

# UNIVERSIDAD NACIONAL DE INGENIERIA

Facultad de Ingeniería Eléctrica y Electrónica



## DISEÑO DE UN CONTROLADOR PID DIGITAL USANDO MICROPROCESADOR

**T E S I S**

Para optar el Título Profesional de  
Ingeniero Electrónico

**CESAR ALBERTO BRICEÑO ARANDA**

PROMOCION 1979 - 1

Lima - Perú  
1985

UNIVERSIDAD NACIONAL DE INGENIERIA

Programa Académico de Ingeniería Mecánica y Eléctrica

DISEÑO DE UN CONTROLADOR PID DIGITAL, USANDO  
MICROPROCESADOR

T E S I S

Para Optar el Título Profesional de  
INGENIERO ELECTRONICO

Presentada por:

CESAR ALBERTO BRICEÑO ARANDA

Promoción 1979 - I

Lima - Perú

1984

UNIVERSIDAD NACIONAL DE INGENIERIA-

Programa Académico de Ingeniería Mecánica y Electrica

DISEÑO DE UN CONTROLADOR PIDDIGITAL, USANDO  
MICROPROCESADOR

Tesis para optar el Título Profesional de:

INGENIERO ELECTRONICO

Presentada por:

CESAR ALBERTO BRICEÑO ARANDA

de la Promoción 1979 - I

Lima - Perú

1984

E X T R A C T O

Modernamente, el microprocesador es empleado como elemento de control digital en tiempo real, el mismo que constituye el CPU de un sistema microcomputador.

La importancia fundamental del uso del microprocesador para este tipo de controlador radica en que éste brinda un ahorro considerable en la lógica digital cableada (Hardware), a cambio de una relativa complejidad en la programación, es decir, Software.

## T A B L A D E C O N T E N I D O S

	Pag.
Prefacio	1
Capítulo I: FUNDAMENTO TEORICO	3
Introducción.	3
1.1 Clasificación de los sistemas de control reali mentado.	3
1.2 Clasificación de los sistemas de control indus triales.	4
1.2.1 Controlador Proporcional.	4
1.2.2 Controlador Integral.	7
1.2.3 Controlador Derivativo.	9
1.2.4 Controlador de dos posiciones ON-OFF.	10
1.2.5 Controlador PID.	10
1.3 Métodos para el análisis de sistemas de control digital.	12
1.3.1 Método de la transformada $z$ .	13
1.3.2 Método del espacio de estado.	16
1.3.3 Digitalización de un controlador PID.	18
1.4 Análisis de la estabilidad de los sistemas de control digital.	20
1.4.1 Análisis en el plano $z$ .	20
1.4.2 Análisis con el método de espacio de es- tado.	21
1.4.3 Segundo Método de Liapunov.	21
1.5 El microprocesador como controlador.	22
1.5.1 El microprocesador 6802 Motorola.	23

1.5.1.1	Arquitectura.	24
1.5.1.2	Componentes del sistema microcomputador MEK6802D5E.	29
1.5.1.3	Manejo de interrupciones.	32
1.5.1.4	PIA (6821), periférico programable.	35
Capítulo II: CONSIDERACIONES PARA EL USO DE MICROPROCESADORES EN SISTEMAS DE CONTROL		40
	Introducción.	40
2.1	Consideraciones preliminares de diseño.	40
2.2	Información de la arquitectura y estructura del sistema.	42
2.3	Algoritmo de control.	44
2.3.1	Tipos de Algoritmo.	44
2.4	Software	46
2.4.1	Lenguajes.	49
2.4.2	Efectos de la longitud de palabra finita sobre la controlabilidad y la localización de los polos en lazo cerrado.	51
2.4.3	Tiempo de retardo en sistemas de control con microprocesadores.	53
2.4.4	Efectos de la cuantificación en el error de estado estable.	54
Capítulo III: DEDUCCION DEL ALGORITMO A IMPLEMENTAR		59
3.1	Estructura del algoritmo.	59
3.2	Algoritmo de regulación.	61
3.2.1	Algoritmo PID.	61
3.2.2	Algoritmo PIDT1.	65

Capítulo IV: PROGRAMACION Y APLICACION DEL CONTROLADOR PID	69
Introducción.	69
4.1 Descripción del programa principal.	69
4.2 Subprogramas matemáticos	71
4.2.1 Subrutina de multiplicación con punto decimal flotante.	72
4.2.2 Subrutinas de adición y resta con punto decimal flotante.	75
4.3 Subrutina de retardo (Delay).	80
4.4 Inicialización y programación del PIA.	80
CONCLUSIONES	82
BIBLIOGRAFIA	84
ANEXO A: Tabla de la transformada de Laplace, transformada $z$ , y transformada $z$ modificada.	86
ANEXO B: Instrucciones del microprocesador 6802 Motorola.	89
ANEXO C: Hojas de datos técnicos de los componentes del microcomputador MEK6802D5E.	92
C.1 Microprocesador: MC6802.	93
C.2 Memoria RAM: MCM6810.	94
C.3 PIA: MC6821.	95
C.4 Memoria ROM: MCM68A316E.	96

## PREFACIO

En las últimas tres décadas los sistemas de control han alcanzado considerable complejidad, por lo tanto, su tratamiento requería de una teoría de control moderna, la misma que ha alcanzado un alto grado de desarrollo. Sin embargo, la implementación física de dichos sistemas resultaba en algunos casos prohibitiva, dados los altos costos y dificultades técnicas. La aparición del microprocesador ha salvado estos problemas, permitiendo, aún nuevas perspectivas en el desarrollo de los sistemas de control.

En el presente trabajo se presenta una aplicación del microprocesador para el diseño de un controlador proporcional, integral y derivativo (PID), el mismo que es muy utilizado en los sistemas de control industriales. Para lo cual se ha desarrollado un algoritmo a partir de la ecuación diferencial del regulador PID. Donde la parte integral se ha aproximado por la suma de áreas rectangulares, la parte derivativa, por la ecuación de diferencia. Dicho algoritmo ha sido programado en lenguaje de máquina, utilizando el programa principal varias subrutinas, siendo estas la de suma, resta, multiplicación con punto decimal flotante.

Dentro de los alcances del presente trabajo esta la presentación de una metodología para el diseño de controladores PID, usando microprocesador, sin intentar mostrar una solución óptima. Dadas las restricciones de disponibilidad de componentes electrónicos, sobre todo, de la familia Motorola en nuestro medio. Para este tipo de trabajos no se ha llegado a realizarse la implementación física. Pero se espera que en un futuro próximo se dispondrá en nuestro medio de mas información y disponibilidad de dichos componentes, lo que permitirá implementar este y otros diseños, mas complejos y sofisticados.

El desarrollo del presente trabajo se ha hecho por capítulos atendiendo a una secuencia lógica y didáctica, la misma que es como sigue:

En el Capítulo I, se presenta el fundamento teórico que contempla la clasificación de los sistemas de control realimentados industriales y los diferentes métodos para el análisis del control digital. También, se presenta una introducción breve sobre el microprocesador.

En el Capítulo II, se presentan las consideraciones para el uso del microprocesador en sistemas de control.

En el Capítulo III, se presenta la deducción del algoritmo a implementarse, finalmente.

En el Capítulo IV, se presenta la programación y aplicación del controlador PID, mostrando los programas a utilizarse.

Deseo expresar mis agradecimientos a mi asesor, el ingeniero, Miguel Sánchez Eravo, por el apoyo que me brindó durante el desarrollo del presente trabajo. Así mismo, deseo agradecer al ingeniero Arturo Rojas Moreno, por la valiosa información que me proporcionó, en forma desinteresada, la misma que ha sido de mucha ayuda en mi trabajo de tesis.



## CAPITULO I

### FUNDAMENTO TEORICO

Introducción.- El uso del Control Automático se ha hecho prácticamente imprescindible en cualquier diseño de ingeniería.

Lo que como es lógica consecuencia, ha dado como resultado que en la industria tenga una gran aplicación, como la metalúrgica, Procesos Químicos, Industria bélica, etc.

#### 1.1 Clasificación de los Sistemas de Control Realimentado

El notable avance tecnológico experimentado en las últimas décadas, hace muy difícil hacer una clasificación.

Así, un Sistema de Control para el cual se conocen todas sus funciones de transferencia, puede clasificarse según el tipo de sistema. Dicho tipo se obtiene no solamente a partir de la información, de los elementos de control, sino también de los elementos del sistema.

Algunos ingenieros de Control en la industria hablan de acciones de control mientras que, en otras industrias se usa la clasificación de tipo de control.

De acuerdo a los componentes del sistema podríamos intentar la siguiente clasificación:

- 1.- Sistemas Lineales
- 2.- Sistemas no Lineales
- 3.- Sistemas por Anticipación (feedward)
- 4.- Sistemas Adaptivos
- 5.- Sistemas de Control Optimo

Según la Señal de Control:

- 1.- Sistemas de datos continuos
- 2.- Sistemas de datos discretos o sistemas muestreados.

El presente trabajo se ubica dentro de los sistemas de datos discretos.

## 1.2 Clasificación de los Sistemas de Control Industriales

De acuerdo al Controlador usado se clasifican en:

### 1.2.1 Controlador Proporcional

El siguiente diagrama de bloques muestra un Controlador Proporcional para un sistema mecánico (Fig. I.1).

$$(1) \quad W_r(s) - W_c(s) = E(s)$$

$$(2) \quad Q(s) = K_p E(s)$$

donde se puede ver que el gasto de combustible es proporcional al error.

Para analizar el efecto de la perturbación sobre el sistema tendríamos (Fig. I.2)

$$(3) \quad \frac{\Delta W_c(s)}{-T_d(s)} = \frac{G(s)}{1 + G(s)H(s)} = \frac{1}{I s + (b + K_p K_1)}$$

$$\text{si } T_d(s) = -T D/s \quad (4)$$

$$\text{se tiene } \Delta W_c(s) = - \frac{TD}{s [I s + (b + K_p K_1)]} \quad (5)$$

$$(6) \quad \Delta W_{css}(t) = \lim_{t \rightarrow \infty} \Delta W_c(t) = \lim_{s \rightarrow 0} s \Delta W_c(s)$$

$$(7) \quad \Delta W_{css}(t) = - \frac{TD}{(b + K_p K_1)}$$

De la ecuación (7) puede observarse que el momento adicional (torque) requerido en la carga resulta de una reducción en la velocidad del sistema.

Esta aplicación presentada sobre control de velocidad ilustra el uso de un control proporcional. Puede verse que cuando es necesaria una corrección sostenida, se obtiene como resultado un error sostenido (en estado estacionario).

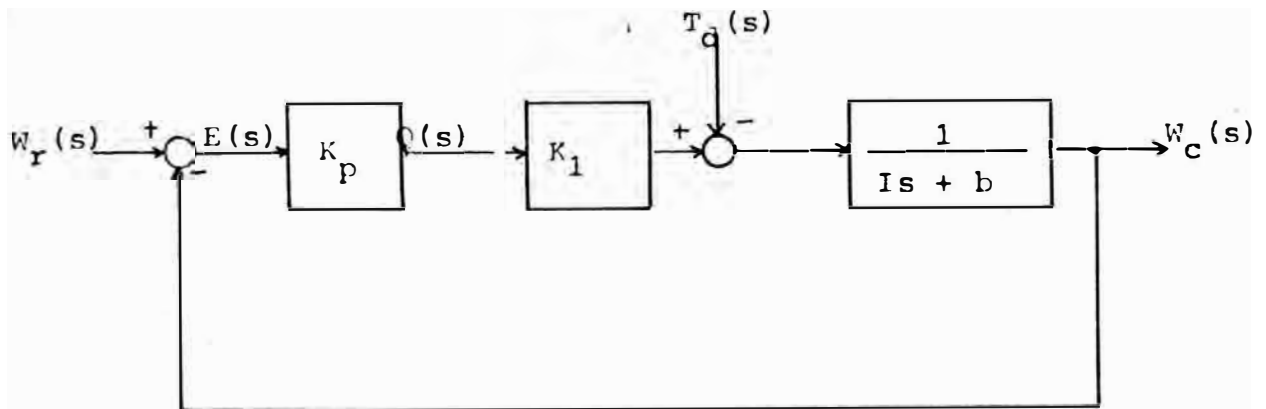


Fig. (1.1)

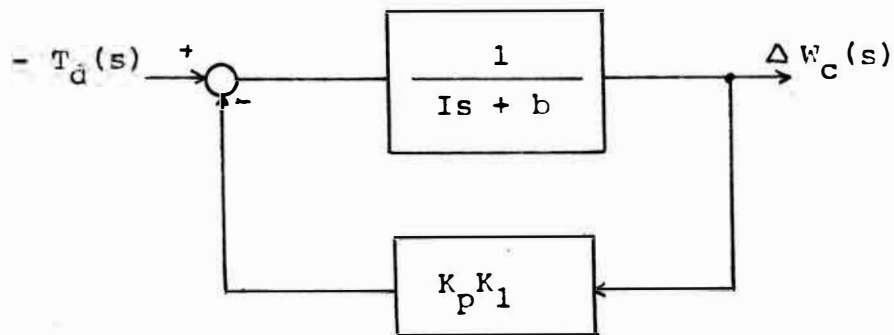


Fig. (1.2)

### I.2.2 Controlador Integral

En la Fig. I.3, el diagrama de bloques muestra para el ejemplo anterior un Controlador o Regulador Integral.

Si estudiamos el efecto de la perturbación el sistema para una perturbación tipo escalón.

$$Td(s) = - \frac{TD}{s} \quad (\text{Fig. I.4})$$

$$(8) \frac{\Delta W_c(s)}{-Td(s)} = \frac{s}{Is^2 + bs + K_i k_1}$$

$$(9) \Delta W_c(s) = \frac{-TD}{Is^2 + bs + K_i k_1}$$

$$(10) \Delta W_{CSS}(t) = \lim_{s \rightarrow 0} s \Delta W_c(s) = 0$$

La principal virtud de este controlador es la de ser capaz de alcanzar la corrección de manera que eliminó el error en la velocidad del sistema.

La aplicación dada ilustra esto; o sea que aunque tendiendo a producir una respuesta oscilatoria, también tiende a eliminar cualquier error en el estado estable del sistema.

Además presenta el inconveniente de poder hacer inestable al sistema, lo cual se evita adionándole un controlador proporcional.

