Este depende de varios factores dinámicos de la planta citados anteriormente.

Desde otro punto de vista este período T es función también del sistema de adquisición de datos, debido a que el tratamiento y codificación de la señal requieren un tiempo para realizarse.

Así pues en la adquisición de un dato se tiene un gasto de tiempo como lo muestra la figura ref 7' Pag 90

$$T = n(t_c, t_t, t_p)$$

∴ el mínimo T dependerá del número de lazos (v.g. $n = 4$, y $t_{total} = 90 \text{ ms}$) se debe escoger un período

$$T = 4(90 \text{ ms}) = 360 \text{ ms}$$

Lo cuál es bastante bueno aún frente a procesos con variables dinámicas rápidas como flujo donde se recomienda $T = 1 \text{ seg}$.

Aplicaciones:

El control DDC se ha aplicado con buenos resultados reemplazando a los controladores analógicos en casi todos los casos. El uso de los minis se ha desarrollado ampliamente en grandes plantas y áreas, desde la fabricación de acero, sistemas de generación de potencia, plantas químicas e industrias de comunicaciones y transportes. Existen ya listas interminables que muestran la aceptación del control DDC. El uso de los micros abre ahora la oportunidad al pequeño productor de usar también el DDC para

mejorar tanto cualitativa como cuantitativamente su producción.

Como ejemplo sencillo y con posible comprobación inmediata sería el caso de un control de temperatura en un cambiador de calor. El sistema de control mantiene la temperatura de un flujo manipulando el flujo de combustible de un elemento calentador (control en cascada). El sistema intercambiador tiene un comportamiento que puede ser modelado por 1er orden ó 2° orden con retraso.

El modelo se obtiene de una ecuación de balance de calor. Suponiendo que el sistema no tiene pérdidas. Sea el calor demandado:

$$Q_0 = F_D K_D (T_0 - T_i) \quad (1)$$

F_D = flujo de salida lb/hr

T_0 = temperatura de salida

T_i = temperatura de entrada

K_D = constante capacidad calorífica BTU/16°F

El calor administrado es proporcional al combustible quemado

$$Q_1 = K_1 F_1 \quad (2)$$

K_1 = eficiencia de los quemadores

por balance de calor obtenemos el flujo F_1 en función del calor demandado

$$F_1 = \frac{1}{K_1} [F_D K_D (T_0 - T_i)] \quad (3)$$

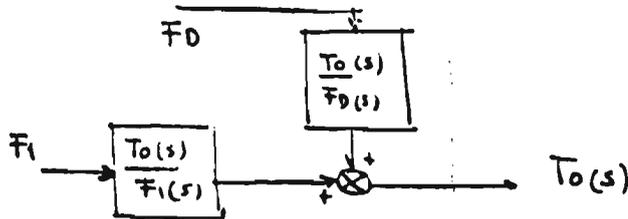
Asumiendo que la temperatura de entrada es más o menos constante tenemos la variación del flujo respecto a T_0

$$\frac{\partial F_1}{\partial T_0} = F_D \frac{K_D}{K_1} \quad (4)$$

En forma de diferencia queda definida la ganancia del lazo de control como

$$\frac{\Delta T_o}{\Delta F_i} = \frac{K_1}{K_2} \frac{1}{F_D} \quad (5)$$

El diagrama linealizado del cambiador se muestra en la figura



El controlador PID con algoritmo de velocidad es usado y sus parámetros son obtenidos cuidadosamente, en función del comportamiento de los procesos. Se deben notar también las características de no linealidad en la respuesta y los rangos propios de los parámetros para así proceder a programar el algoritmo de modo que se adapte lo mejor posible al sistema.

La estabilidad del sistema debe estudiarse tratando de evitar polos cercanos a -1 en el plano z .

