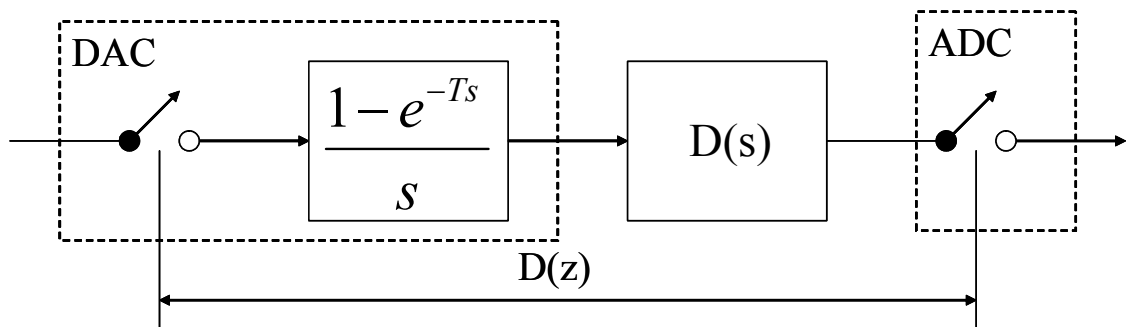


Departamento de Ingeniería Electrónica

SISTEMAS DIGITALES DE INSTRUMENTACIÓN Y CONTROL

TEMA 6

Sistemas Digitales de Control en Tiempo Discreto



Rafael Ramos Lara
Febrero 2007

TEMA 6

Sistemas Digitales de Control en Tiempo Discreto

Índice (I)

- 6.1. Sistemas de control digital
 - 6.1.1. Introducción al diseño de sistemas de control digital
 - 6.1.2. Discretización de sistemas analógicos
 - 6.1.3. Ejemplo: modelo discreto de un motor DC
- 6.2. Diseño de un control digital PID
 - 6.2.1. Introducción al control PID digital
 - 6.2.2. Aproximación rectangular del control PID
 - 6.2.3. Aproximación trapezoidal del control PID
- 6.3. Arquitecturas de realización de controladores digitales
 - 6.3.1. Programación directa
 - 6.3.2. Programación estándar
 - 6.3.3. Programación serie
 - 6.3.4. Programación paralelo
 - 6.3.5. Programación en escalera

- 6.4. Ejemplo de diseño: control de un motor DC mediante el driver L293E
 - 6.4.1. Introducción al sistema de control digital de un motor
 - 6.4.2. Modelo equivalente eléctrico del motor DC
 - 6.4.3. Interface con el motor: L293E
 - 6.4.4. Control en lazo abierto de la velocidad de giro
 - 6.4.5. PWM digital
 - 6.4.6. Monitorización de la velocidad de giro
 - 6.4.7. Regulación de la velocidad de giro

6.1 Sistemas de control digital

6.1.1 Introducción al diseño de sistemas de control digital

Componentes de un sistema de control

Los componentes principales de un sistema de control son:

- Sensores
- Actuadores
- Controlador: permite obtener el comportamiento deseado del sistema a controlar, manteniendo al mismo tiempo la estabilidad del sistema

Tipos de controladores:

- **Controladores analógicos**: implementados con amplificadores, resistencias, condensadores que realizan estructuras semejantes a filtros que modifican la respuesta frecuencial del sistema
- **Controladores digitales**: implementados con microprocesadores, microcontroladores, DSP, FPGA, CPLD, etc... Necesitan conversores ADC y DAC

Controlador	Ventajas	Desventajas
Analógico	<ul style="list-style-type: none"> • Elevado ancho de banda • Elevada resolución • Fácil de diseñar 	<ul style="list-style-type: none"> • Envejecimiento de componentes • Derivas con la temperatura • Bueno para diseños simples
Digital	<ul style="list-style-type: none"> • Diseño programable • Comportamiento preciso • Implementación algoritmos complejos • Fácilmente ampliable 	<ul style="list-style-type: none"> • Dificultad de diseño • Necesita procesadores de altas prestaciones • Genera problemas numéricos

Pasos de diseño de un sistema de control:

- Obtención del modelo del sistema a controlar
- Diseño del controlador para obtener el comportamiento deseado del sistema a controlar

El diseño de controles digitales implica la conversión del sistema en una forma discreta. Dos métodos de conversión:

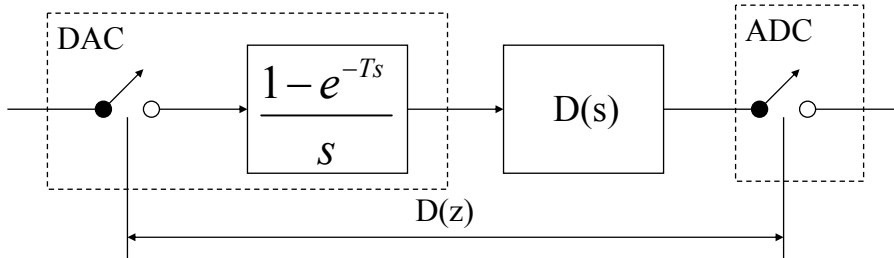
- Diseño analógico y conversión a discreto para su implementación
- Diseño discreto \Rightarrow se debe obtener el modelo de la planta en forma discreta (transformada Z)

6.1.2 Discretización de sistemas analógicos

Tres técnicas para convertir un sistema analógico en discreto:

- Mantenedor de orden zero (ZOH)
- Correspondencia polos-ceros
- Transformación bilineal

Esta técnica asume que el controlador tiene a la entrada un mantenedor de orden cero



$$D(z) = (1 - z^{-1})Z \left[\frac{D(s)}{s} \right]$$

Los polos y ceros “ s_i ” de $D(s)$ se mapean como polos y ceros de $D(z)$ de acuerdo con:

$$z_i = e^{s_i T} \quad T = \text{periodo de muestreo}$$

Si $D(s)$ tiene más polos que ceros se añaden ceros en $z=-1$ en el numerador para igualar el número de polos y ceros.