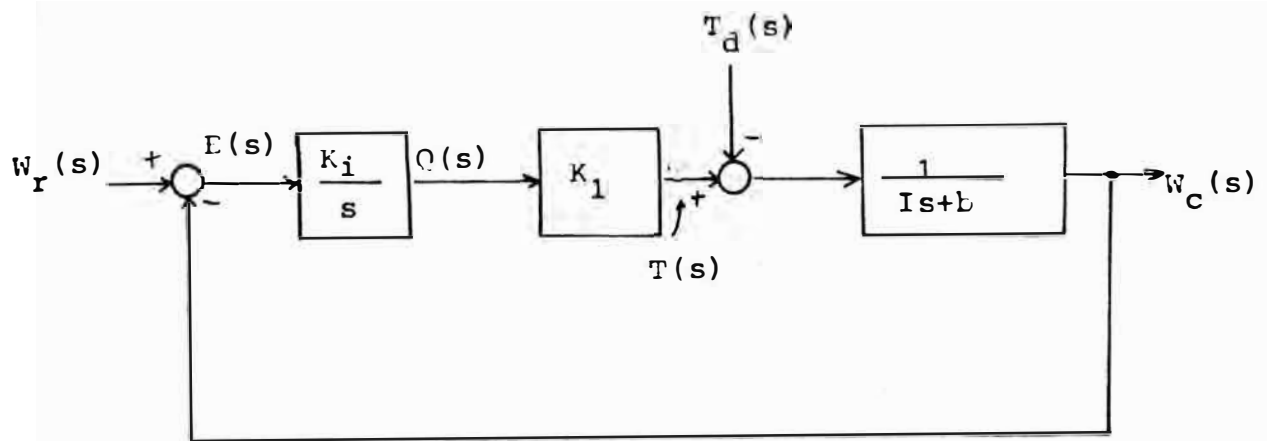


Fig. (1.3)

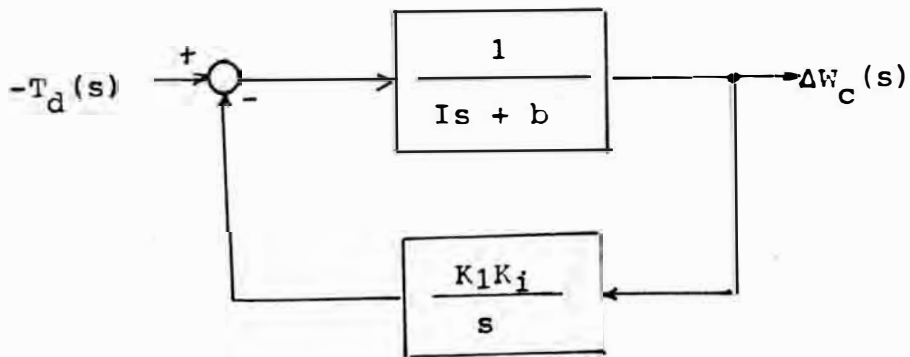


Fig. (1.4)

### 1.2.3 Controlador Derivativo

En el control derivativo se hace una corrección que es proporcional a la derivada del error respecto al tiempo, expresado matemáticamente.

$$(11) \quad \delta_m = K_d \frac{d e}{dt}$$

$K_d$  = ganancia del Control Derivativo.

El control derivativo es útil porque responde a la rapidez de cambio de error y puede producir una corrección significativa antes que la magnitud real del error sea grande.

Por esta razón, se dice que se anticipa al error y, de esta manera, inicia una prematura corrección del error. El control derivativo no puede usarse sólo porque no responde a un error en estado estable, debe usarse en combinación con otras acciones de control. (Fig. I.5).

La función de transferencia del sistema es:

$$(12) \quad \frac{\theta_o(s)}{\theta_i(s)} = \frac{G(s)}{1 + G(s)} = \frac{K_p + K_d s}{I s^2 + (b + K_d) s + K_p}$$

De la ecuación característica se puede observar que el control derivativo aumenta el amortiguamiento del sistema.

Para mayor información el sistema es un posi

cionador angular.

#### 1.2.4 Controlador de dos posiciones (ON-OFF)

En un sistema de control de dos posiciones, el elemento final de control tiene solamente dos posiciones fijas que son "todo" o "nada".

La variable de control debe moverse continuamente entre estos dos límites.

El rango entre el cual debe moverse se llama diferencial de la variable de control. Este tipo de controlador aunque nos da una pobre o mucha corrección del sistema es relativamente simple y barato, ésta es la razón por la que es ampliamente usado en la industria.

#### 1.2.5 Controlador PID

El controlador PID viene a ser la suma de los tres controladores: el proporcional, el derivativo y el integral.

En esta sección como es obvio veremos un controlador analógico.

Este controlador actúa tanto sobre la parte estática como dinámica del sistema a controlar.

Cada controlador contribuye con la mejor de su característica.

La parte integral del controlador nos reduce el error en estado estable y el efecto de una perturba -



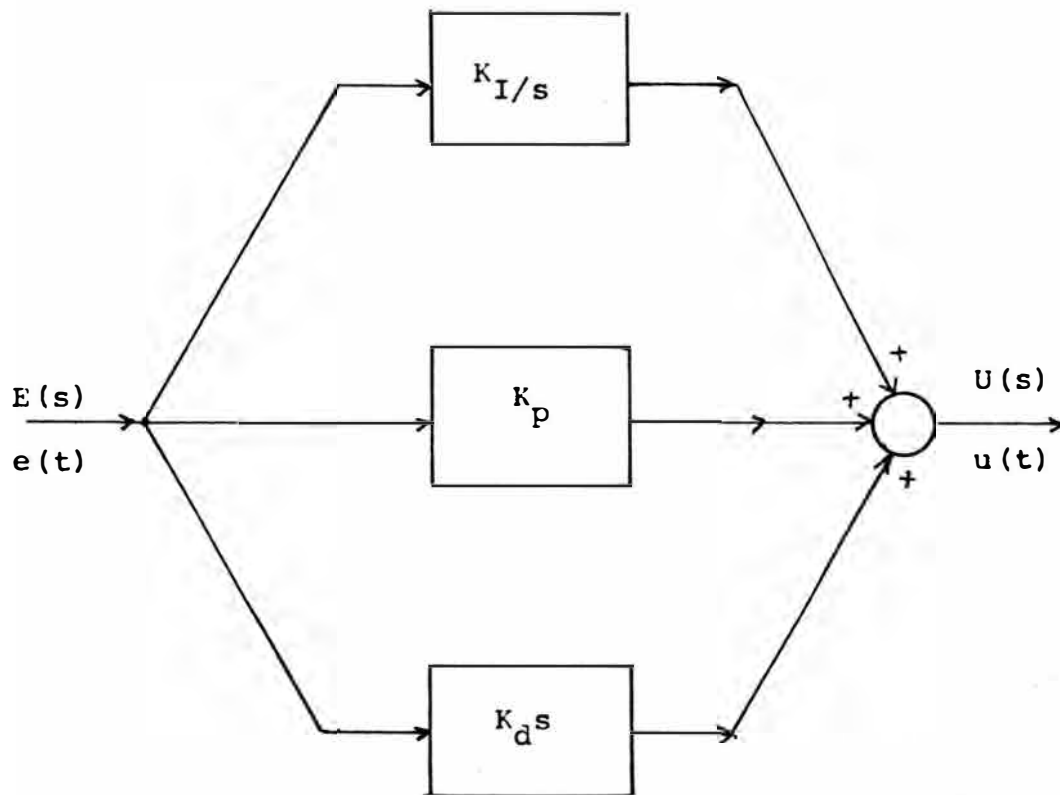


Fig. (1.6)

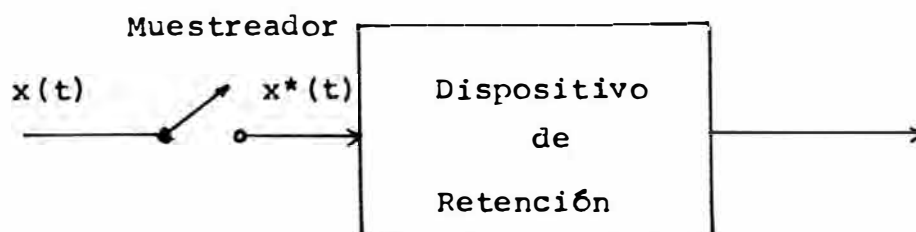


Fig. (1.7)

ción inadecuada.

Como el control derivativo provee una acción anticipatoria reduce los efectos de los sobreimpulsos en la respuesta. La fig. I.6 nos muestra la suma de las acciones.

Para el controlador PID digital de la sección 1.3.3 se aplican los mismos principios del Control Analógico básicamente.

### 1.3 Métodos para el análisis de Sistemas de Control Digital

Introducción.- Los sistemas de datos muestreados o sistemas de tiempo discreto, son sistemas dinámicos en los cuales una o más variables pueden variar solamente en ciertos instantes. Estos instantes se pueden indicar por  $KT$  o  $T_k$  ( $K = 0, 1, 2, \dots$ ); pueden especificar el momento en el cual se realiza alguna medición física o el tiempo en el cual se lee la memoria de una computadora digital, se toma el intervalo de tiempo entre dos instantes suficientemente pequeños, de manera que los datos de tiempo entre esos instantes pueden ser aproximados por interpolación simple.

La diferencia fundamental entre los sistemas de control continuo y los de tiempo discreto es que en estos últimos las señales de control aparecen en forma de tren de pulsos.

El método análogo al de la transformada de Laplace (sistemas continuos invariantes en el tiempo) para sistemas de tiempo discreto es el de la transformada Z.

### 1.3.1 Método de la Transformada Z

En la Fig. I.7 se muestra un dispositivo muestreado y de retención.

El dispositivo esencial en un sistema de tiempo discreto es el muestreador. En un muestreador convencional una <sup>γ</sup> lla de sierra para admitir una señal de entrada cada T segundos.

En la práctica, la duración del muestreo es muy breve en comparación con la corriente de tiempo más significativo de la planta. El muestreador convierte una señal continua en un tren de pulsos, producidos en los instantes de muestreo 0, T, 2T, ... , etc.

La función de transferencia del dispositivo de Retención, que viene a ser un filtro pasa bajo es

$$(13) \quad G_h(s) = \frac{1 - e^{-TS}}{s}$$

La función de este dispositivo es alisar la señal muestreada  $x^*(t)$  produciendo la señal  $X_h(t)$  que es corriente desde el último valor muestreado hasta disponer del próximo valor de muestreo.

Se puede considerar  $X^*(t)$  como una señal modulada según la Fig. I.8

$$(14) \quad X^*(t) = S_T(t) X(t)$$

$$(15) \quad T(t) = \sum_{K=-\infty}^{\infty} X(t) S(t - KT)$$

$$(16) \quad X^*(t) = \sum_{K=-\infty}^{\infty} X(KT) \delta(t - KT)$$

Como se muestra en las figuras I.9 , I.10, I.11

Si definimos las funciones temporales, como cero para  $T < 0$

La ecuación (16) queda de la siguiente forma:

$$(17) \quad X^*(t) = \sum_{K=0}^{\infty} X(t) \delta(t - KT) = \sum_{K=0}^{\infty} X(KT) \delta(t - KT)$$

Si a la última ecuación le tomamos la transformada de Laplace se tiene:

$$(18) \quad X^*(s) = L(X^*(t)) = \sum_{K=0}^{\infty} X(KT) e^{-KTS}$$

Se define  $e^{TS} = Z$  y se escribe  $X^*(s) = X(Z)$

$$(19) \quad X(Z) = X^*(s) = X^*\left(\frac{1}{T} \ln Z\right) = \sum_{K=0}^{\infty} X(KT) Z^{-K}$$

En la transformación Z solo se consideran los valores de la señal en los instantes de muestreo. Por tanto, la transformada Z de  $X(t)$  y la de  $X(t)$  dan el mismo resultado.

## MULSTREADOR

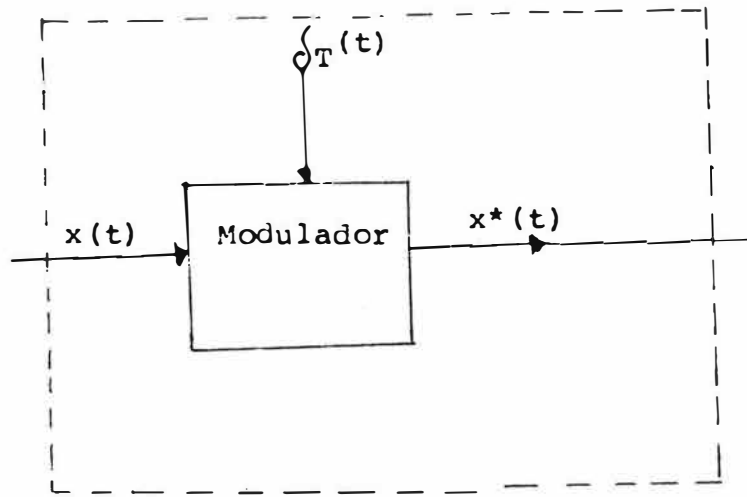


Fig. (1.8)

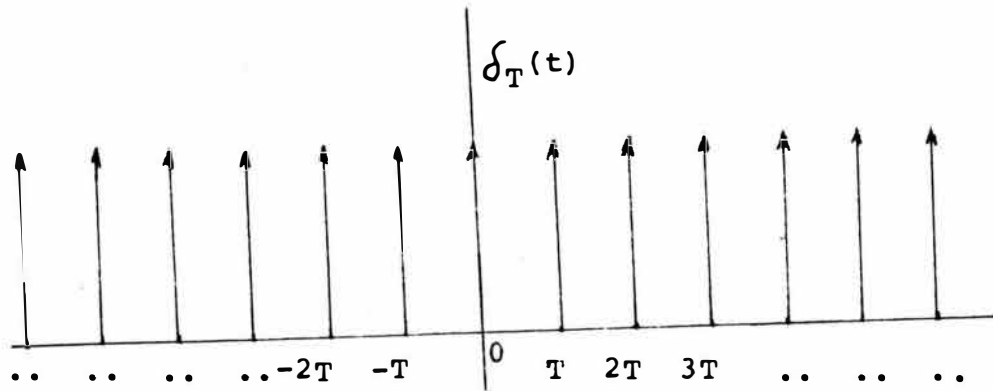


Fig. (1.9)

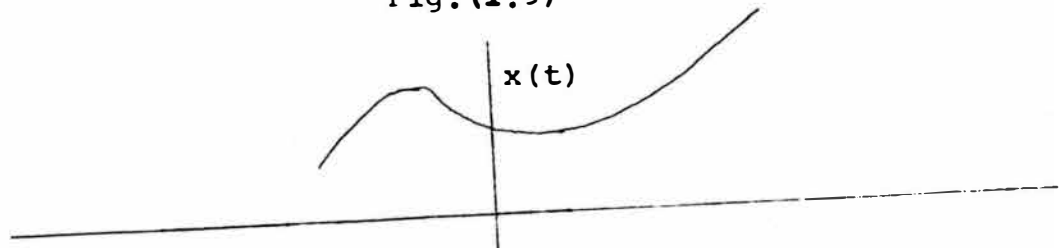


Fig. (1.10)

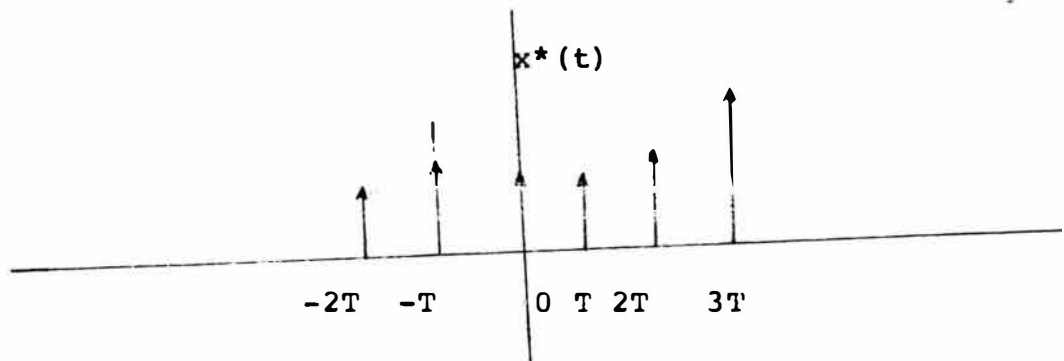


Fig. (1.11)

estado de sistemas lineales de tiempo discreto es:

$$(23) \quad X(K+1) = G(K) X(K) + H(K) U(K)$$

$$(24) \quad Y(K) = C(K) X(K) + D(K) U(K)$$

donde  $X(K)$  es el vector de estado

$U(K)$  es el sector de entrada

Para simplificar la notación se ha considerado  $K$  como  $KT$ , donde  $T$  es el período de muestreo.

Si el sistema lineal discreto en el tiempo es invariable en el tiempo, las ecuaciones (23) y (24) se convierten en

$$(25) \quad X(K+1) = G X(K) + H U(K)$$

$$(26) \quad Y(K) = C X(K) + D u(K)$$

donde por comodidad se ha denotado  $K$  por  $KT$ .