Removiendo estos polos se obtiene un controlador PI que para la planta dada por

$$G(s) = \frac{-0.0103 e^{-2.3s}}{(12.55s + 1)} \quad (6)$$

es:

$$D(z) = \frac{1 - e^{-\lambda T}}{k(C_1 + C_2)(3 - ze^{\lambda T})} \frac{1 - e^{-\gamma_0 T} z^{-1}}{1 - z^{-1}}$$

Se obtiene

$$T_i = 12 \text{ min}$$

$$K_c = -325 \frac{\text{lb/min}}{\text{°F}}$$

La respuesta del sistema y la variable manipulada se muestran en las figuras 14 y 16

Diagrama de flujo simplificado del programa adquisitor.

- 1ª Selección del puerto R para uso en mediciones de la variable x. (temperatura, presión, flujo etc.).
- 2ª Inicio del multiplex con generación de señales de control para conversión A/D.
- 3ª Lectura del dato, identificación, prueba de valores de alarma y almacenamiento en el bloque correspondiente
- 4ª Control al programa del "Controlador"
- 5ª Espera de resultados y se repiten la parte 1ª y 2ª pero ahora para salida es decir entregando el comando a los elementos del control.

El paso 4ª se realiza entonces a partir de valores medidos y calcula la salida o acción a tomar en base a esa información y a parámetros característicos del controlador como KI, KD, Kp.

B. Diagramas de flujo de los algoritmos de control

1. Algoritmo PI de velocidad

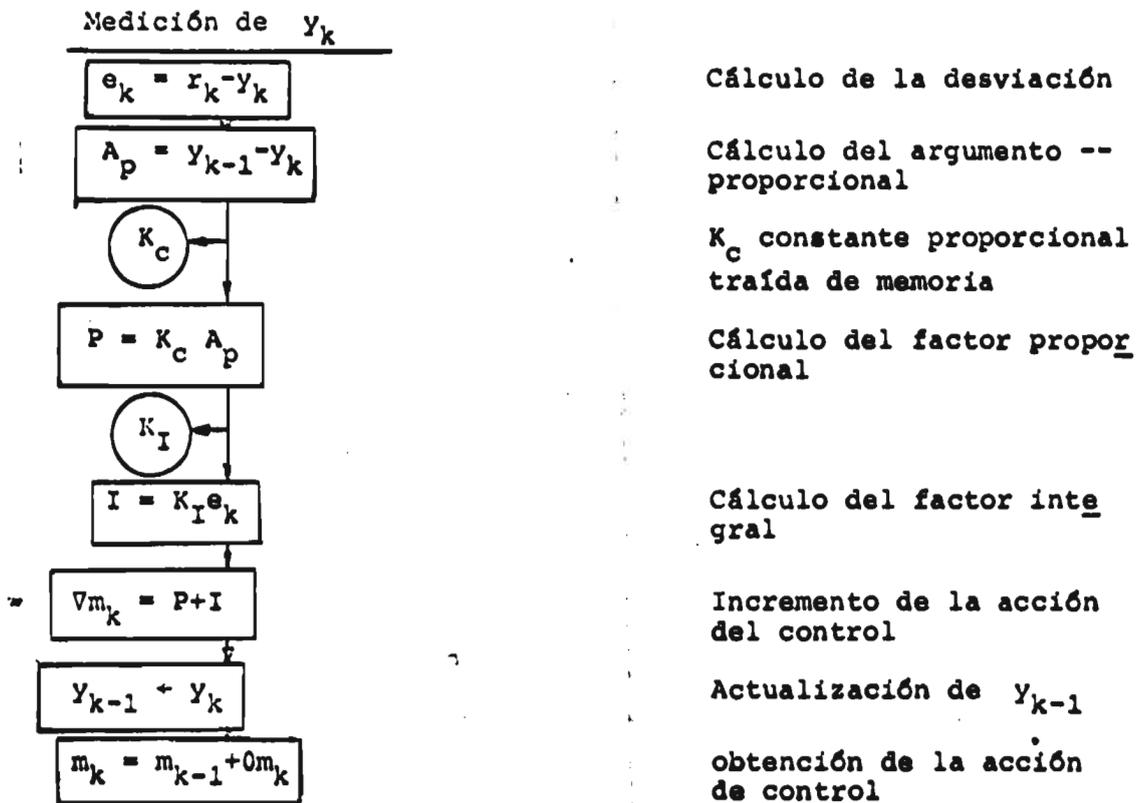
a) Ecuación en z $D(z) = K_c \frac{z - (1 - \frac{T}{T})}{z - 1}$