La ganancia se escoge adecuadamente para que se cumpla:

$$D(z) \Big|_{z=1} = D(s) \Big|_{s=0}$$

También denominada aproximación de Tustin o trapezoidal, utiliza la siguiente relación

$$s = \frac{2}{T} \frac{(z-1)}{z+1} \quad T = \text{periodo de muestreo}$$

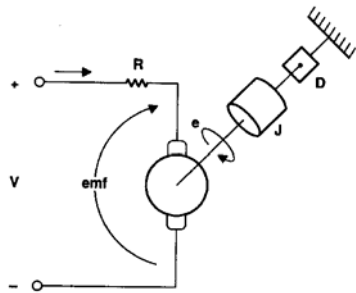
para transformar el dominio en “s” en un dominio en “z”

6.1.3 Modelo discreto de un motor DC

Modelo de la planta (I)

El primer paso en el diseño de un control es obtener el modelo de la planta

Ejemplo: modelo de un motor DC de imán permanente



Características eléctricas: $L \frac{di}{dt} + Ri = V - emf$

L : inductancia del motor

R : resistencia

V : tensión de alimentación

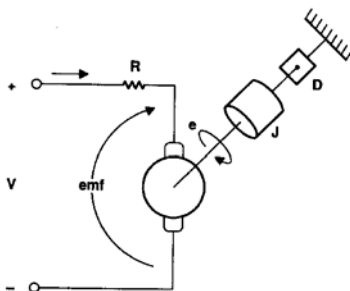
i : corriente

emf : fuerza contraelectromotriz = $K_e \cdot \theta$

K_e : constante de fuerza contraelectromotriz

Modelo de la planta (II)

Características mecánicas: $J_M \frac{d^2\theta}{dt^2} + B \frac{d\theta}{dt} + K\theta = T_L - J_L \frac{d^2\theta}{dt^2}$



J_M : inercia del motor

θ : desplazamiento angular

K : constante de rigidez

B : coeficiente de fricción viscosa

J_L : inercia de la carga = $K_t \cdot i$

T_L : par de rotación de la carga

K_t : constante de par de rotación

Parámetros del motor
DC Pittman 9412G316

$$\begin{aligned}
 R &= 6.4 \Omega \\
 J &= J_m + J_L = 60 \cdot 10^{-6} \text{ kg}\cdot\text{m}^2 \\
 K_t &= 0.0207 \text{ (N}\cdot\text{m)/A} \\
 K_e &= 0.0206 \text{ volt/(rad/s)}
 \end{aligned}$$

Modelo eléctrico

$$L \frac{di}{dt} + Ri = V - emf$$

Modelo mecánico

$$J_M \frac{d^2\theta}{dt^2} + B \frac{d\theta}{dt} + K\theta = T_L - J_L \frac{d^2\theta}{dt^2}$$

$$G(s) = \frac{\theta(s)}{V(s)} = \frac{J^{-1}(K_t/R)}{s(s + J^{-1}(K_t K_e/R))}$$

$$G_m(s) = \frac{\theta(s)}{V(s)} = \frac{53.906}{s(s + 1.116)}$$

Transformación ZOH: $G(z) = (1 - z^{-1}) \mathcal{Z} \left[\mathcal{L}^{-1} \left(\frac{G(s)}{s} \right) \right]$

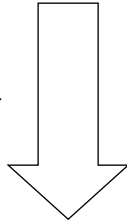
$$\frac{G(s)}{s} = \frac{b}{s^2(s+a)} = \frac{A_1}{s} + \frac{A_2}{s^2} + \frac{A_3}{s+a} \quad \begin{aligned} b &= J^{-1}(K_t/R) \\ a &= J^{-1}(K_t K_e/R) \end{aligned}$$

$$\frac{G(s)}{s} = \frac{(-b/a^2)}{s} + \frac{(b/a)}{s^2} + \frac{(b/a^2)}{s+a} \quad \begin{aligned} T: & \text{ periodo de muestreo} \\ T &= 0.001 \end{aligned}$$

$$G(z) = \frac{b/a^2 (e^{-aT} - 1 + aT)z^{-1} + b/a^2 (1 - e^{-aT} - aTe^{-aT})z^{-2}}{1 - (1 + e^{-aT})z^{-1} + e^{-aT}z^{-2}}$$

$$G(z) = \frac{b/a^2 (e^{-aT} - 1 + aT)z^{-1} + b/a^2 (1 - e^{-aT} - aTe^{-aT})z^{-2}}{1 - (1 + e^{-aT})z^{-1} + e^{-aT}z^{-2}}$$

$$\begin{aligned} a &= 1.116 \\ b &= 53.906 \\ T &= 0.001 \end{aligned}$$



K_m : Factor de ganancia

$$G(z) = \frac{\theta(z)}{V(z)} = \frac{0.2694z^{-1} + 0.2693z^{-2}}{1 - 1.999z^{-1} + 0.999z^{-2}} \cdot K_m$$

6.2 Diseño de un control digital PID

- **Técnicas de compensación:** el controlador añade polos y ceros al sistema para obtener la respuesta deseada .
- **PID:** el control PID es la suma de tres términos: **Proporcional** al error + **Integral** del error + **Derivada** del error. Es el algoritmo de control más utilizado.
- **Deadbeat:** Se utiliza cuando se desea una respuesta rápida. Se diseña en el dominio Z
- **Modelos en el espacio de estado:** describen matricialmente el modelo de sistema a controlar. Se utiliza cuando hay muchas variables de estado a controlar.
- **Control óptimo:** se utiliza cuando se desea minimizar un parámetro específico del sistema (p.e. Tiempo de establecimiento, energía, ...). El controlador o compensador debe minimizar el parámetro.
- **Control adaptativo:** se utiliza en sistemas cuyos parámetros cambian con el tiempo haciendo inestable el control. El control adaptativo sigue los cambios de la planta y rediseña el controlador para obtener un control óptimo del sistema.