Estas últimas ecuaciones se pueden representar mediante un diagrama de bloques, como el de la Fig. I.12.

Para el lector interesado en esta sección puede referirse a la Bibliografía, las referencias

Se puede mencionar las ventajas de usar este método.

a.- Se utiliza para sistemas multivariados.

- b.- Para calculos con el computador digital resulta un método prácticamente natural.
- c.- Puede ser aplicable a ciertos tipos de sistemas no lineales y variables en el tiempo.

Sin embargo, el método de la transformada Z nos brinda la función de transferencia del sistema en forma compacta. Y es muchas veces, el método preferido por los diseñadores de sistemas de control digital.

### 1.3.3 Digitalización del controlador PID

Para terminar con esta breve introducción de los métodos usados para el análisis, expresaremos el controlador PID digitalizado refiriéndonos a la Fig. I.6.

K I/S puede ser aproximado por la transformada Z para una integración polinomial  $\frac{KI T(Z+1)}{2 (Z-1)}$  (27)

La derivada en el instante de muestreo (T) puede ser aproximado de la siguiente forma.

$$\left. \frac{de}{dt} \right|_{t=T} = \frac{e(KT) - e(K-1) T}{T} \quad (28)$$

tomando la transformada Z a ambos lados de la ecuación

$$G_d(z) = k_d \frac{z - 1}{T z} \quad (29)$$

Luego el diagrama de bloques quedaría así:

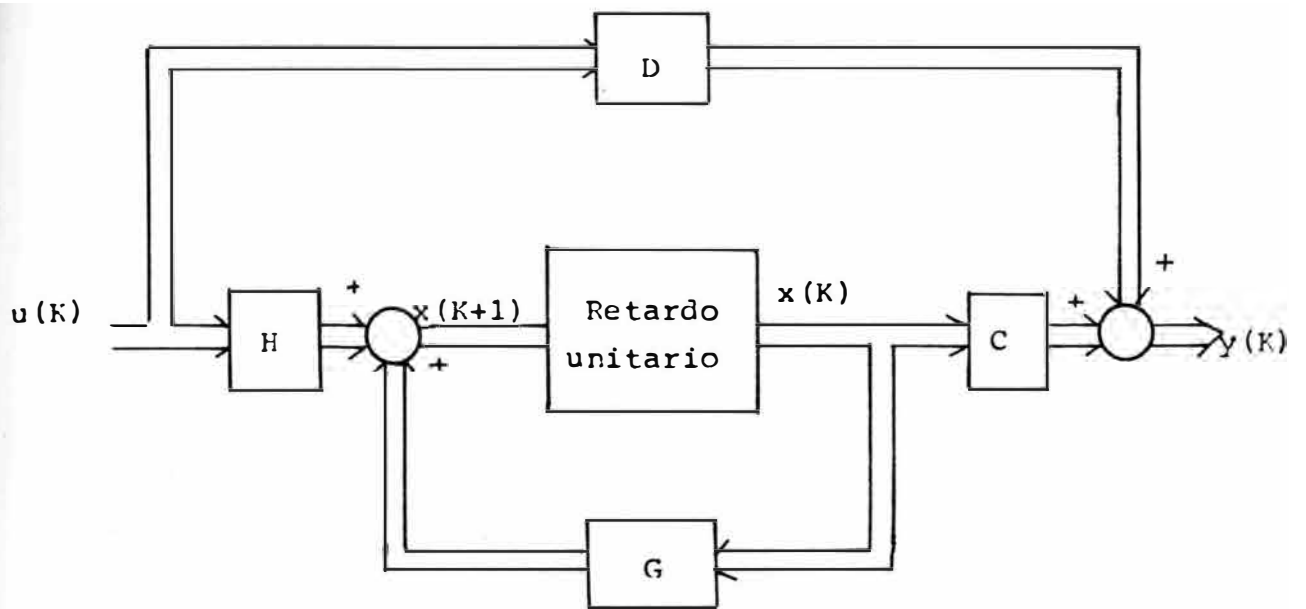


Fig. (1.12)

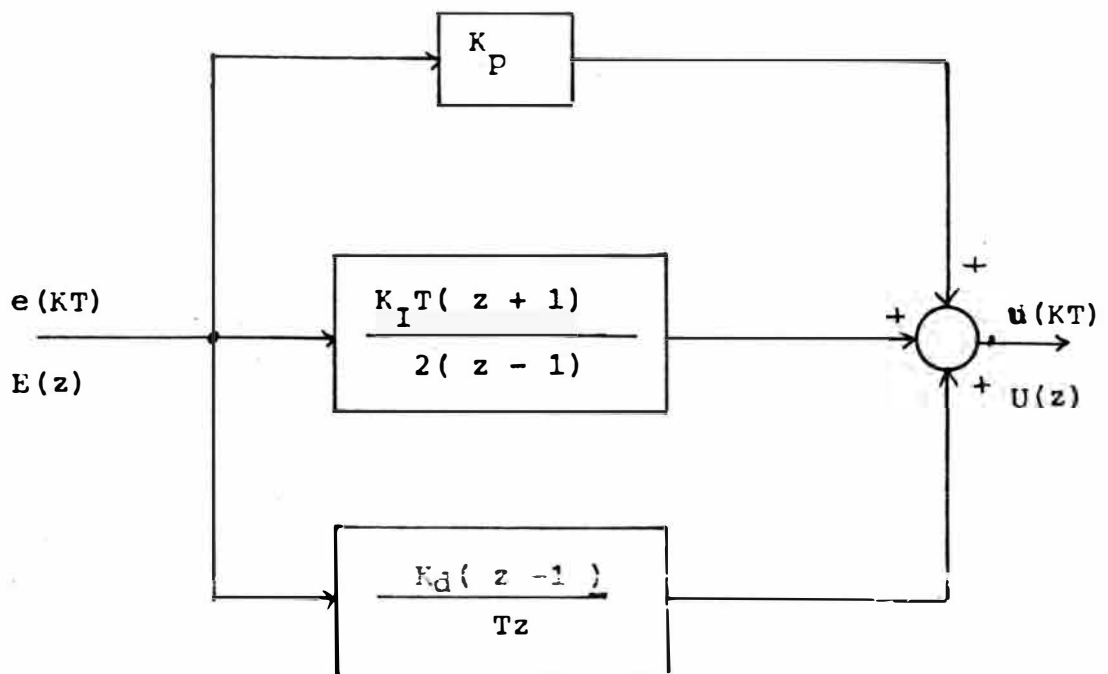


Fig. (1.13)



#### 1.4 Análisis de la Estabilidad de los Sistemas de Control Digital

Se verán tres maneras o métodos de analizar la estabilidad de sistemas de control digital.

##### 1.4.1 Análisis en el plano Z

Si la función de transferencia de un sistema de control digital es

$$(30) \quad \frac{C(Z)}{R(Z)} = \frac{G(Z)}{1 + GH(Z)}$$

La ecuación característica es:

$$(31) \quad 1 + GH(Z) = 0$$

Como podemos convertir este polinomio en  $Z$  en una relación entre dos polinomios en  $Z$ , para obtener estabilidad todos los raíces  $Z_i$  de la ecuación característica ec (31) deben quedar dentro del círculo unitario, o sea  $Z_i \rightarrow 1$

El sistema de lazo cerrado se vuelve inestable si cualquiera de los polos de lazo cerrado cae fuera del círculo unitario y/o hay polos múltiples sobre el círculo unitario.

Para aplicar el criterio de Routh, se hace una transformación en el polinomio de la ecuación (31).

Se cambia  $Z$  por  $\frac{r+1}{r-1}$ , si hay cambio de signo una vez analizado y aplicado el criterio de Routh,

una raíz cae fuera del círculo unitario.

#### 1.4.2 Análisis con el método de espacio de Estado

Puede deducirse de la sección 1.3.2 que la ecuación característica está dada por

$$(32) \quad |ZI - G| = 0$$

Luego el sistema discreto en el tiempo es estable si y solo si todas las raíces de la ecuación característica están en el círculo unitario centrado en el origen del plano Z.

#### 1.4.3 Segundo método de Liapunov

Sea el sistema de tiempo discreto descrito por:

$$(33) \quad X(K+1) = G X(K)$$

Se elige una función de Liapunov posible tal como

$$(34) \quad V(X(K)) = X'(K) P X(K)$$

$X(K)$  es el vector de estado

$X'(K)$  es el transpuesto de  $X(K)$

$P$  es una matriz positiva definida simétrica real, en lugar de  $\dot{V}(X)$  se usa la diferencia entre

$$(35) \quad V(X(K+1)) \text{ y } V(X(K))$$

$$(36) \quad \Delta V(X(K)) = V(X(K+1)) - V(X(K))$$

$$(37) \quad \Delta V(X(K)) = X'(K+1) P X(K+1) - X'(K) P X(K)$$

$$(38) \quad \Delta V(X(K)) = (G X(K))' P G X(K) - X'(K) P X(K)$$

$$(39) \quad \Delta V(X(K)) = X'(K) \{ G' P G - P \} X(K)$$

$$(40) \quad \Delta V(X(K)) = - X' Q X$$

$$(41) \quad \{ G' P G - P \} = - Q$$

$Q$  es una matriz positiva simétrica real para que el sistema sea asintóticamente estable.  $V(X(K))$  debe ser negativa definida. El lector interesado en esta sección puede ver en la Bibliografía las referencias y

### 1.5 El Microprocesador como Controlador

En la figura I.14, el diagrama de bloques muestra al Microcomputador como un controlador digital P.I.D.

A/D Convertidor analógico/Digital

u.P Microprocesador (Microcomputador estrictamente hablando)

D/A Convertidor Digital/Analógico

$G(Z)$  función de transferencia del controlador digital P.I.D.

$G_p(S)$  Planta o proceso a controlar.

### 1.5.1 El Microprocesador 6802 (Motorola)

El Microprocesador 6802 pertenece a la familia 6800 cuyo fabricante es Motorola.

Motorola es una de las principales fábricas de Microprocesadores y elementos de soporte de (u.P).

Se podría incluso afirmar que junto con Intel son los líderes y pioneros en Estados Unidos.

Intel fabrica la serie 8080 y toda su familia.

Además tenemos la Zilog que fabrica Z.80.

Otra fábrica es la Mos-Technology que fabrica el 6500.

La Rockwell y su PPS8 surgió como competencia de los 8080 y 6800. La sigmática que a mediados de la década del 70, fue comprada por la Philips de Holanda, lanzó el 2650 que es un micro potente aunque más lento que el 6800 y el 8080.

Tenemos también el SC/MP (simple cost-effective microprocessor) de la National Semi Conductor, que aunque no tiene la potencia de proceso de un 8080, 6800, ó un 2650, es adecuada en aplicaciones en que se busque la economía de componentes.

El LP8000 de General Instruments, el MCP 1600 de Western Digital es un diseño microprogramado de dos pastillas de circuito.

El 6100 de Intersil de un micro C.MOS de 12 bits, ha sido diseñado específicamente para emular el Minicomputador PDP8 de Digital Equipment.

Se han mencionado fabricantes y microprocesadores (u.P), en todos los casos se refieren a u.P de 8 bits.

Volviendo a nuestro caso el 6802 podemos decir que la diferencia con el 6800 es de que además de incluir el clock (reloj) tiene una RAM adicional de 128 bytes.

#### 1.5.1.1 Arquitectura

Como se dijo el 6802 es un Microprocesador (u.P) que contiene todos los registros y acumuladores del 6800 más el reloj (clock); dos líneas son usadas en este último uP para proveer las fases  $\phi_1$  y  $\phi_2$  del reloj.

El Software es completamente compatible entre estos dos Microprocesadores. Además el 6802 tiene un RAM de 128 bytes. En el apéndice se da la tabla de construcciones para el uP 6800 que es la misma para el 6802

En la figura I.15 se muestra la arquitectura del Microprocesador en ~~mención~~.

A Acumulador "A"

B Acumulador "B"

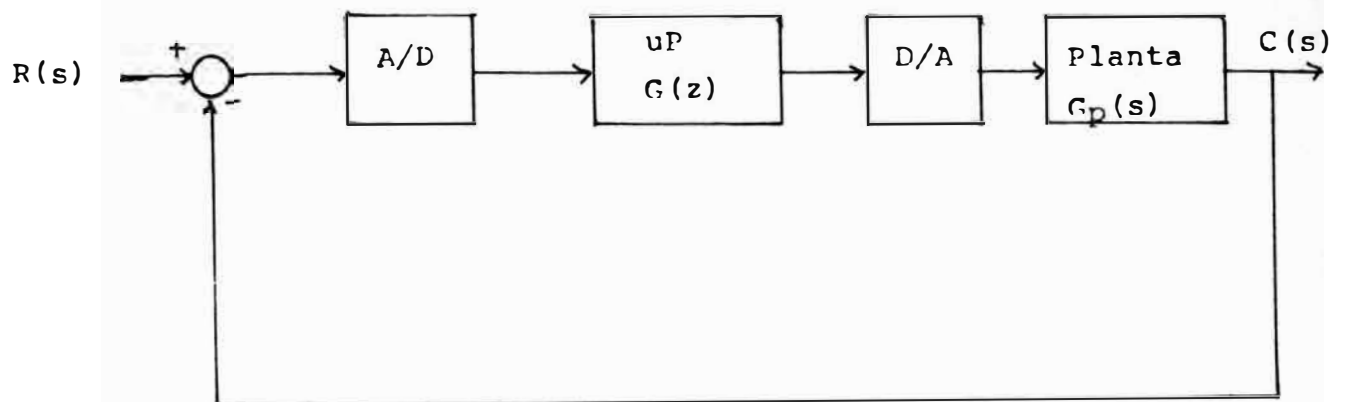


Fig. (1.14)

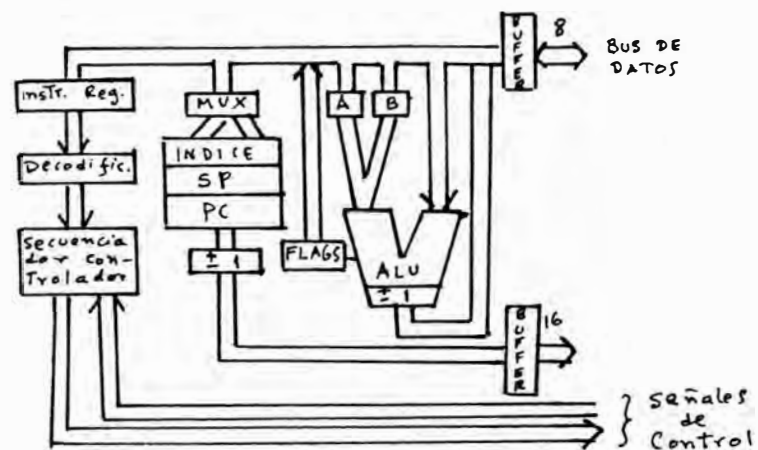


Fig. (1.15)

FLAG: Registro de condición de códigos

---

I I H I N Z V C

---

C bit de carry

V bit indicador de overflow

Z bit indicador de cero

N bit indicador de negativo

I bit de interrupciones

H bit indicador de medio carry

S.P.: Puntero de stack (es el indicador de la dirección de una pila)

P.C.: Contador de programa (es el indicador de la dirección eje cut + 1)

I.X.: Registro índice es un registro especial el cual se usa con el modo de direccionamiento indexado.

En cuanto a los buses tenemos:

(1) 16 líneas de dirección (bus de direcciones)

(2) 8 líneas de datos (bus de datos)

(3) 9 líneas de control.

(a) Enable.- Esta señal es del sistema de reloj. Un cristas de 3.57945 MHz es usado por el procesador generados 894.8 KHz del sistema de reloj (clock)

(b) R/ $\bar{W}$  - Read/write.- Es el control de lectura escritura su estado de lógica determina la dirección del dato (entra o sale) a el chip seleccionado.