b) Ecuación recursiva representación directa

$$m_k = m_{k-1} + K_c (y_{k-1} - y_k) + K_I e_k$$

c) Diagrama de flujo

Comentarios



Acción de control

Cálculo de la desviación

Cálculo del argumento --
proporcional

K_c constante proporcional
traída de memoria

Cálculo del factor proporcional

Cálculo del factor integral

Incremento de la acción
del control

Actualización de y_{k-1}

obtención de la acción
de control

2. Algoritmo PID de velocidad

a) Representación en z

$$D(z) = K_C \frac{(1+K_D)z^2 - (1+2K_D - K_I)z + K_D}{z(z-1)}$$

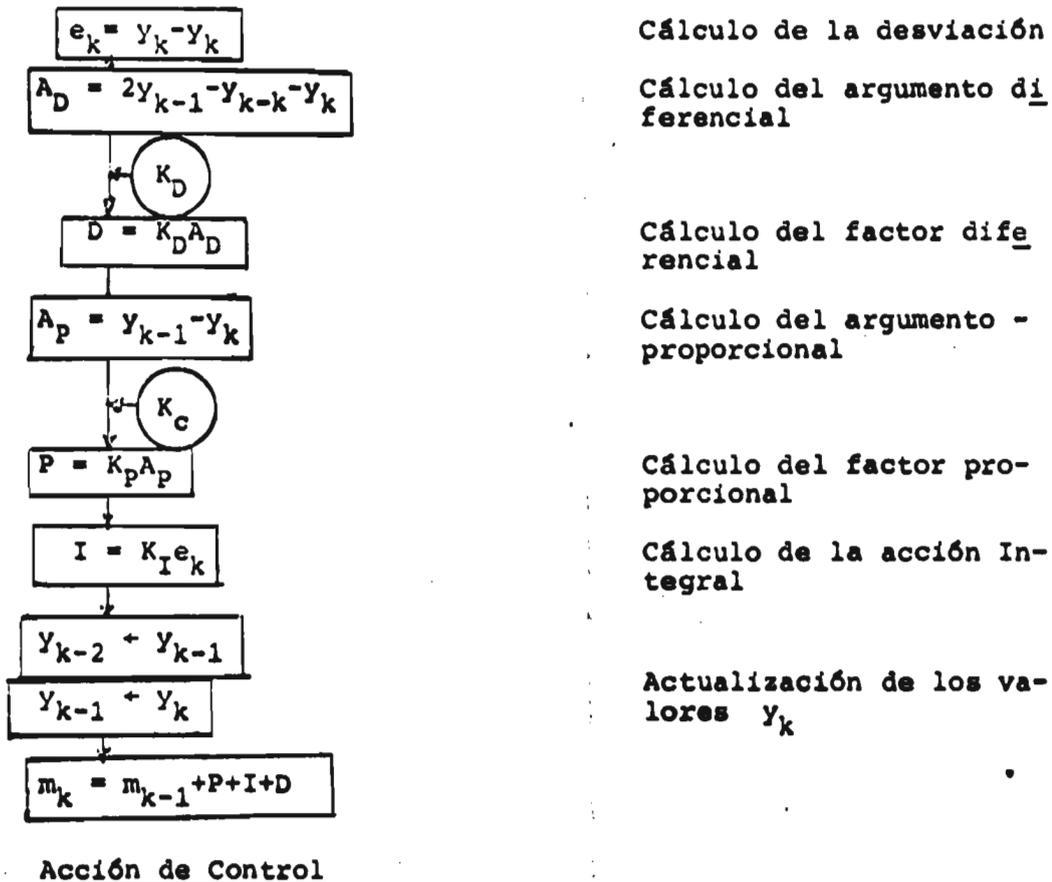
b) Ecuación de recurrencia

$$m_k = m_{k-1} + K_C (y_{k-1} - y_k) + K_I (y_k - y_k) + u_D (2y_{k-1} - y_{k-2} - y_k)$$

c) Diagrama de flujo

Comentarios

Medición de y_k



4. Algoritmo FTS para un proceso de 2° orden

a) Ecuación en z
$$D(z) = \frac{K(z-a)^2}{(z-1) [(1-a)^2 z - a(1-a-T/\tau)]}$$

sea $(1-a)^2 = A$

$A+B = C$

$a(1-a-T/\tau) = B$

b) Ecuación recurrente

$$m_k = \frac{C}{A} m_{k-1} - B m_{k-2} + K(e_k - 2ae_{k-1} + a^2 e_{k-2})$$

Transformando a variable de estado

$$X_{1,k+1} = X_{2,k}$$

$$X_{2,k+1} = X_{3,k}$$

$$X_{3,k+1} = a_2 X_{2,k} - a_1 X_{3,k} + e_k$$