6.2.1 Introducción al control PID digital

Expresión general control PID

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

Minimiza el error

Reduce el error a cero en régimen permanente

- Incrementa la estabilidad de la planta
- Acción anticipativa que reduce el sobreimpulso

K_p : constante proporcional

K_i : constante integral

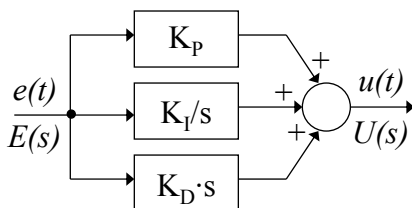
K_d : constante derivativa

$u(t)$: salida del control

$e(t)$: señal de error

Controlador PID analógico y discreto

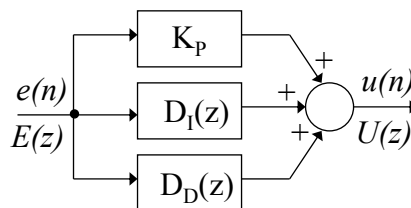
Control PID analógico



Función de transferencia

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

Control PID digital



Función de transferencia

$$U(z) = K_p E(z) + D_i(z) E(z) + D_d(z) E(z)$$



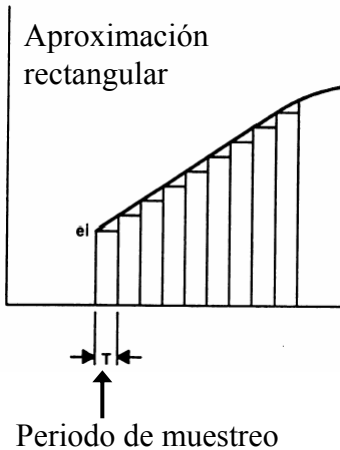
Existen diversos modos de implementar $D_i(z)$

Dos técnicas de implementación del control PID digital:

- Aproximación rectangular:
 - El diseño se realiza en el dominio analógico y a continuación se transfiere al dominio discreto
 - Es fácil de implementar y proporciona resultados satisfactorios
- Aproximación trapezoidal:
 - El diseño se realiza en el dominio discreto directamente utilizando técnicas de ubicación de polos

6.2.2 Aproximación rectangular del control PID

Aproximación rectangular del PID (I)



Término proporcional

$$K_p e(t) = K_p e(n)$$

Término integral

$$K_i \int e(t) = K_i T \sum_i e_i$$

Término derivativo

Si T es suficientemente pequeño se puede aproximar por:

$$K_d \frac{e(t)}{dt} = K_d \frac{e(n) - e(n-1)}{T}$$

Si se conoce $e(n+1)$ se puede obtener una mejor aproximación de la derivada:

$$K_d \frac{e(t)}{dt} = K_d \frac{e(n+1) - e(n)}{T}$$

Algoritmo de posición

$$u(n) = K_p e(n) + K_i T \sum_i e_i + K_d [e(n) - e(n-1)]/T$$

Aproximación rectangular del PID (II)

Algoritmo de posición

$$u(n) = K_p e(n) + K_i T \sum_i e_i + K_d [e(n) - e(n-1)]/T$$

Inconveniente: en caso de malfuncionamiento del sistema digital que calcula $u(n)$ se podría generar una salida $u(n)=0$

Algoritmo de velocidad

$$\Delta u(n) = u(n) - u(n-2)$$

- Es el algoritmo que se utiliza habitualmente
- El sistema de control solo calcula el incremento de la señal de control
- Presenta mejor comportamiento en arranque y frente a transitorios bruscos en la señal de referencia.

Algoritmo de velocidad

$$\Delta u(n) = u(n) - u(n-2)$$



$$u(n-2) = K_p e(n-2) + K_i T \sum_{i=1}^{n-2} e_i + K_d [e(n-1) - e(n-2)]/T$$

$$u(n) - u(n-2) = K_p [e(n) - e(n-2)] + K_i T [e(n) + e(n-1)] + K_d / T [e(n) - 2e(n-1) + e(n-2)]$$



$$\begin{aligned} u(n) &= u(n-2) + K_1 e(n) + K_2 e(n-1) + K_3 e(n-2) \\ K_1 &= K_p + K_d / T + K_i T \\ K_2 &= K_i T - 2K_d / T \\ K_3 &= K_d / T - K_p \end{aligned}$$

Método del margen de fase (MF) y margen de ganancia (MG)

Se escoge como parámetros de diseño:

- MF = 55°
- Frecuencia de transición (fase: -180°) = 100Hz



Aplicando técnicas de control clásico en el dominio frecuencial se obtiene:

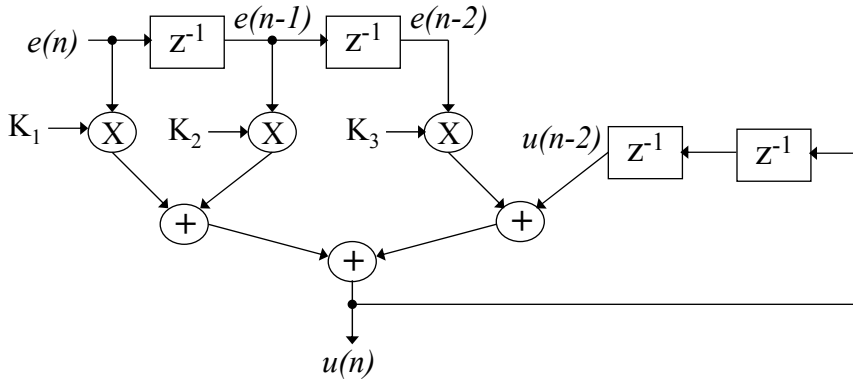
$$\begin{aligned} K_p &= 4181 \\ K_d &= 9.569 \\ K_i &= 1 \end{aligned}$$



$$\begin{aligned} \text{MG} &= 77\text{dB} (f=100\text{Hz}) \\ K_1 &= 13751 \\ K_2 &= -19138 \\ K_3 &= 5387 \end{aligned}$$

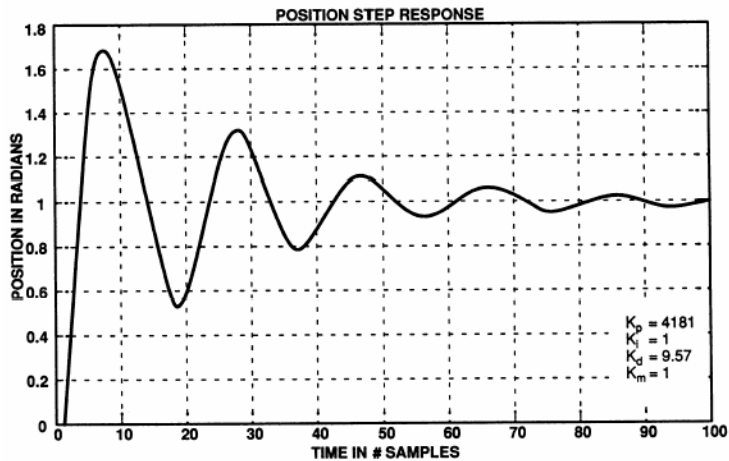
Implementación digital del PID

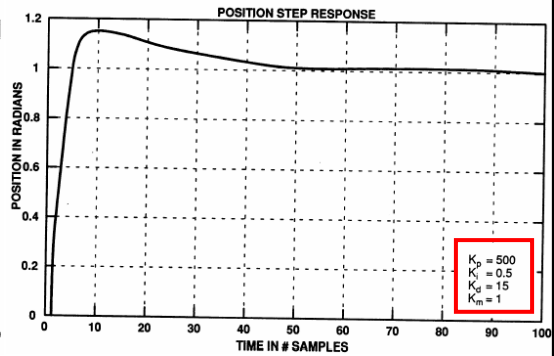
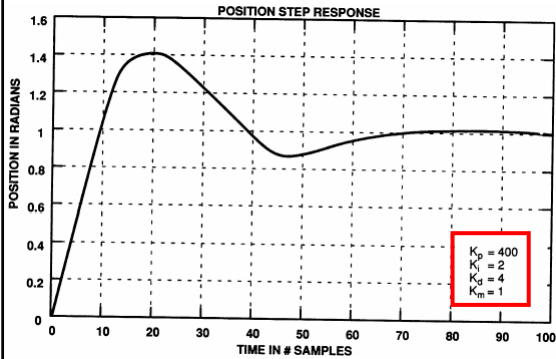
$$u(n) = u(n-2) + K_1 e(n) + K_2 e(n-1) + K_3 e(n-2)$$



Respuesta al escalón con el control PID

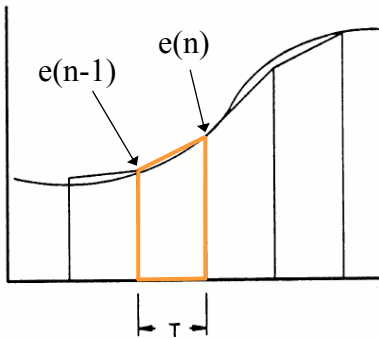
$$u(n) = u(n-2) + K_1 e(n) + K_2 e(n-1) + K_3 e(n-2)$$





6.2.3 Aproximación trapezoidal del control PID

- Se utiliza cuando se requiere una mayor precisión en la conversión discreta
- La integral se determina con la suma de trapezoides



$$\text{Área del trapezoide: } \frac{T}{2}[e(n) + e(n-1)]$$

Función transferencia término integral

$$u(n) = u(n-1) + K_I \frac{T}{2}[e(n) + e(n-1)]$$

$$U(z)(1 - z^{-1}) = K_I \frac{T}{2}[1 + z^{-1}]E(z)$$

$$D_I(z) = \frac{U(z)}{E(z)} = K_I \frac{T}{2} \frac{(1 + z^{-1})}{(1 - z^{-1})}$$

$$u(n) = K_p e(n) + K_i T \sum_i e_i + K_d [e(n) - e(n-1)]/T$$

Transformada Z de cada término

$$U(z) = K_p E(z) + K_I \frac{T}{2} \frac{(1 + z^{-1})}{(1 - z^{-1})} E(z) + \frac{K_d}{T} (1 - z^{-1}) E(z)$$

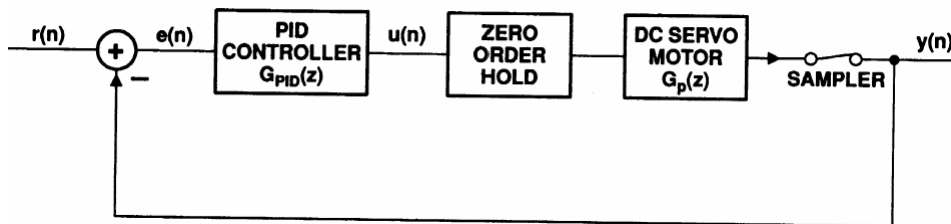
Función de transferencia discreta

$$D(z) = \frac{U(z)}{E(z)} = K_p + K_I \frac{T}{2} \frac{(1 + z^{-1})}{(1 - z^{-1})} + \frac{K_d}{T} (1 - z^{-1})$$