Cuando es alto, la dirección del dato es hacia el up (lectura), cuando es bajo, la dirección de dato sale del microprocesador (u.p) escritura.

- (c) V.M.A. dirección de memoria valida.- Cuando es baja ("0") esta señal indica que la dirección sobre el bus de direcciones es invalida.
- (d) M.R. - Memory Ready.- Este control puede causar a la señal E (clock) se haga "stretched". Cuando MR es alto ("1") E podría estar en la operación normal, si MR es bajo ("b") E puede ser dividido en un múltiplo de medios períodos, que proporciona interfase con dispositivos lentos.
- (e) Reset esta línea es usada para pasar y empezar el microprocesador (u.p)
- (f) B.A. Cuando esta señal está activa, u.p. 6802 es parado y el bus de direcciones está disponible para dispositivos externos (pero no tristate)
- (g) Hult.- Cuando esta señal está activa "0", toda la actividad del u.p 6802 se suspende.

Las siguientes líneas de control:

- (h) IRQ
- (i) NMI

Van a ser tratados con extensión en la sección 1.5.1.4, debido a que se trata de pines usados para el tratamiento de las interrupciones.



Como se puede observar, tenemos 16 líneas de direcciones, más ocho de datos, 9 de control, serían 33 pines del Microprocesador (u.p. 6802). Además el pin 8 y el 35 para +Vcc; los pines 38 y 39 que están conectados al cristal; el pin 36 es para conectar Vcc a la memoria RAM Interna; y los pines 1 y 21 se conectan a tierra con lo que se completan los 40 pines del Microprocesador 6802.

Por considerarlo de interés dibujaremos un diagrama de flujo y las secuencias lógicas a seguir en un programa en lenguaje de máquina (Fig. I.16). Los pasos son los siguientes:

- 1.- Se trae la instrucción a ejecutarse desde la memoria al registro de instrucción M P.C. IR
- 2.- El PC se incrementa para que apunte a la próxima instrucción a ejecutarse:  $PC \longrightarrow PC + 1$
- 3.- Determinar el tipo de instrucción recién traída.
- 4.- Ver si la instrucción necesita datos de la memoria, traerlos a los acumuladores en caso afirmativo después de ubicar la dirección de este dato en memoria.
- 5.- Ejecutar la instrucción.
- 6.- Almacenar los resultados en un lugar apropiado.
- 7.- Ir al paso 1 y ejecutar la nueva instrucción.

El registro de instrucción es el mismo que se dibuja en la Fig. I.15 en la arquitectura del u.p. 6802.

#### 1.5.1.2 Componentes del Sistema Micro computador MEK 6802 D5E.

El Microprocesador para ser de utilidad requiere de memoria adicional y de dispositivos de entrada y salida E/S lo que convierte al conjunto en un Microcomputador.

Los componentes que usa este Micro computador son todos de la familia 68, Motorola.

Tenemos en primer lugar una memoria RAM. RAM 6810 es un chip de 24 pines, 6 de selección del chip de  $R/\bar{w}$ , 7 de dirección que nos pueden direccionar hasta 128 bytes, 8 de datos, 2 a fuente Vcc y 1 a tierra que suman 24 pines.

En el sistema este RAM es usado de la dirección E400 a E47F, que contiene 128 bytes de FLAG, datos y Stack para el sistema operativo del MEK 6802D5E, que está en una ROM de 2 k byte que veremos a continuación.

ROM 68131GE.- Como vimos contiene el sistema operativo del Sistema Microcomputador.

El Chip consta de 24 pines, 11 de dirección que nos direcciona 2 K bytes, 8 de datos, 3 de selección del Chip, una a fuente y otra a tierra.

Posee el sistema además dos PIA (Pheriferial Interface Adapter), uno del sistema para la interface con el teclado (hexadecimal), display y con una grabadora de audio, el otro es para el usuario.

Veremos en detalle este periférico programable en la sección 1.5.1.3. Además, el sistema posee 6 MC displays (diodos de 7 segmentos) y también dos memoris RAM 2114, los dos nos dan 1 K Byte debido a que se trata de una RAM estática de 1 K x 4 bits, tenemos también puertas lógicas 74500.

El 74LS244 que es un buffer tri - state octal (ocho en un chip); el decodificador 74156 que es un DUAL 1 de 4 de codificado y de multiplicador Open Colector (Colector Abierto).

Transistores que son drivers (excitadores) de cátodos. Como lo adelantamos un teclado hexadecimal para realizar la programación en lenguaje de máquina.

En el anexo se dan las hojas de datos de los componentes del sistema micro computador.

Se adjunta además en esta sección los planos del MEK 6802P5E.

### 1.5.1.3 Manejo de Interrupciones para el u.p 6802

Las interrupciones pueden ser clasificadas como de Hardware y Software. Las interrupciones de Hardware son  $\overline{\text{IRQ}}$  y  $\overline{\text{NMI}}$ .

Los de Software SWI y WAI la secuencia de funcionamiento para la interrupción  $\overline{\text{IRQ}}$  es como sigue:

- (1) El bit N°4 del registro de código (CCR) es cero ("0") y la línea  $\overline{\text{IRQ}}$  también va al estado activo ("0") al final de un ciclo de  $\phi_2$ , la secuencia  $\overline{\text{IRQ}}$  es enterrada.
- (2) Después que el C.P.U. ha completado una instrucción corriente; los registros PC, X, A, B, y CC son generados en localizaciones de memoria indicados por el stack pointer en forma descendente (7 Bytes en total).
- (3) El bit I del registro CCR es puesta a "1".
- (4) El dato en FFF8 es puesto en los ocho bits más significativos del PC.
- (5) El Dato en FFF9 es cargado en la parte de los ocho bits menos significativos de PC.
- (6) El contenido de PC sale del bus de direcciones durante  $\phi_1$ .

- (7) El contenido de la dirección de memoria contenida en PC va al registro de instrucción y es decodificado durante  $\phi_2$ .
- (8) Si son necesarios más de un byte de instrucción, los bytes adicionales entran al MPU para su ejecución, sino se va al siguiente paso.
- (9) Después la ejecución (7) es repetida por las siguientes instrucciones. El Loop es repetido hasta encontrar la instrucción RTI y ejecutado.

La secuencia de la interrupción

$\overline{\text{NMI}}$  es como sigue:

- (1)  $\overline{\text{NMI}}$  expone en el nivel activo ("0") el último ciclo de  $\phi_2$ , el C.P.U. completa la última instrucción del programa.
- (2) PC, X, A, B y CCR son generados en la pila.
- (3) El bit I del CCR es puesto a 1
- (4) Datos del FFF2 son cargados en PCH
- (5) Datos de FFFD son cargados en PCL
- (6) El contenido de PC sale del bus de direcciones durante  $\phi_1$
- (7) El contenido de la dirección localizada entra al registro de instrucción durante  $\phi_2$  y es decodificada como la primera instrucción de subrutina NMI.

- (8) Si hay más de un byte de instrucción, adicionales bytes entran a la CPU para su ejecución. Sino se va al paso (9)
- (9) Después de la ejecución, el paso 7 es repetido para las siguientes instrucciones hasta encontrar la instrucción RTI y esta se ejecute.

La secuencia de la interrupción de Software SWI es como sigue:

- (1) PC, X, A, B, CCR son guardados en la pila.
- (2) El bit I de CCR es puesto a "1"
- (3) El dato de FFFA va a PCH
- (4) El dato de FFFB va a PCL
- (5) El contenido de PC va fuera del bus de direcciones durante  $\phi_1$
- (6) El contenido del byte direccionado entra al registro de instrucción durante  $\phi_2$  y es decodificado como la primera instrucción de la subrutina de SWI.
- (7) Si hay más de un byte de instrucción, bytes adicionales entran al CPU para su ejecución.
- (8) Después de la ejecución el paso (6) es repetido, el Loop es repetido hasta encontrar la instrucción RTI es ejecutada.

El Microprocesador 6802 posee además un pin HALT que cuando es cero ("0") detiene toda - operación del C.P.U. Este PIN es muy usado para operaciones de debugging donde debugging es el depurado del programa instrucción por instrucción, ésta sería una interrupción del CPU por Hardware.

La otra interrupción de Software es la W.A.I. (wait a interrupt) que pone en STAND-BY la CPU en espera de una interrupción. La figura I.17 nos muestra un diagrama de flujo de interrupciones para el U.P. 6802.

#### 1.5.1.4 PIA (6821) Periférico Programable

El PIA 6821 es un circuito integrado, que se usa para la comunicación en paralelo, entre el Microprocesador y una unidad E/S (entrada salida) como puede ser un teclado, una lectora de tarjetas, una impresora, etc.

En realidad el primer adaptador periférico programable usado por la Motorola fue el 6820, que es pin a pin compatible con el 6821.

La comunicación con el Microprocesador es por medio del bus de datos; el PIA tiene dos puertos (A y B) que consta de ocho bits bidireccionales que están en buses de datos para la interconexión con el mundo exterior.

Estas 16 líneas bidireccionales pueden ser programadas para actuar como líneas de entrada o salida.

Si consideramos que cada bit o línea de estos dos puertos, viene a ser un pin allí tendríamos 16 pines.

Además hay 24 pines ya que el 6821 tiene 40 pines, de los cuales 8 son para la comunicación con el u.p. tenemos 2 líneas de control para cada puerta CA y CB, son líneas que llegan al PIA, CA<sub>2</sub> y CB<sub>2</sub>, son líneas bidireccionales (llegan y salen).

Un pin para  $\overline{\text{Reset}}$ , uno para IRQR, uno para IRQB, otro para R/ $\overline{w}$ , otro para ENABLE (reloj,  $\phi_2$ ), dos para la selección de registros (RS<sub>1</sub>, RS<sub>0</sub>) tres para la selección del circuito (CS<sub>0</sub>, CS<sub>1</sub>, CS<sub>2</sub>) uno para fuente (Vcc), otro para tierra (Vcs); que hacen un total de 40 pines.

El esquema del 6821 se muestra en la figura I.18.

Lo más novedoso y que para un diseñador de hardware puede ser incluso una herejía, es que como se dijo anteriormente las líneas de los puertos pueden ser programadas individualmente. Como entrada o salida; un "0" en el registro de control de E/S define una entrada y un "1" una salida.



Como se ve en la figura, el 6821 tiene dos líneas de control por puerto, CA1 y CB1 son fijas como entrada; mientras la segunda línea de control CA2 y CB2 puede programarse como entrada o como salida.

El circuito lleva además dos líneas de demanda de interrupción, IRQA, IRQB, una por puerto.

La razón de esto es que, en un sistema sencillo las dos demandas de interrupción A y B se unirán a la misma línea de interrupción. Sin embargo, en un sistema más complicado puede usarse una estructura de interrupciones más elaborado, con prioridades y posiblemente interrupciones vectorizadas.

El grupo de líneas de selección de circuito ( $CS_0$ ,  $CS_1$ ,  $CS_2$ ) responde, al igual que el grupo de selección de registros ( $RS_0$ ,  $RS_1$ ) a una filosofía propia de Motorola. La mayoría de componentes de la familia 6800 tienen tres terminales de selección de circuito o más. Esto en un sistema pequeño permite el ahorro de decodificadores. Sin embargo, esto requiere de más terminales de pastilla, esto motiva que la selección de registro se haga con solo dos líneas ( $RS_0$ ,  $RS_1$ ).

Como se ve en la figura anterior el 6821 tiene seis registros internos y solo dos señales de

RS; se sabe que con dos señales se pueden direccionar uno de 4 registros.

Esto es lo que hace el PIA, el registro de interfase periférico y el registro de control de E/S de cada puerto tienen la misma dirección, la selección de uno u otro registro se hace mediante un bit del registro de control. Un cero en la posición segunda del bit de registro de control selecciona el registro de control de E/S y un "1" el registro de interfase periférico. El fabricante presenta esto como una ventaja, ya que al inicializar el sistema, o cuando se aplica una señal de reposición en general, el contenido del registro de control pasa a cero, seleccionando automáticamente. El registro del control de E/S esto hace que no se pueda producir señal espuria alguna en el registro de interfase, ya que este no ha sido seleccionado. En su contra puede decirse que esto lleva a una programación un tanto tediosa, pues si hay que acceder secuencialmente al registro de control de E/S y al de interfase tendremos que pasar forzosamente por una maniobra intermedia adicional de acceso al registro de control.

## CAPITULO II

### CONSIDERACIONES PARA EL USO DE MICROPROCESADORES EN SISTEMAS DE CONTROL

#### Introducción

El diseño en todas las ramas de la ingeniería tiene dos grandes componentes que vienen a ser la parte técnica y la económica. Y nuestro caso no es una excepción; en este capítulo se tomaron en cuenta las consideraciones de orden técnico solamente, debido a que como se señala el objetivo de este trabajo es el de introducir un controlador digital en tiempo real, usando un microcomputador.

Quizás la más grande contribución de los microprocesadores en el Control Digital Directo (D.D.C.) sea el ahorro significativo de Hardware, comparado con el que se tendría de haber usado lógica cableada. A cambio de complejidad en el software.

#### 2.1 Consideraciones Preliminares en el Diseño

Globalmente se puede definir un sistema claramente por sus especificaciones funcionales. Aspectos adicionales a ser considerados para un inicio son los requerimientos realizables y satisfactorios, número de unidades, adaptabilidad a las especificaciones del usuario.

Comunicaciones a Operadores y/o otros computadores,

desarrollos de Hardware y Software y capacidades de complementación, servicio, segundos fuentes, etc.

Mucho costo adicional de tiempo y dinero puede ocasionar el seleccionar inicialmente un sistema demasiado pequeño.

De allí, que los requerimientos funcionales deben ser evaluados cuidadosamente en esta fase.

Se podrían considerar 3 aspectos como dominantes:

- a) Requerimientos de proceso de datos para propósitos de control puede ser expresado como el número de operaciones de punto flotante que pueden ser ejecutados por segundo.
- b) Requerimientos de Comunicaciones puede ser expresado como el número de canales a ser monitomeados por consulta sobre diferentes prioridades.
- c) Requerimientos de Realización generalmente puede ser prescritos por los efectos de fallas en el sistema.

A continuación, se expresan estos tres requerimientos en un sistema de tres coordenadas, Fig. II.1.

Donde:

- (a) es un regulador de calor
- (b) control distribuido en una planta química
- (c) sistema de aterrizaje de avión

- (d) turbina hidráulica de control
- (e) control de posición de máquinas herramientas.

El gráfico nos muestra que el sistema (a) puede ser cubierto con un microprocesador de 8 bits.

- (d) debe ser cubierto con un sistema de 16 bits con una microprogramación aritmética.

(c), (b) y (e) requieren arreglos multiprocesadores.