$$u_{k+1} = b_1 X_{3,k+1} + b_2 X_{2,k+1} + b_3 X_{1,k+1} + b_0 e_k$$

c) Diagrama de flujo

medición de y_k

$$e_k = Y_k - y_k$$

Cálculo del error

$$A + X_{3,k}$$

Guarda valor anterior de X_3

$$X_{3,k+1} = -a_2 X_{2,k} - a_1 X_{3,k} + e_k$$

Cálculo de los estados actuales

$$X_{1,k+1} = X_{2,k}$$

Actualización de los estados de 1 a N-1

$$X_{2,k+1} = A$$

$$u_{k+1} = b_1 X_{3,k+1} + b_2 X_{2,k+1} + b_3 X_{1,k+1} + b_0 e_k$$

Acción de control

5. Algoritmo Dead beat para un proceso con retraso $\theta = NT$

a) Ecuación en z

$$D(z) = K \frac{1 - az^{-1}}{1 - az^{-(N+1)}}$$

b) Ecuación recurrente por variable de estado

$$x_{1k+1} = x_{2k}$$

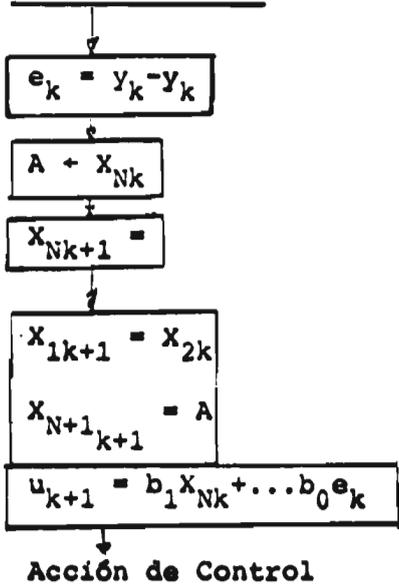
⋮

$$x_{Nk+1} = -a_1 x_{Nk} - a_2 x_{N-1k} - \dots + a_k$$

$$u_{k+1} = b_1 x_{Nk} + \dots + b_0 e_k$$

c) Diagrama de flujo

Medición de y_k



Cálculo del error

Guarda valor de x_{3k}

Cálculo de x_{Nk+1} por ec b)2

Actualizar los demás estados de 1 a N-1

6.- Algoritmo de Dahlin para un proceso de 2° orden con retraso (Modificado).

a) Ecuación en z

$$D(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + \dots + a_{N+2} z^{-(N+2)}}$$

b) En forma recurrente de variable de estado se realiza igual que el algoritmo 5.