Reordenando términos

$$D(z) = \frac{(2TK_p + K_I T^2 + 2K_d) + (K_I T^2 - 2K_p T - 4K_d)z^{-1} + 2K_d z^{-2}}{2T(1 - z^{-1})}$$

Diseño del control PID de un motor DC



$$G_{PID}(z) = \frac{K_1 + K_2 z^{-1} + K_3 z^{-2}}{1 - z^{-1}}$$

$$\text{Con: } K_1 = K_p + \frac{K_i}{2}T + \frac{K_d}{T}, K_2 = -K_p + \frac{K_i}{2}T - \frac{2K_d}{T}, K_3 = \frac{K_d}{T}$$

$$G_p(z) = \frac{\theta(z)}{V(z)} = \frac{0.2694z^{-1} + 0.2693z^{-2}}{1 - 1.999z^{-1} + 0.999z^{-2}} \cdot K_m$$

Determinación coeficientes PID

$$G_s(z) = \frac{G_p(z)G_c(z)}{1 + G_p(z)G_c(z)} \quad \text{Función de transferencia global del sistema}$$

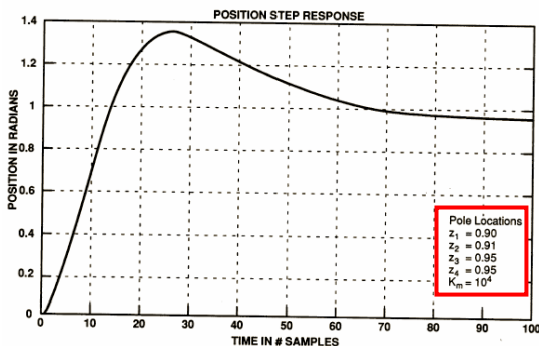
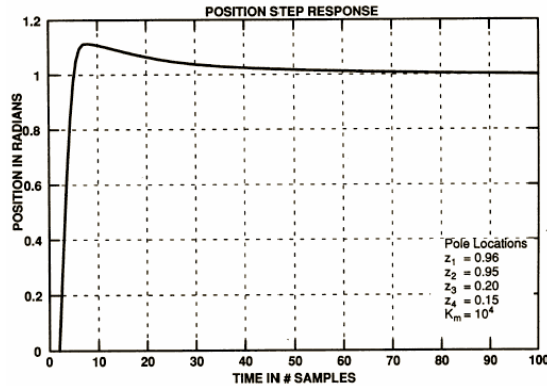
Matlab \Rightarrow ubicación de polos en 0.96, 0.95, 0.2 y 0.15

Resolviendo el denominador para la ubicación de polos propuesta se obtiene:

$$\begin{matrix} K_1 = 1.4795 \\ K_2 = -2.845 \\ K_3 = 1.3636 \end{matrix} \quad \Rightarrow \quad G_{PID}(z) = \frac{K_1 + K_2 z^{-1} + K_3 z^{-2}}{1 - z^{-1}}$$

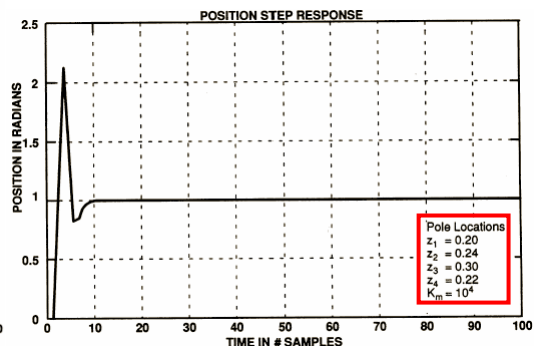
$$u(n) = u(n-1) + K_1 e(n) + K_2 e(n-1) + K_3 e(n-2)$$

$$u(n) = u(n-1) + K_1 e(n) + K_2 e(n-1) + K_3 e(n-2)$$



Polos cerca del círculo unidad:

- Aumenta el tiempo de respuesta
- El sistema puede hacerse inestable

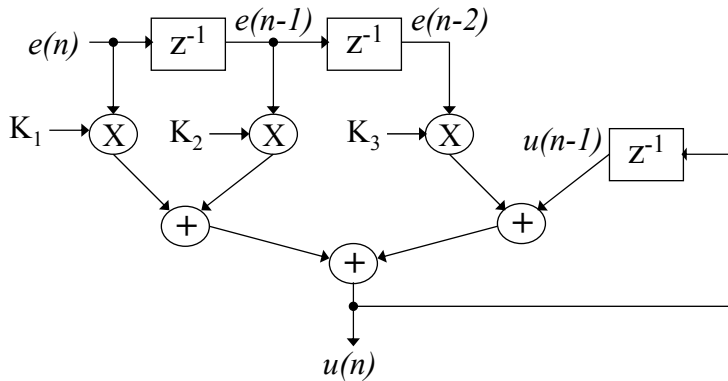


Polos cerca del origen:

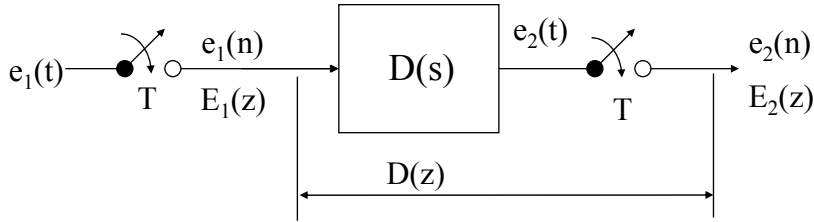
- Disminuye el tiempo de respuesta
- Aumenta el sobre impulso