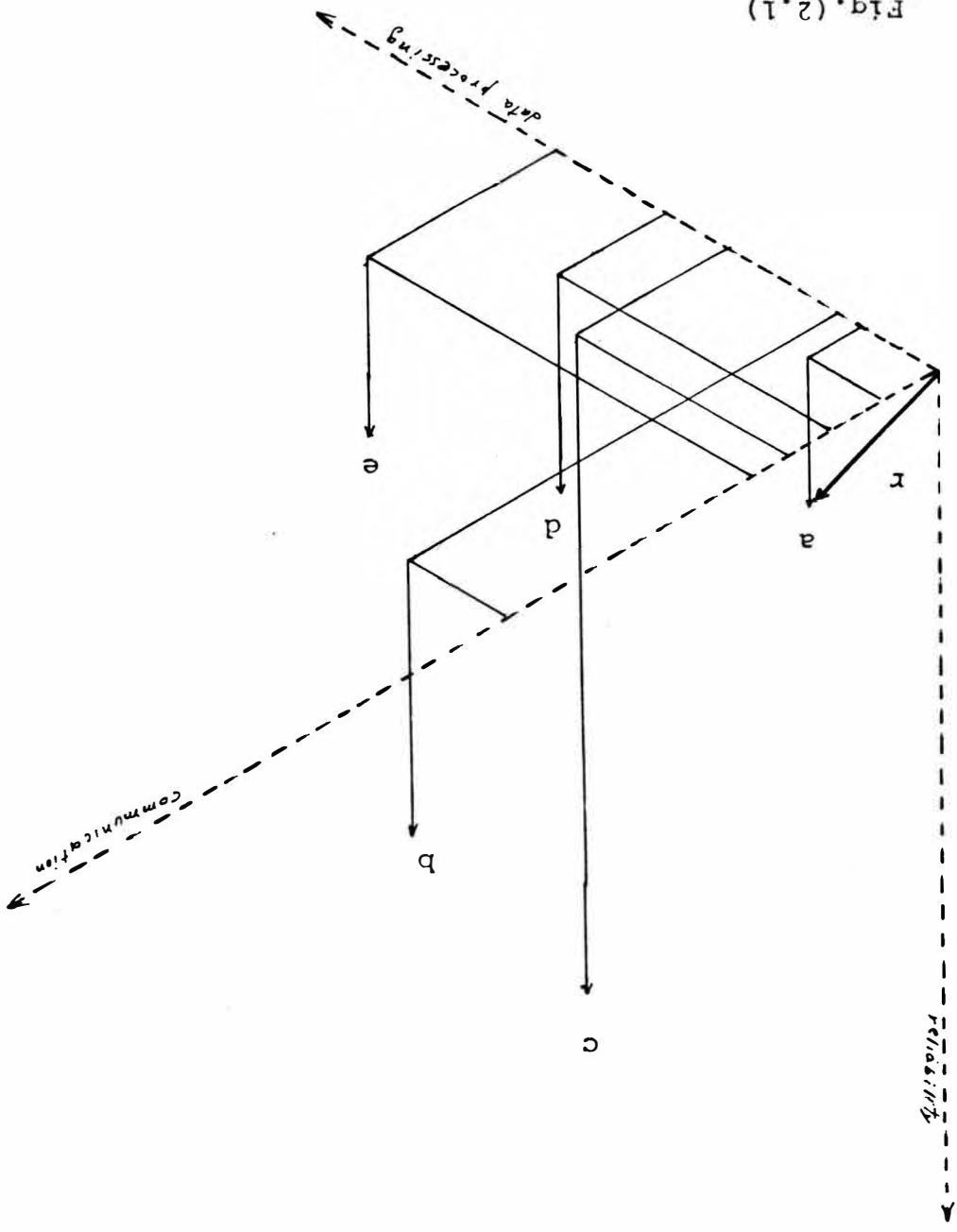
La distancia  $r$  desde el origen es una medida de la complejidad funcional e indica que solución de software y hardware es la requerida.

Como en todos los diseños, las iteraciones siempre son requeridas.

## 2.2 Información de la Arquitectura y Estructura del Sistema

La performance de los microprocesadores puede variar según la elección de la Arquitectura del mismo, en el Capítulo I vimos que existen muchos fabricantes de Microprocesadores, luego en la fase inicial se debe contar con información acerca de las arquitecturas y demás detalles técnicos de varios u.p para realizar luego la elección del microprocesador a usar. Existen tanto programas como Microprocesadores desarrollados para propósitos específicos de Control. Que son vendidos como controladores industriales para realizar trabajos específicos.

Fig. (2.1)



### 2.3 Algoritmos de Control

Los algoritmos de control son varios para el caso de sistemas de control digital.

En el caso de uso de Microprocesadores la arquitectura y el Microprocesador de que se dispone es fundamental para elegir el algoritmo a utilizar.

Es lógico que en el Control Digital Directo se intentan superar los inconvenientes de la regulación PID analógica, tales como iteración entre los ctes. PID, método de determinación y ajuste óptimo de parámetros, compromiso entre ajuste para variaciones de carga y punto de referencia. Como se verá luego la relación  $T/T_n$  y  $T_v/T$  donde  $T$  es el tiempo de muestreo, es adimensional. Por lo tanto, dependen solo de la dificultad del proceso a controlar y el margen de variación es mucho menor que con los reguladores analógicos, donde  $T_n$  y  $T_v$  tienen dimensión de tiempo, ya que el período  $T$  se determina en selección de la cte. de tiempo principal del sistema o proceso.

Aunque no se cite explícitamente, la ganancia integral y derivativa deben limitarse para las frecuencias bajas y altas respectivamente, como ocurre en los reguladores analógicos.

#### 2.3.1 Tipos de Algoritmos

1.- Posición absoluta: Normalmente se emplea el algoritmo PI. El término derivativo va perdiendo efecto cuando

el intervalo de muestreo va aumentando si se desea una independencia entre las constantes PID puede emplearse.

$$(1) \quad Y_n = K_R E_n + \frac{K_T}{T_n} \sum_{K=1}^n E_K + \frac{K_{TV}}{T} (E_n - E_{n-1}) + Y_m$$

2.- Posición Incremental. Quizás sea el más empleado. Debido a que permite un sistema sencillo de transferencia operador/ordenador. Requiere un actuador lineal y repetitivo. Puesto que al calcular la segunda derivada del error, debe filtrarse para no incorporar demasiado ruido.

$$(2) \quad \bar{E}_n = \frac{1}{4} (E_n + E_{n-1} + E_{n-2} + E_{n-3})$$

La mejora aportada por la segunda derivada decrece rápidamente al aumentar el período de muestreo, siendo despreciable cuando éste es aproximadamente igual a la cte. de tiempo principal del proceso.

3.- Velocidad. Normalmente la velocidad de actuación no puede tener un valor cualquiera sino que debe escogerse de entre una serie de valores permitidos, puesto que usando salida a impulsos y motores paso a paso se requiere un generador para cada velocidad permitida.

Si la salida es un nivel de tensión, se usan motores C.C. La señal se aplica durante todo el período de muestreo. Normalmente no se emplea el término derivativo por su reducida eficacia.



4.- Tiempo. Este método es más empleado que el anterior por una razón funcional y económica.

En general da un funcionamiento mejor y permite emplear motores de alterna y válvulas de velocidad cte. como actuadores. Normalmente no se emplea el término derivativo.

Aunque el tiempo de actuación puede ser cualquiera (entre 0 y T), en algunos casos se han usado solo determinales valores, lo que ha aumentado la calidad de la regulación.

#### 2.4 Software

El sistema de software puede cubrir los dos aspectos de comunicación y proceso de datos.

La comunicación requiere de mecanismos sincronizados, sistemas de operación en tiempo real pueden ser realizados por varios ordenadores instalados.

La programación de una aplicación de control de un proceso industrial continua tiene características importantes debido a varias causas tales como:

- El ordenador es un instrumento más de control del proceso, lo que significa que:

- El proceso es mucho más importante que el ordenador (y el control más que la programación)

- . El operador es un instrumentista

En la planta no se dispone de varios programadores buenos y experimentados.

- . La comunicación operador/ordenador debe ser similar a la de operador cuadros de mando.
- . Si la aplicación es de tiempo real; los fallos de programación pueden resultar muy caros.

Las aplicaciones no son genéricas sino relativamente similares. Lo que permite una especialización del sistema de programación.

Los requerimientos del usuario son:

- El sistema de programación debe:
  - . ser fácil de aprender y utilizar
  - . no requerir más programador que el ingeniero y técnico de Control.
  - . estar adaptado a las necesidades del control de procesos.
  - . ser eficiente: requerir poca memoria central y ser rápido en su exploración.
  - . tener un buen soporte de programas estándar y científicos.
  - . ser universal para poder intercambiar programas de aplicación entre distintos tipos de ordenador.
  - . permitir introducir (compilar) nuevos programas y modificar los existentes sin alterar las funciones usuales de control.
  - . Posibilitar la escritura eficiente de un sistema operativo.

Las características de la programación en un control de procesos se deben a dos fuentes: ordenador y proceso.

- Juego reducido de instrucciones
- Gran velocidad de ciclo base
- Periféricos lentos
- Memoria central bastante limitada.

Las exigencias del proceso en cuanto al sistema de programación son:

- Parte de la programación debe hacerse desde la consola del operador, con entradas convencionales en instrumentación.
- Un programa debe empezar a ser ejecutado por cualquiera de los siguientes causas:
  - a) Tiempo : absoluto o transcurrido desde un incidente
  - b) Interrupción: Incidente en el proceso, requerimiento del operador, llamada desde otro programa.
- Gran rapidez de respuesta a ciertas interrupciones.
- Tratamiento de los errores de forma que combine continuidad de ejecución/seguridad de la mejor forma.
- El grado de dependencia de un programa respecto al proceso depende del nivel en que esté situado el programa.
- Deben controlarse en buen número de bucles a una frecuencia elevada. La figura II.2 muestra la estructura general del Software de control de procesos.

### 2.4.1 Lenguajes

Los lenguajes empleados pueden ser los de alto nivel y bajo nivel. Alto nivel como FORTRAN, Pascal, Basic, PL/1, etc.; bajo nivel como Assembler y lenguaje de máquina.

Las ventajas de la programación en bajo nivel son: buen aprovechamiento de la máquina, ocupa poca memoria, ejecución rápida, difícil de aprender. Adecuados para operaciones de entrada/salida, mientras que sus desventajas son Costo de programación elevado, dependencia absoluta del tipo de máquina, difíciles de depurar, es difícil introducir cambios, poco adecuado para cálculos algebraicos. Normalmente, ampliados con Macros para el cálculo de control de procesos.

Costo de programación normal, fácil de aprender, adecuados para cálculos algebraicos.

Las desventajas son: Mayor ocupación de memoria y menor velocidad de ejecución que en bajo nivel, dependen parcialmente del tipo de máquina, inadecuados para cálculos lógicos. Inadecuados para operaciones de entrada/salida. Como observación se puede decir que normalmente son ampliados para adaptarlos mejor a control de procesos en tiempo real.

Lenguajes de tiempo real como Pearl (Werum, 1978); Portal (Lienhard, 1978) o Modula (1976) han sido diseñados para el uso en tiempo real y programaciones pa-

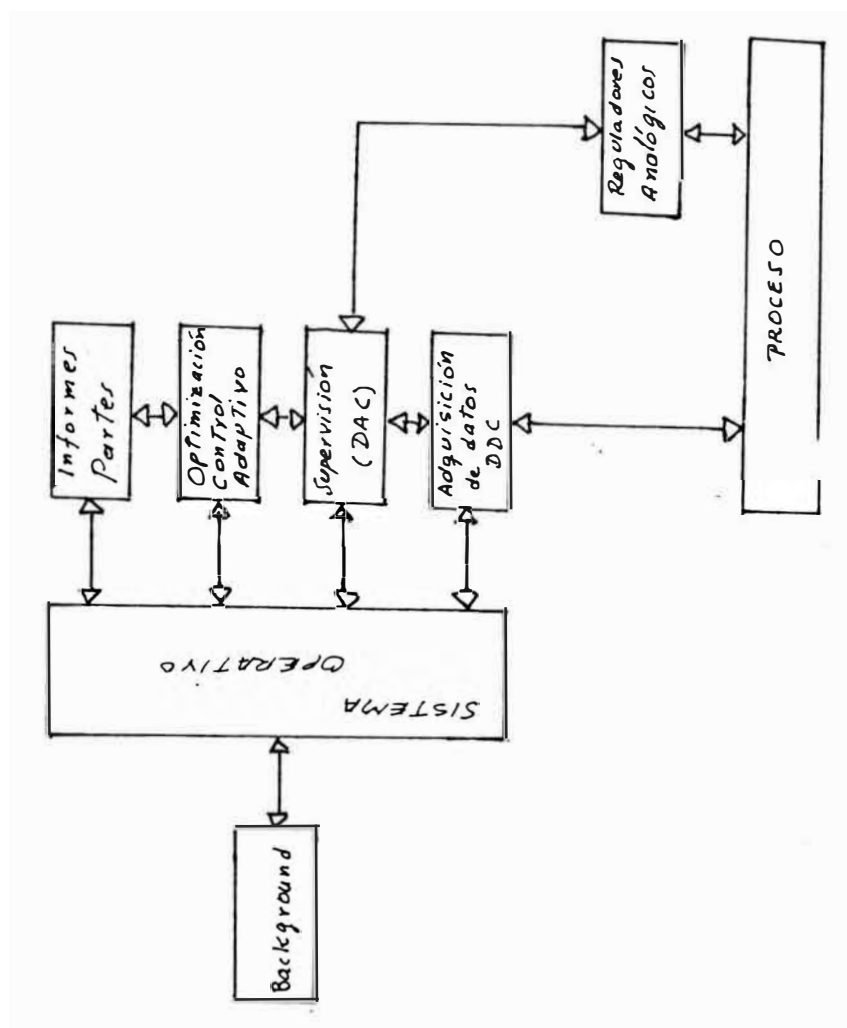


Fig. (2.2)

ralelas.

Pero en los actuales momentos, estos lenguajes están disponibles solo para un restringido rango de procesadores u ordenadores.

Los Macro librerías para propósitos de control mejoran el desarrollo del proceso al nivel del lenguaje Assembler.

Dos aspectos que tienen que ser tomados en cuenta por el diseñador son, los retardos de tiempo y los efectos de la cuantificación; esta última debida a la longitud de palabra en nuestro caso (8 bits).

#### 2.4.2 Efectos de la longitud de palabra finita sobre la controlabilidad y la localización de los polos en lazo cerrado

Debido a la limitación en la longitud de palabra, en general, las señales en y fuera del microprocesadores son truncados y cuantificadas; luego es imprescindible ver los efectos en la performanaje del sistema cuando los parámetros del sistema pueden ser realizados solo por un número finito de operaciones.

Los parámetros de diseño son la ganancia de estados realimentados o la ganancia a la salida de la realimentación.

Si el sistema es completamente controlable, los polos del sistema en lazo cerrado pueden ser localiza

dos arbitrariamente.

Debido a la limitación en la longitud de palabra, los polos en lazo cerrado del sistema no tienen un dominio con infinita resolución.

Cuando un sistema de control digital está sujeta a una longitud de palabra finita y cuantización, éste no es controlable.

La limitación en la resolución de los parámetros del sistema es trasladado a problemas en la estabilidad y el error, los cuales son resultados directos de la cuantización. Los efectos de cuantización refiriéndonos sobre la entrada  $U(K)$  puede ser investigada.

$$(3) \quad X(K+1) = A X(K) + B u(K)$$

donde  $u(K)$  es la entrada escolar. Asumiendo que  $u(K)$  es restringida a los valores de cuantización, tendríamos:

$$(4) \quad u(K) = n_K q$$

$n_K = 0, \pm 1, \pm 2, \dots$  y que en el nivel de cuantificación la solución de (1) es:

$$(5) \quad \underline{X}(N) = A^N X(0) + \sum_{K=0}^{N-1} A^{N-K-1} B \underline{u}(K)$$

de (4) y (3)

$$X(N) = A^N X(0) + q \sum_{K=0}^{N-1} A^{N-K-1} B n_K$$

Como  $u(K)$  es limitado a los distintos niveles de  $n_K q$ , el estado final  $X(N)$  es parametrado por los enteros.

$n_K$ ,  $K = 0, 1, 2, \dots, N-1$ . Si el par  $(A, B)$  es controlable entonces  $X(0)$  puede ser cualquier punto  $X(N)$  en el espacio de estado solo si  $u(K)$  puede ser seleccionado desde valores continuos. Como  $u(K)$  es limitado a valores cuantizados, entonces  $X(N)$  es restringido.