7.- Algoritmo para realimentación con variable de estado. Usando medición múltiple proceso de 1er orden con retraso

a) Ecuación en z

$$D(z) = \left[\frac{1+\alpha}{1-\alpha}, \frac{\alpha^2}{1-\alpha} \right]$$

La ecuación lineal correspondiente es

$$D(z) = \frac{1}{\alpha-1} \frac{\alpha(1+\alpha) - (\alpha^2 + \alpha + 1)z}{2 + (1+\alpha)}$$

El algoritmo corresponde a un modo PI que ya a sido tratado.

Los demás algoritmos que se obtienen de los métodos de Dahlin, y Kalman son implementables con PI, PID cuyos diagramas de flujo ya han sido mostrados.

8.- Algoritmo E_0 /Retroalimentación.

Este ha sido tratado extensivamente en las páginas 36, 37 mostrán dosedose su diagrama de flujo en la página 39.

Referencias

- 1.- Cadzow Martens
Discrete Time and Computer Control Systems
Prentice Hall 1970
- 2.- Smith Cecil I.
Digital Computer Process Control.
Intext Education 1972.
- 3.- Takahashi Y. et at.
Control and Dynamic Systems.
Addison Wesley 1960.
- 4.- Takahashi Y.
On line Digital Modern Control & Data Processing
Lecture Series CIEA Ago. 1975.
- 5.- Tov J. T.
Digital & Sampled data Control Systems.
Mc. Graw Hill 1960
- 6.- Monroe.
Digital Processes for Sampled Data Systems.
J. Wiley 1962
- 7.- Lefkowitz
Notes for Short course on RT computers, Data Acquisition and
Control
Case W. University Print. Enero 1975.

- 21.- JAMES G. MILLER
DDC - THE INHERENT DIFFERENCE.
INSTRUMENTS AND CONTROL SYSTEMS, JULIO 1970
- ✓22.- R.D. PEROLT AND B.F. SOWELL
A "NATURAL" MODE CONTROLLER ALGORITHM FOR DDC.
ISA JOURNAL, SEPTEMBRE 1966.
- ✓23.- ROGER H. BAKKE
THEORETICAL ASPECTS OF DIRECT DIGITAL CONTROL.
ISA TRANSACTIONS , VOL 8, N° 4 .
- 24.- T.L. WILLMOTT
A SURVEY OF SOFTWARE FOR DIRECT DIGITAL CONTROL.
ISA TRANSACTIONS VOL 3 N° 4.
- 25.- JAMES E. HEATON.
WHAT'S IN STORE FOR THE CONTROL SPECIALIST ?
INSTRUMENTS AND CONTROL SYSTEMS, ABRIL 1974.
- 26.- WALTER G. RUDD.
MICROCOMPUTERS IN INSTRUMENTATION AND CONTROL.
INSTRUMENTS AND CONTROL SYSTEMS, ABRIL 1971
- 27.- D. DRINKER.
DDC: THE MONEY MACHINE.
INSTRUMENTS AND CONTROL SYSTEMS | SEPTIEMBRE 1973.
- 28.- RAY GADDIS AND CHARLES F. MOORE.
MICROS AND MINIS WORK TOGETHER IN DATA ACQUISITION AND CONTROL.
INSTRUMENTS AND CONTROL SYSTEMS, MAIO 1975
- 29.- ALAN J. WEISSBERGER,
APPLICATION IDEAS FOR MICROPROCESSORS.
INSTRUMENTS AND CONTROL SYSTEMS, OCTUBRE 1975.
- 30.- M.S. BECK.
DIRECT DIGITAL CONTROL OF CHEMICAL PROCESSES.
CONTROL, SEPTIEMBRE 1967.