Implementación digital del PID

$$u(n) = u(n-1) + K_1 e(n) + K_2 e(n-1) + K_3 e(n-2)$$



6.3 Arquitecturas de realización de controladores digitales



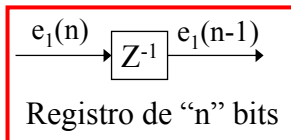
Expresión general de la función de transferencia del controlador digital

$$D(z) = \frac{E_2(z)}{E_1(z)} = \frac{b_0 + b_1z^{-1} + b_2z^{-2} + \dots + b_mz^{-m}}{a_0 + a_1z^{-1} + a_2z^{-2} + \dots + a_pz^{-p}}$$

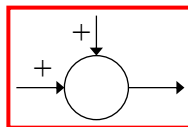
$$D(z) = \frac{E_2(z)}{E_1(z)} = \frac{b_0 + b_1z^{-1} + b_2z^{-2} + \dots + b_mz^{-m}}{a_0 + a_1z^{-1} + a_2z^{-2} + \dots + a_pz^{-p}}$$



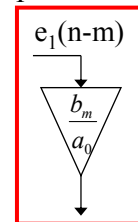
Elementos de retardo



Sumadores c.a.2



Multiplicadores c.a.2



Estructuras de implementación

- Programación directa: implementa la ecuación en diferencias
 - Programación estándar: reduce el número de registros a utilizar
 - Programación en serie
 - Programación en paralelo
 - Programación en escalera
- } La función de transferencia se descompone en funciones de primer y segundo orden para disminuir los errores de truncado de coeficientes

6.3.1 Programación directa

$$D(z) = \frac{E_2(z)}{E_1(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_m z^{-m}}{a_0 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_p z^{-p}}$$



$$E_2(z) [a_0 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_p z^{-p}] = E_1(z) [b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_m z^{-m}]$$



Transformada "z" inversa

$$a_0 \cdot e_2(n) + \sum_{i=1}^p a_i e_2(n-i) = \sum_{i=0}^m b_i e_1(n-i)$$

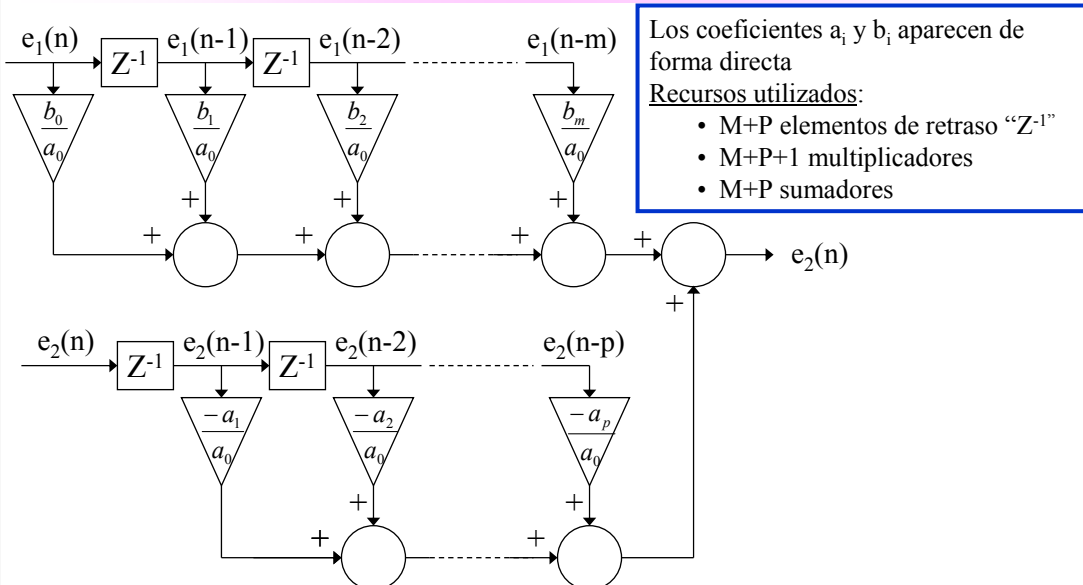


Salida actual $\rightarrow e_2(n) = \frac{1}{a_0} \sum_{i=0}^m b_i e_1(n-i) - \frac{1}{a_0} \sum_{i=1}^p a_i e_2(n-i)$

Entradas

Salidas anteriores

Implementación directa



6.3.2 Programación estándar

Descomposición de la función de transferencia

$$D(z) = \frac{E_2(z)}{E_1(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_m z^{-m}}{a_0 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_p z^{-p}} = \frac{E_2(z)}{H(z)} \frac{H(z)}{E_1(z)}$$



$$\frac{E_2(z)}{H(z)} = b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_m z^{-m} \quad \frac{H(z)}{E_1(z)} = \frac{1}{a_0 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_p z^{-p}}$$