#### 2.4.3 Tiempos de Retardo en Sistemas de Control con Microprocesadores

Los microprocesadores (u.p) son máquinas de cómputo relativamente lentas, en varias aplicaciones de cómputo digital; las cuales no corresponden a el manipuleo de datos en tiempo real, esta lentitud no es importante; el cómputo en tiempo real es usualmente necesario en aplicaciones de sistemas de control, luego el tiempo de retardo en la ejecución del manipuleo y ejecución de datos tiene un efecto significativo en la respuesta del sistema.

Si el tiempo de retardo es demasiado largo, podría no ser el tiempo suficiente para ejecutar todos los cálculos necesarios requeridos para ejecutar los algoritmos de control, otro efecto es lo adverso que es el tiempo de retardo en la estabilidad de los sistemas de control en lazo cerrado.

Con respecto al cálculo del algoritmo podría-



mos decir que en las subrutinas de multiplicación, suma y división debe considerarse el uso del punto flotante, lo cual requiere de un consumo de tiempo de máquina adicional y mayor empleo de memoria.

#### 2.4.4 Efectos de la cuantificación en el error de estado estable

En esta sección veremos el efecto que tiene la cuantificación en el error en el estado estable.

Las figuras II.3 y II.4 nos muestran un cuantificador. La línea punteada representa la característica de transferencia deseada, y la escalera es la salida del cuantificador, donde la entrada  $r(t)$  puede ser de cualquier forma o amplitud. Pero la salida  $y(t)$  puede tener solo niveles cuantizados vecinos al valor de  $r(t)$ .

Así de  $-q/2$  a  $q/2$  la salida es cero, de  $q/2$  a  $3q/2$  es  $q$  y así sucesivamente.

La relación entre la longitud de palabra y el nivel de cuantización es la sgte.

$$q = 2^{-N} F_s$$

donde  $N$  es el número de hilos de la palabra,  $F_s$  (Full escala). El error de cuantización es  $q/2$ .

$$q/2 = 2^{-N-1} F_s$$

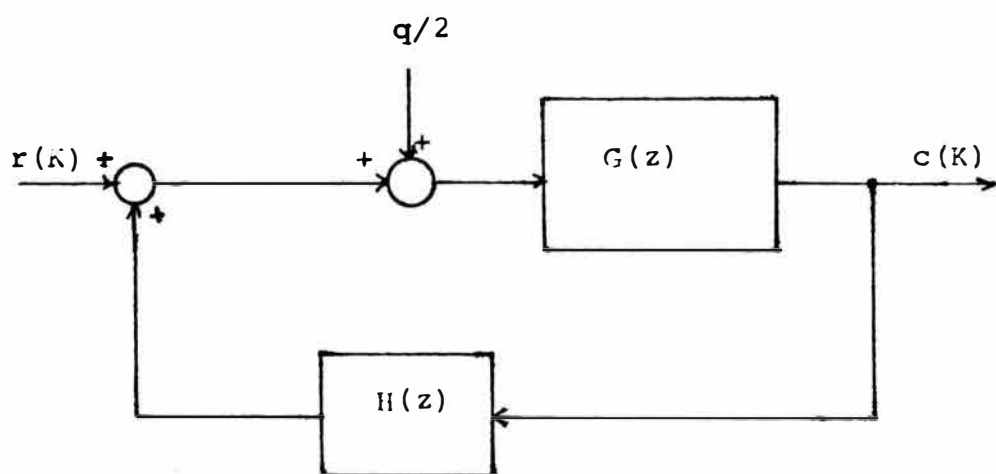
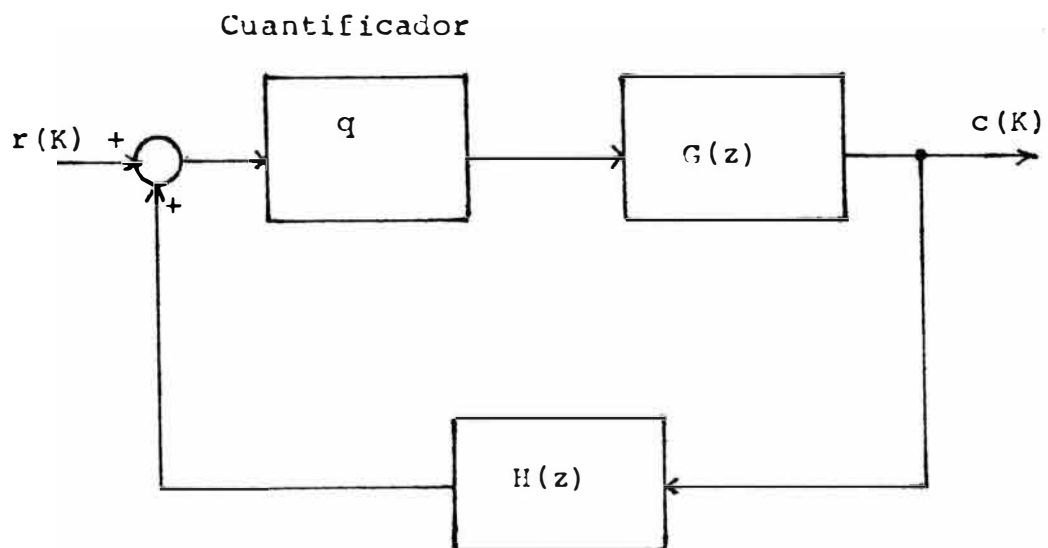


Fig. (2.4)

Como la figura nos muestra, el dispositivo es no lineal. Luego el análisis de los efectos de cuantización puede ser complejo analíticamente.

Los dos más comunes efectos de los niveles de cuantización en los sistemas de control de lazo cerrado son las oscilaciones sostenidas y el error en estado estable.

El efecto sobre el error en estado estable de la cuantización puede ser estudiado si el cuantificador se le reemplaza por una fuente de ruido de  $\pm q/2$

Como se muestra en la figura II.4 (B)

## 2.5 Hardware

Varios microprocesadores de 8 y 16 bits están disponibles en el mercado.

Los controladores digitales basados en microprocesadores requieren de elementos periféricos de interfase paralela en nuestro caso el PIA 6821 de Motorola.

La descripción y uso de este periférico se hizo en el primer capítulo.

Se requerirá además de interfases entre las Señales Digitales y Analógicas, Convertidores Analógicos Digitales y Digitales-Analógicos, A/D y D/A.

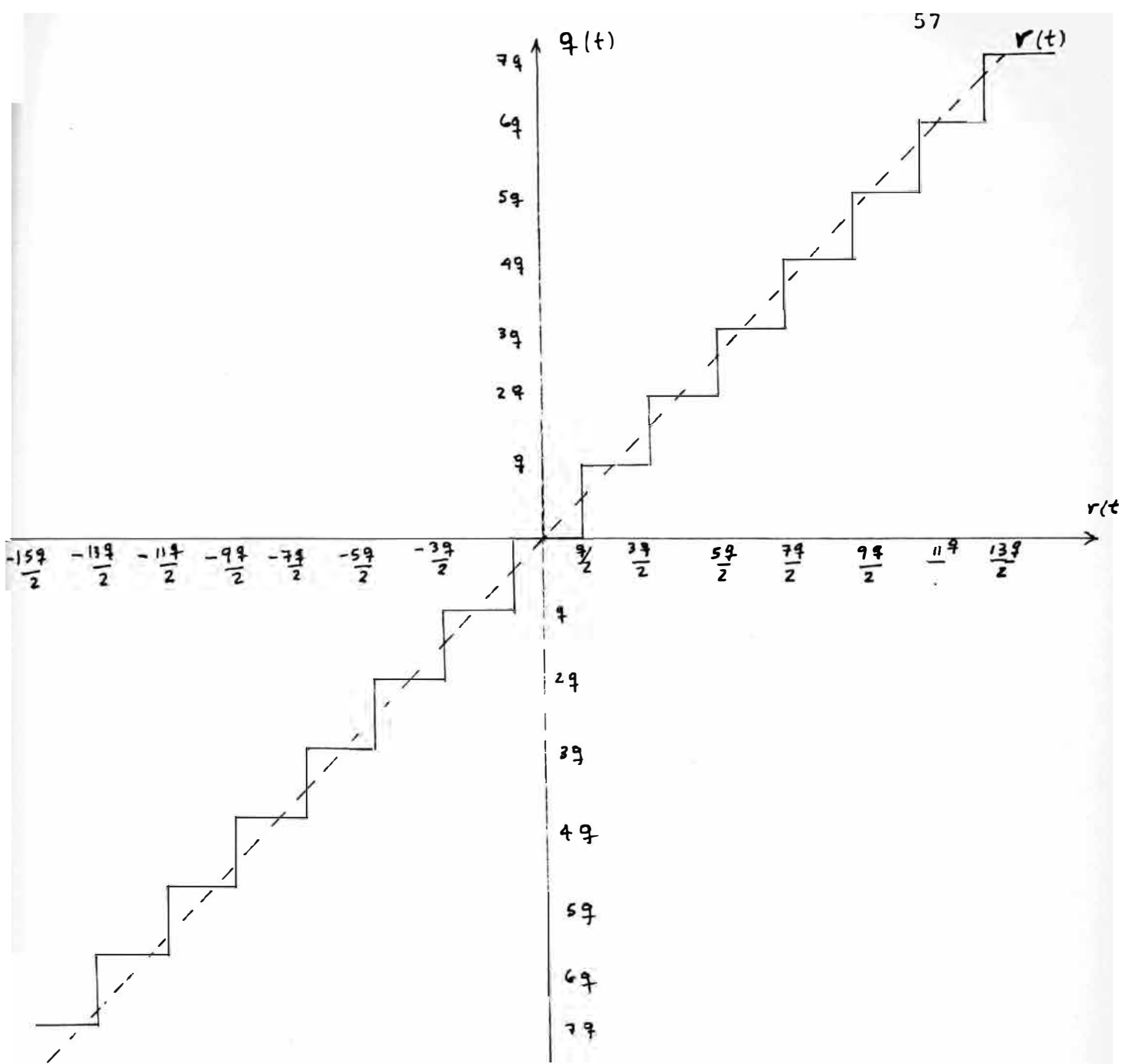


Fig. (2.4)



Fig. (2.3)

Las características de éstos y su uso se detallaron en el Capítulo IV.

Para la conversión analógica-digital se necesitará elegir una que responda a los requerimientos de diseño, sobre todo velocidad, ya que existen en el mercado, gran variedad para los 8 bits que necesitamos, así hay los que usan el método de conversión de la Rampla simple, doble rama, aproximaciones sucesivas con distintas velocidades.

Se necesitará también un sistema de interrupciones en casi todos los casos, este sistema se describe para el caso de el fabricante Motorola en el capítulo I.

## CAPITULO III

### DEDUCCION DEL ALGORITMO A IMPLEMENTAR

#### 3.1 Estructura del Algoritmo

La figura III.1 nos muestra el diagrama de bloques para un control digital directo (D.D.C.) usando micro - procesadores.

x (magnitud controlada)

e (señal de error)

z (perturbación)

w (entrada)

A/D ADC (Convertidor Analógico-Digital)

D/A DAC (Convertidor Digital-Analógico)

Y (fuerza de control)

El programa de regulación comprende un conjunto de funciones particulares (Módulos). Los cuales son programados como sub-programas.

Estos subprogramas son incluidos en un programa principal.

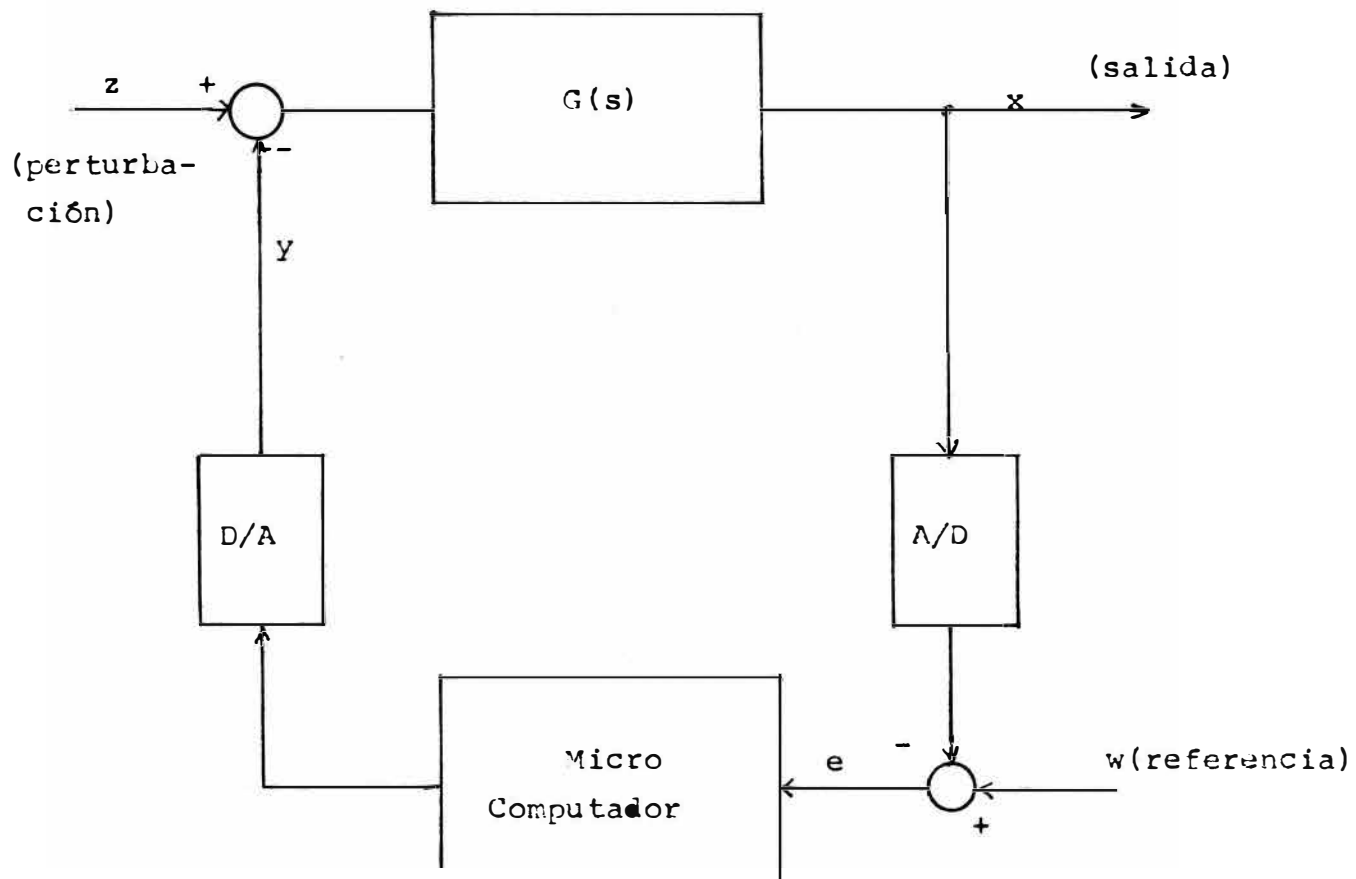


Fig. (3.1)

### 3.2 Algoritmo de Regulación

A pesar de que la rectificación con el Computador Digital no presenta casi ninguna dificultad o limitación especial en la elección del algoritmo de regulación. Nosotros nos referimos primero a dos conocidas aproximaciones discretas de la ecuación diferencial de un regulador PID.

Una viene a ser la aproximación rectangular y la otra la aproximación trapezoidal.

Esto viene a ser para nosotros el algoritmo PID, también vamos a ver el algoritmo PIDT, que es una variante del anterior y va a ser descrito en una sección aparte.

#### 3.2.1 Algoritmo PID

La ecuación diferencial de un regulador PID ideal puede ser escrito de la siguiente forma:

$$y(t) = K_R \left[ e(t) + \frac{1}{T_n} \int_0^t e(r) dr + T_v \frac{d}{dt} e(t) \right] \quad (1)$$

esta misma ecuación escrita en tiempo discreto

$$y_n = K_R e_n + K_R \frac{I}{T_n} \sum_{i=0}^n e_i + K_R \frac{T_v}{T} (e_n - e_{n-1}) \quad (2)$$

donde la integral se ha aproximado por la suma de rectángulos y la derivada por la ecuación diferencia.



si definimos:

$$Y_{I,n} = K_R \frac{I}{T_n} \sum_{i=0}^n e_i \quad (3)$$

$$Y_{I,n} = Y_{I,n-1} + K_R \frac{T}{T_n} e_n \quad (4)$$

$$Y_{I,n-1} = Y_{I,n} - K_R \frac{T}{T_n} e_n \quad (5)$$

$$Y_{n-1} = K_R e_{n-1} + Y_{I,n-1} + K_R \frac{Tv}{T} (e_{n-1} - e_{n-2}) \quad (6)$$

de (3) y (6)

$$Y_{n-1} = K_R e_{n-1} + Y_{I,n} - K_R \frac{T}{T_n} e_n + K_R \frac{Tv}{T} (e_{n-1} - e_{n-2}) \quad (7)$$

de (2) y (3)

$$Y_n = K_R e_n + Y(I,n) + K_R \frac{Tv}{T} (e_n - e_{n-1}) \quad (8)$$

de (7) se tiene

$$Y(I,n) = Y_{n-1} - K_R e_{n-1} + K_R \frac{T}{T_n} e_n - K_R \frac{Tv}{T} (e_{n-1} - e_{n-2}) \quad (9)$$

de (8) y (9) tenemos

$$Y_n = K_R e_n + Y_{n-1} - K_R e_{n-1} + K_R \frac{T}{T_n} e_n - K_R \frac{Tv}{T} (e_{n-1} - e_{n-2}) + K_R \frac{Tv}{T} (e_n - e_{n-1}) \quad (10)$$

$$Y_n = Y_{n-1} + K_R \left[ \left(1 + \frac{T}{T_n} + \frac{Tv}{T}\right) e_n + \left(-1 - 2 \frac{Tv}{T}\right) e_{n-1} + \frac{Tv}{T} e_{n-2} \right] \quad (11)$$

$$Y_n = Y_{n-1} + d_0 e_n + d_1 e_{n-1} + d_2 e_{n-2} \quad (12)$$

donde:

$$d_0 = K_R \left(1 + \frac{T}{T_n} + \frac{Tv}{T}\right) \quad d_1 = K_R \left(-1 - 2 \frac{Tv}{T}\right)$$

$$d_2 = K_R \frac{Tv}{T}$$

si la integral la aproximamos por suma de trapecios en vez de rectángulos.

$$Y_{I,n} = K_R \frac{T}{T_n} \sum_{i=0}^n \frac{e_i + e_{i-1}}{2} \quad (13)$$

$$Y_{I,n} = Y_{I,n-1} + K_R \frac{T}{T_n} \left(\frac{e_n + e_{n-1}}{2}\right) \quad (14)$$

$$Y_{n-1} = K_R e_{n-1} + Y_{I,n-1} + K_R \frac{Tv}{T} (e_{n-1} - e_{n-2}) \quad (15)$$

$$Y_{I,n-1} = Y_{I,n} - K_R \frac{T}{T_n} \left(\frac{e_n + e_{n-1}}{2}\right) \quad (16)$$

$$Y_{n-1} = K_R e_{n-1} + Y_{I,n} - K_R \frac{T}{T_n} \left(\frac{e_n + e_{n-1}}{2}\right) + K_R \frac{Tv}{T} (e_{n-1} - e_{n-2}) \quad (17)$$

$$Y_{I,n} = Y_{n-1} - K_R e_{n-1} + K_R \frac{T}{T_n} \left(\frac{e_n + e_{n-1}}{2}\right) - K_R \frac{Tv}{T} (e_{n-1} - e_{n-2}) \quad (18)$$

$$\begin{aligned}
Y_n = & K_R e_n + Y_{n-1} - K_R e_{n-1} + K_R \frac{T}{2T_n} e_n + K_R \frac{T}{2T_n} e_{n-1} - \\
& - K_R \frac{Tv}{T} e_{n-1} + K_R \frac{Tv}{T} e_{n-2} + K_R \frac{Tv}{T} e_n - \\
& - K_R \frac{Tv}{T} e_{n-1} \quad (19)
\end{aligned}$$

$$\begin{aligned}
Y_n = & y_{n-1} + K_R \left(1 + \frac{T}{2T_n} + \frac{Tv}{T}\right) e_n + K_R \left(-1 + \frac{T}{2T_n} - \frac{2Tv}{T}\right) \\
& e_{n-1} + K_R \frac{Tv}{T} e_{n-2} \quad (20)
\end{aligned}$$

$$Y_n = Y_{n-1} + d_0 e_n + d_1 e_{n-1} + d_2 e_{n-2} \quad (21) *$$

$$d_0 = K_R \left(1 + \frac{T}{2T_n} + \frac{Tv}{T}\right); \quad d_1 = K_R \left(-1 + \frac{T}{2T_n} - \frac{2Tv}{T}\right)$$

$$d_2 = K_R \frac{Tv}{T}$$

Para la programación de microcomputadoras se hace un cálculo intermedio

$$Z_w = Y_{n-1} + d_1 e_{n-1} + d_1 e_{n-1} + d_2 e_{n-2} \quad (22)$$

para luego calcular  $y_n$

$$y_n = d_0 e_n + Z_w \quad (23)$$

donde  $Z_w$  es el valor en el tiempo entre minutos exploratorios, después de dicha exploración está el valor de  $Y_n$  por cero.

Como referencia daremos en la próxima sección la deducción del algoritmo  $PIDT_1$

### 3.2.2 Algoritmo $PIDT_1$

La ecuación diferencial del algoritmo sería:

$$T_D \dot{y}(t) + y(t) = K_R \left[ e(t) + \frac{1}{T_n} \int_0^t e(r) dr + T_v \frac{d}{dt} e(t) \right] \quad (24)$$

Escrito en tiempo discreto tendríamos:

$$Y_n + \frac{T_D}{T} (Y_n - Y_{n-1}) = K_R \left[ e_n + \frac{T}{T_n} \sum_{i=0}^n e_i + \frac{T_v}{T} (e_n - e_{n-1}) \right] \quad (25)$$

si hacemos

$$Y'_n = Y_n + \frac{T_D}{T} (Y_n - Y_{n-1}) \quad (26)$$

tendríamos:

$$Y'_n = K_R \left[ e_n + \frac{T}{T_n} \sum_{i=0}^n e_i + \frac{T_v}{T} (e_n - e_{n-1}) \right] \quad (27)$$

que es el algoritmo PID, luego por lo ya deducido podemos escribir directamente

$$Y'_n = Y'_{n-1} + K_R \left[ \left(1 + \frac{T}{T_n} + \frac{T_v}{T}\right) e_n + \left(-1 - \frac{2T_v}{T}\right) e_{n-1} + \frac{T_v}{T} e_{n-2} \right] \quad (28)$$

Luego por definición tenemos:

$$Y'_n = Y_n + \frac{T_D}{T} (Y_n - Y_{n-1})$$

$$Y'_n = Y_{n-1} + \frac{T_D}{T} (Y_{n-1} - Y_{n-2})$$

Estas dos últimas ecuaciones escritas en (5) dan como resultado:

$$Y_n \left(1 + \frac{T_D}{T}\right) = \left(1 + 2\frac{T_D}{T}\right) Y_{n-1} - \frac{T_D}{T} Y_{n-2} + K_R \left(1 + \frac{T}{T_n} + \frac{Tv}{T}\right) e_n + (-1 - 2\frac{Tv}{T}) e_{n-1} + \frac{Tv}{T} e_{n-2}$$

Luego se puede escribir:

$$Y_n = C_1 Y_{n-1} + C_2 Y_{n-2} + d_0 e_n + d_1 e_{n-1} + d_2 e_{n-2}$$

con: 
$$d_0 = K_R \frac{1 + T/T_n + Tv/T}{1 + TD/T}$$

$$C_1 = \frac{1 + 2TD/T}{1 + TD/T}$$

$$d_1 = K_R \frac{-1 - 2Tv/T}{1 + TD/T}$$

$$C_2 = \frac{-TD/T}{1 + TD/T}$$

$$d_2 = K_R \frac{Tv/T}{1 + TD/T}$$

Como comentario final de este capítulo mostramos las figuras III.1 y III.2.

En la figura III.1 se muestra  $Y_n$  vs  $t/T$  para un punto exploratorio  $n = 3$ , un salto de valor positivo de cero a diez.

Como se esperaba reaccionó el controlador - debido a la parte derivativa a medida que es incrementado el valor de  $n$ , el regulador actúa en forma normal. Mientras en la figura III.2 se ve que el controlador  $PIDT_1$  mejora la respuesta sobre todo la influencia de la parte derivativa.

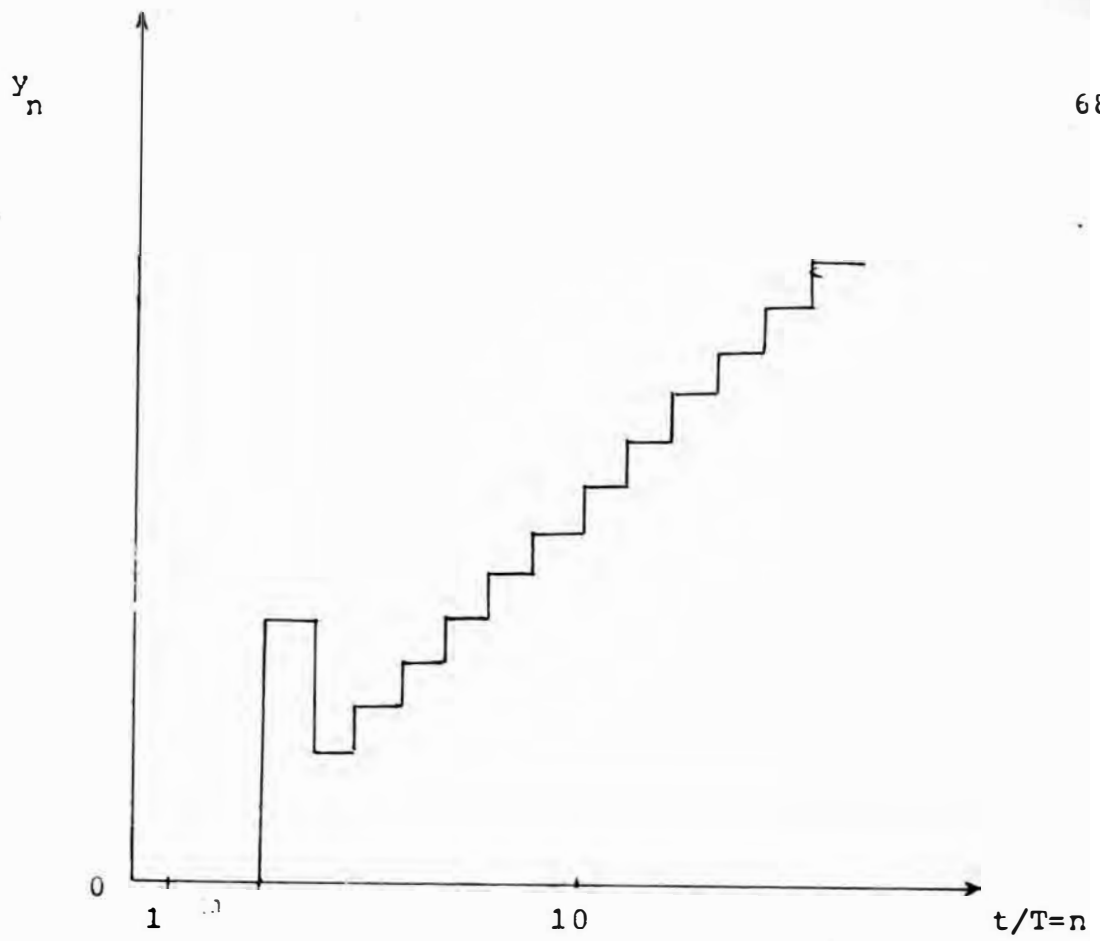


Fig.(3.1) Regulador que usa algoritmo PID.

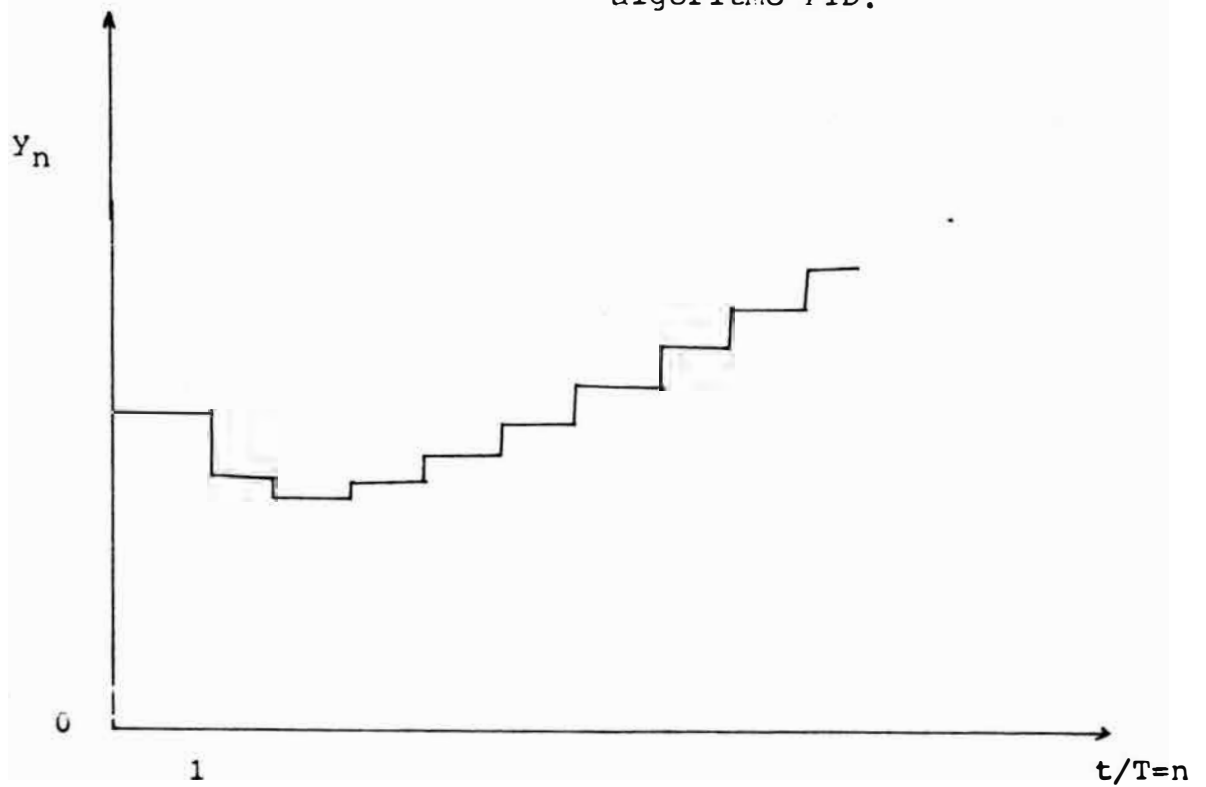


Fig.(3.2) Regulador que usa algoritmo PIDT1.

## CAPITULO IV

### PROGRAMACION Y APLICACION DEL CONTROLADOR PID

#### Introducción

El programa de regulación consta de un programa principal (MAIN Program) y de varios subprogramas o subrutinas.