Transformada “z” inversa

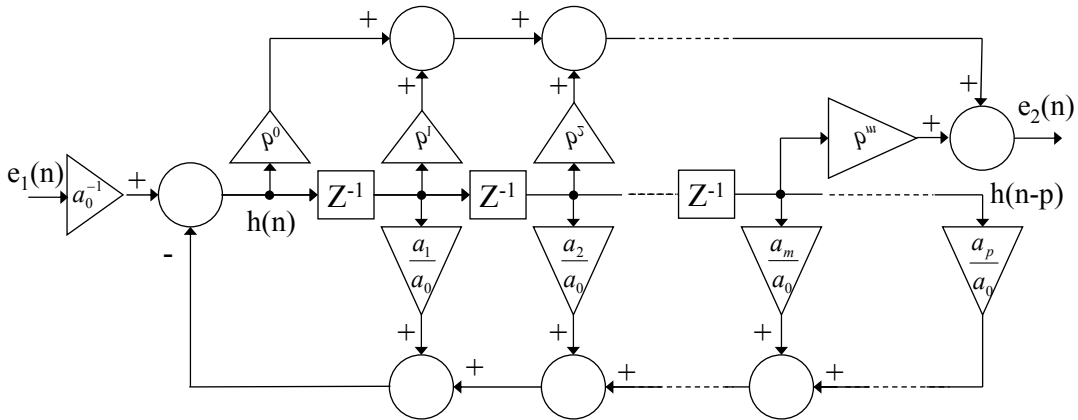
$$E_2(z) = [b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_m z^{-m}] \cdot H(z) \quad \Rightarrow e_2(n) = \sum_{i=0}^m b_i h(n-i)$$

$$H(z) = \frac{1}{a_0} E_1(z) - \frac{1}{a_0} [a_1 z^{-1} + a_2 z^{-2} + \dots + a_p z^{-p}] \cdot H(z) \Rightarrow h(n) = \frac{1}{a_0} e_1(n) - \frac{1}{a_0} \sum_{i=1}^p a_i h(n-i)$$

Implementación estándar

Recursos utilizados:

- P elementos de retraso “Z⁻¹”
- P+M+2 multiplicadores
- P+M+1 sumadores



Tema 6: Sistema Digitales de Control en Tiempo Discreto

51

Fuentes de error

La precisión en la implementación de controles digitales es importante para obtener un buen resultado.

Hay tres fuentes de error que afectan a la precisión:

- El error de cuantificación de los ADC
- Redondeo en las operaciones aritméticas
- Truncamiento de los coeficientes a_i y $b_i \Rightarrow$ este error aumenta al aumentar el orden de la función de transferencia \Rightarrow un pequeño error en los coeficientes de un filtro de orden elevado provoca un gran error en la ubicación de polos y ceros

Este error se puede reducir matemáticamente descomponiendo las funciones de transferencia de orden elevado en combinaciones de funciones de primer y segundo orden

Tema 6: Sistema Digitales de Control en Tiempo Discreto

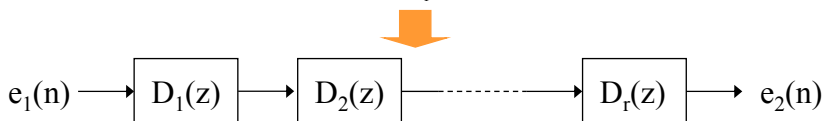
52

6.3.3 Programación serie

Descomposición de la función de transferencia

La función de transferencia se descompone en un producto de funciones sencillas de primer o segundo orden

$$D(z) = \frac{E_2(z)}{E_1(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_m z^{-m}}{a_0 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_p z^{-p}} = D_1(z) \cdot D_2(z) \cdot \dots \cdot D_r(z) = \prod_{k=1}^r D_k(z)$$



Las funciones de transferencia $D_i(z)$ dependen de los polos y ceros de $D(z)$:

Polo y cero reales

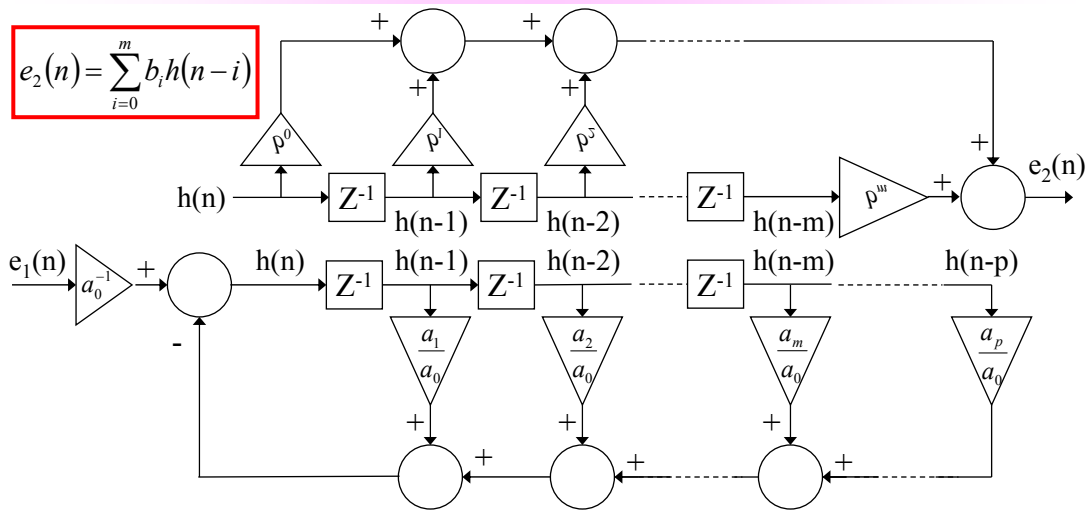
$$D_i(z) = \frac{1 + b_i z^{-1}}{1 + a_i z^{-1}}$$