De los cuales tres son programas matemáticos que vienen a ser las subrutinas de resta, suma y multiplicación con punto decimal flotante, otros dos para la interfase Digital-Analógica y Analógica-Digital.

Como las operaciones deben ser ejecutadas en los tiempos justos de muestreo se pueden programar los retardos (delays) mediante dos subrutinas del Microcomputador MWK 6802D5E, que en su sistema operativo posee dos subrutinas de retardo de 1 ms y 25 ms, localizados en las direcciones F171 y F169 respectivamente.

#### 4.1 Descripción del Programa Principal

El programa principal consta de las siguientes variables:

$X_N$  es la variable controlada

$W_N$  la referencia



$$E_N = W_N - X_N \text{ error}$$

$$Y_N = E_N D_0 + Z_W \text{ fuerza de control}$$

$$Z_W = Y_{N-1} + D_1 E_{N-1} + D_2 E_{N-2}$$

Los datos o corrientes serían:

$D_0$ ,  $D_1$ ,  $D_2$  que vienen a ser funciones de  $K_R$ ,  $T_Y$ ,  $T_n$  y  $T$ .

$K_R$  = corriente proporcional

$T_V$  = corriente de tiempo derivativa

$T_n$  = corriente de tiempo integral

$T$  = período de muestreo. Estos definidos por la ecuación 12 del Capítulo III.

Podemos definir el programa así:

$X_N$  llega del conversor A/D por el PIA usaremos para esto el puesto A.

Almacenamos  $X_N$  en un lugar de memoria, saltamos a la subrutina de resta  $E_N = W_N = X_N$ .

Saltamos a la Subrutina de multiplicación  $D_0 \cdot E_N$

Almacenamos el resultado en un lugar de memoria.

Saltamos a la subrutina de suma  $Y_N = E_N D_0 + Z_W$

$Z_W$  está almacenado en un lugar de memoria.

El resultado  $Y_N$  debemos sacarlo a un conversor D/A por el puerto B del PIA.

Introducimos un retardo a la C.P.U. de T (período de muestreo). Cargamos el registro índice  $N = 10$ , hacemos una transferencia:

$$\begin{array}{cc} Y_{N-1} & Y_N \\ E_{N-2} & E_{N-1} \end{array}$$

Decrementamos el registro índice. Preguntamos si el registro índice es igual a cero, si es paramos, sino almacenamos  $E_{N-2}, D_2, E_{N-1}, D_1, Y_{N-1}$

Calculamos  $Z_W$

Saltamos a la subrutina  $E_{N-2}, D_2$  de multiplicación. Saltamos a la subrutina de multiplicación  $E_{N-1}, D_1$ . Saltamos a la subrutina de suma  $E_{N-2}, D_2 + E_{N-1}, D_1$ . Saltamos a la subrutina de suma del resultado anterior con  $Y_{N-1}$  o sea

$$Z_W = Y_{N-1} + D_1 E_{N-1} + D_2 E_{N-2}$$

Almacenamos el resultado en  $Z_W$  en un lugar de memoria, luego volvemos al primer paso  $W_N - X_N = E_N$  para el segundo período de muestreo.

#### 4.2 Subprogramas matemáticos

El formato usado, para las operaciones con punto decimal flotante es el siguiente:

3 bytes de Mantisa  
 1 byte de exponente  
 1 byte de signo del exponente.

El primer bit de la Mantis siempre es 1 luego debe leerse .1

La suma estaría definida por ADD:  $AR = AR + X$

La resta SUBTR:  $AR = AR - X$

La multiplicación Mult:  $AR = AR \times X$

#### 4.2.1 Subrutina de multiplicación con punto decimal flotante

Lenguaje de máquina	Lenguaje assembler
	425 Multiplicación
	426 Reservación del Multiplicando de AR a AR + C
	427 Reservación Multiplicador por el registro índice.
43C1 A604	428 Mult LDAA 0,X
43C3 2A03	429 BPL MULTP
43C5 73F416	430 Comp FLAG
43C8 BD4438	431 Multp JSR ZTST
43CB 5F	432 CLRB
43CC F7F40F	433 STAB PAR
43CF F7F410	434 STAB PAR + 1

43D2 F7F411	435	STAB PAR+2
43D5 G618	436	LDAB #24
43D7 74F412	437	LSR AR
43DA 76F413	438	ROR AR+1
43DD 76F414	439	ROR AR+2
43EO 2418	440 M1	BCC SHFT
43E2 A602	441	LDAA 2, X
43E4 BBF411	442	ADDA PAR+2
43E7 B7F411	443	STAA PAR+2
43EA A601	444	LDAA 1, X
43EC B9F410	445	ADCA PAR+1
43EF B7F410	446	STAA PAR+1
43F2 A600	447	LDAA 0, X
43F4 B9F40F	448	ADCA PAR
43F7 B7F40F	449	STAA PAR
43FA 76F40F	450 SHFT	ROR PAR
43FD 76F410	451	ROR PAR+1
4400 76F411	452	ROR PAR+2
4403 76F412	453	ROR AR
4406 76F413	454	ROR AR+1
4409 76F414	455	ROR AR+2
440C 5A	456	DECB
440D 26D1	457	BNE M1
440F A603	458	LDAA 3, X
4411 BBF415	459	ADDA AR+3
4414 292A	460	BVS Z1
4416 B7F415	461	STAA AR+3

4419 B6F40F	462	LDAA PAR
441C 2B11	463	BMI M2
441E 79F412	464	ROL AR
4421 79F411	465	ROL PAR+2
4424 79F410	466	ROL PAR+1
4427 79F40F	467	ROL PAR
442A 7AF415	468	DEC AR+3
442D 2911	469	BVS Z1
442F BD4443	470 M2	JSR MOVE
4432 B6F412	471	LDAA AR
4435 2709	472	BEQ Z1
4437 39	473	RTS
	474 * ZTsT	
4438 BEF412	475 ZTsT	LDAA AR
443B 2602	476	BNE R1
443D 31	477	INS
443E 31	478	INS
443F 39	479 R1	RTS
4440 7E4555	480 Z1	JMP Zero
	481 * Move	
4443 B6F40F	482 Move	LDAA PAR
4446 B7F412	483	STAA AR
4449 B6F410	484	LDAA PAR+1
444C B7F413	485	STAA AR+1
444F B6F411	486	LDAA PAR+2
4452 B7F414	487	STAA AR+2
4455 39	488	RTS

#### 4.2.2 Subrutina de adición y resta con punto decimal flotante

Lenguaje de Máquina		Lenguaje Assambler
	553 *	sub-rutina de adición
	554 *	Reservación de 1er de AR a AR+4
	555*	
02DE A604	556ADD	LDAA 4,X
02E0 2A04	557	BPL ADDP
02E2 BDO357	558	JSR SUBP
02E5 39	559	RTS
02E6 DF19	560 ADDP	STX T1
02E8 BDO361	561	JSR ADJEXP
02EB 9614	562	LDAA AR
02ED 9A15	563	ORAA AR+1
02EF 9A16	564	ORAA AR+2
02F1 2607	565	BNE A1
02F3 7E0018	566	CLR FLAG
02F6 BDO3AA	567	JSR GET
02F9 39	568	RTS
02FA 9618	569 A1	LOAA FLAG
02FC 2703	570	BEQ L1
02FE BD0398	571	JSR COMPL
0301 BD02B8	572 L1	JSR ADD1
0304 250C	573	BCS L2
0306 9618	574	LOAA FLAG
0308 2719	575	BEQ OUT1

030A	BD0398	576	JSR COMPL
030D	BD032E	577	JSR NORM
0310	2011	578	BRA OUT1
0312	9618	579 L2	LDAA FLAG
0314	2610	580	BNE L3
0316	0D	581	SEC
0317	760014	582	ROR AR
031A	760015	583	ROR AR+1
031D	760016	584	ROR AR+2
0320	7C0017	585	INC AR+3
0323	DE19	586 OUT1	LDX T1
0325	39	587	RTS
0326	7F0018	588L3	CLR FLAG
0329	BD032E	589	JSR NORM
032C	20F5	590	BRA OUT1
		591	* Sub normalización
		592	*
032E	9614	593 NORM	LDAA AR
0330	2B15	594	BMI R2
0332	8619	595	LDAA #25
0334	4A	596 L5	DECA
0335	2718	597	BEQ L4
0337	7A0017	598	DEC AR+3
033A	290C	599	BVS ZERO
033C	780016	600	ASL AR+2
033F	790015	601	ROL AR+1
0342	790014	602	ROL AR
0345	2AED	603	BPL L5

0347 39	604 R2	RTS
	605 *	
	606 * Sub Zero	
	607 *	
0348 4F	608 Zero CLRA	
0349 9714	609	STAR AR
034B 3715	610	STAA AR+1
034D 9716	611	STAA AR+2
034F 8680	612 L4	LDAA # \$ 80
0351 9717	613	STAA AR+3
0353 7F001B	614	CLR FLAG
0356 39	615	RTS
	616 * SUBP	
	617 *	
0357 730018	618 SUBP	COMP FLAG
035A BDO2E6	619	JSR ADDP
035D 7E0018	620	COM FLAG
0360 39	621	RTS
	622**	
	623 * Sub ADJExp	
	624	
0361 A603	625 ADJExp	LDAA 3,X
0363 9117	626	COMPA AR+3
0365 2712	627	BEQ RETI
0367 2D11	628	BLT RELOC
0369 740014	629 ADJI	LSR AR
036C 760015	630	ROR AR+1



036F 760016	631	ROR AR+2
0372 7C0017	632	INC AR+3
0375 9117	633	COMPA AR+3
0377 26F0	634	BNE ADJ1
0379 39	635 RET1	RTS
	636 *	
	637 * RELOC	
	638 *	
037A E600	639 RELOC	LDAB 0,X
037C D711	640	STAB PAR
037E E601	641	LDAB 1,X
0380 D712	642	STAB PAR+1
0382 E602	643	LDAB 2,X
0384 D713	644	STAB PAR+2
0386 740011	645 ADJ2	LSR PAR
0389 760012	646	ROR PAR+1
038C 760013	647	ROR PAR+1
038F 4C	648	INCA
0390 9117	649	COMPA AR+3
0392 26F2	650	BNE ADJ2
0394 CE0011	651	LDX # PAR
0397 39	652	RTS
	653 *	
	654 * COMPL	
	655 *	
0998 4F	656 COMPL	CLRA
0399 9016	657	SUBA AR+2

039B 9716	658	STAA AR+2
039D 8600	659	* LDAR #00
039F 9215	660	SBCA AR+1
03A1 9715	661	STAA AR+1
03A3 8600	662	LDAA #00
03A5 9214	663	SBCA AR
03A7 9714	664	STAA AR
03A9 39	665	RTS
	666 *	
	667 * GET	
	668 *	
03AA A600	669 GET	LDAA 0,X
03AC 9714	670	STAA AR
03AE A601	671	LDAA 1,X
03B0 9715	672	STAA AR+1
03B2 A602	673	LDAA 2,X
03B4 9716	674	STAA AR+2
03B6 A603	675	LOAA 3,X
03B8 9717	676	STAA AR+e
03BA	677	RTS
	678 *	
	679 * Sub substracción o Resta	
	680 *	
03BB A604	681 SUBTR	LDAA 4,X
03BD 2A98	682	BPL SUBP
03BF BD02E6	683	JSR ADDP
03C2 39	684	RTS

#### 4.3 Subrutina de Retardo (delay)

Suponiendo un retardo de 500 ms o sea 0.5 seg., el programa sería el siguiente:

Lenguaje de Máquina		Lenguaje Assembler
0000 CE0014	10	LDX #\$14
0003 BDF169	11 SUBSO	JSR DL y 25
0006 09	12	DEX
0007 26 FA	13	BNE
0009 39	14	RTS SUBSO

4.4 Inicialización y programación del PIA, como se dijo en la sección 1.5.1.4 el PIA 6821 posee dos pines para seleccionar registros pero tiene seis registros internos.

Luego hay un registro de control de puerto A y B, CRA y CRB, el bit dos de este registro nos selecciona el registro PRA o el DDRA según sea "1" ó "0" respectivamente.

Ejemplo de direccionamiento del puerto "A" con los ocho bits como salida y todos en nivel alto "1"

Lenguaje de Máquina		Lenguaje Assembler
E000 4F	20	CLRA
E001 B7E481	21	STAA \$E481
E004 43	22	COMA
E005 B7 E480	23	STAA \$E480
E008 8604	24	LDA # 04

E00A B7E481	25	STAA E481
E00B 86FF	26	LDAA # FF
E010 B7E480	27	STAA \$E480
E013 39	28	RTS

Para el uso del puesto B hay que considerar que el CRB está en E483 y el PRB y PDRB en E482.

Las líneas de control CA1 y CB1, así como CA2 y CB2 son programados mediante el registro de control CRA.

Mediante estas líneas es que se realiza el diálogo con el convertidor Analógico-Digital y el Digital-Analógico.

## CONCLUSIONES

- 1) El uso de microprocesadores en sistemas de control digital se va difundiendo cada vez más.
- 2) El consumo de memoria que requieren las subrutinas matemáticas con punto decimal flotante hace recomendable la grabación de las mismas en una memoria EPROM, mediante un quemador de memoria.
- 3) Como conclusión lógica de lo anterior se deduce que el microprocesador (CPU) necesitará de muchos ciclos de máquina para la ejecución del mismo. Este retardo puede ser crítico para la estabilidad del sistema en bucle cerrado como se puede apreciar en el Capítulo II.
- 4) Se ha dispuesto de un sistema microcomputador pequeño con un KByte en memoria RAM estática (2114), para la programación, y con un ciclo de máquina de 894.8 KHz.
- 5) La resolución  $1/2^n$ , para la conversión analógica-digital, donde  $n=8$ , en nuestro caso, nos va a originar -- problemas en el error en estado estacionario y la controlabilidad del sistema.
- 6) El programa principal y sus subrutinas matemáticas pueden ser grabados en un ROM y usados con un microprocesador como un controlador digital PID para propósitos particulares.
- 7) La mas grande contribución del microprocesador, además de su inteligencia programada es la del ahorro de gran cantidad de lógica digital cableada (hardware).

- 8) El avance de la tecnología electrónica en la integración en gran escala VLSI a posibilitado que aparezcan en el mercado los microprocesadores de 16, y recientemente, de 32 bytes, Motorola ha lanzado al mercado - su microprocesador M68000 de 16 bytes, que es el líder dentro de los de 16 bytes. Tiene un ciclo de reloj igual a 0.1 microsegundos, varios registros de 32 bytes (gran potencia de proceso). Esto nos lleva a pensar - que los problemas presentados en la resolución y el tiempo de retardo presentados en el controlador digital PID serán resueltos mucho mejor con éste.

## B I B L I O G R A F I A

1. BISHOP RON - Basic Microprocessors and the 6800, Hyden Book Company Inc.
2. CORIMAS VIÑAS JOAQUIN - Introducción al Control de Procesos por Ordenador.
3. DORF C. RICHARD - Sistemas Automáticos de Control. Fondo Educativo Internacional S.A.
4. ELGERD OLLE - Control Systems Theory.
5. HARRISON - Controles Automáticos. Fondo Educativo Internacional S.A.
6. KUO BENJAMIN - Digital Control Systems.
7. MOTOROLA, MANUAL: Microcomputer Components.
8. ALVIN V. MOORE, FRONHEISER KARL and VINAY KANNA - Microprocessor Application Manual.
9. OGATA KATSUITO - Ingeniería de Control Moderna. Prentice-Hall International, 1974.

10. MOTOROLA, MANUAL: The Acquisition and Recovery of Analog Signals in a M6800 Data Processing System.
11. ZAKS RONALD -, Del Chip al Sistema. Editorial Marcombo.