Polos y ceros complejo conjugados

$$D_i(z) = \frac{1 + e_i z^{-1} + f_i z^{-2}}{1 + c_i z^{-1} + d_i z^{-2}}$$

Implementación estándar

$$e_2(n) = \sum_{i=0}^m b_i h(n-i)$$

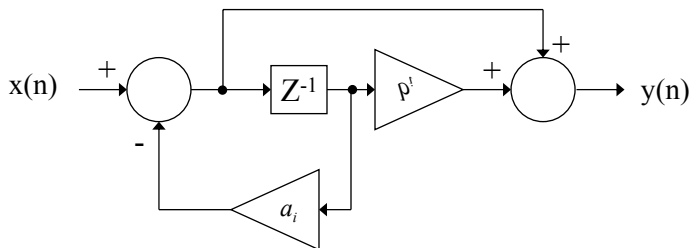


$$h(n) = \frac{1}{a_0} e_1(n) - \frac{1}{a_0} \sum_{i=1}^p a_i h(n-i)$$

Implementación funciones $D_i(z)$ (I)

Polo y cero reales

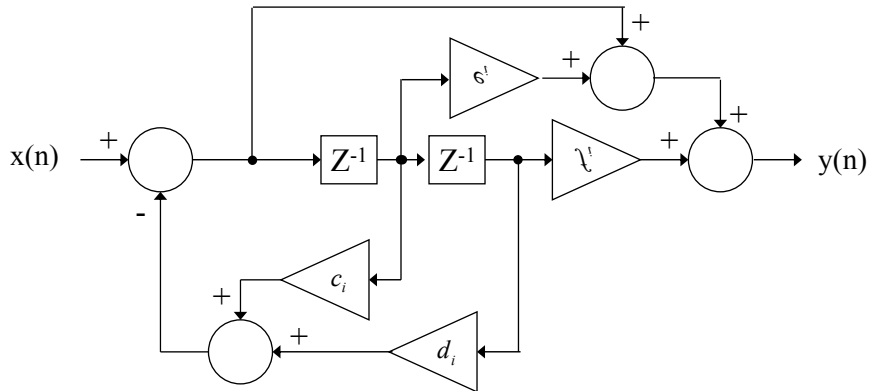
$$\frac{Y(z)}{X(z)} = \frac{1 + b_i z^{-1}}{1 + a_i z^{-1}}$$



Implementación funciones $D_i(z)$ (II)

Polos y ceros complejo conjugados

$$\frac{Y(z)}{X(z)} = \frac{1 + e_i z^{-1} + f_i z^{-2}}{1 + c_i z^{-1} + d_i z^{-2}}$$



6.3.4 Programación paralelo

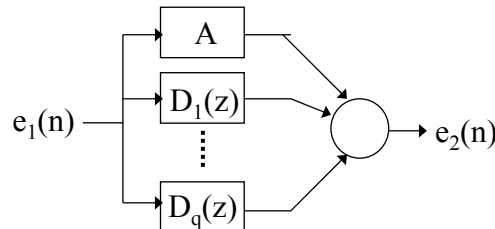
La función de transferencia se descompone en suma de fracciones parciales de primer y segundo orden:

$$D(z) = \frac{E_2(z)}{E_1(z)} = \frac{b_0 + b_1z^{-1} + b_2z^{-2} + \dots + b_mz^{-m}}{a_0 + a_1z^{-1} + a_2z^{-2} + \dots + a_pz^{-p}} = A + D_1(z) + D_2(z) + \dots + D_q(z)$$

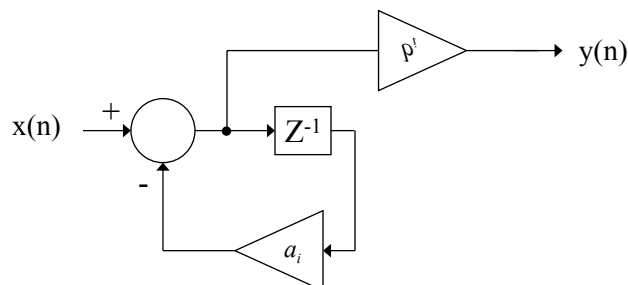
$$= A + \sum_{i=1}^j D_i(z) + \sum_{i=j+1}^q D_i(z) = A + \sum_{i=1}^j \frac{b_i}{1 + a_i z^{-1}} + \sum_{i=j+1}^q \frac{e_i + f_i z^{-1}}{1 + c_i z^{-1} + d_i z^{-2}}$$

Polos reales

Polos complejos



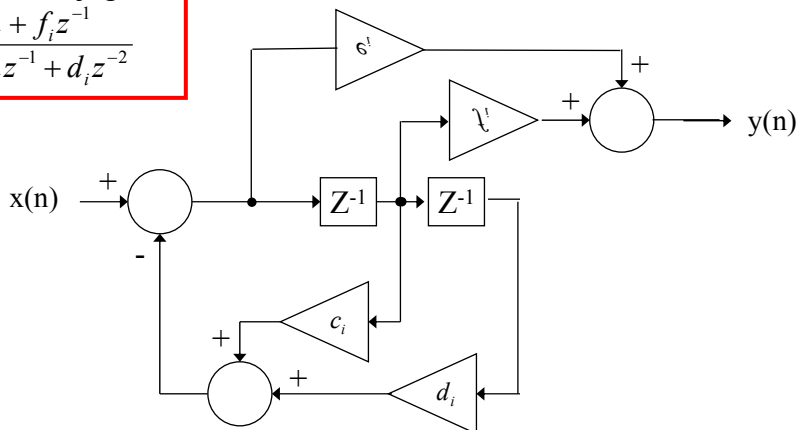
$$\text{Polos reales} \\ \frac{Y(z)}{X(z)} = \frac{b_i}{1 + a_i z^{-1}}$$



Implementación funciones $D_i(z)$ (II)

Polos complejos conjugados

$$\frac{Y(z)}{X(z)} = \frac{e_i + f_i z^{-1}}{1 + c_i z^{-1} + d_i z^{-2}}$$



6.3.5 Programación en escalera

$$D(z) = A_0 + D_1^{(B)}(z) \quad \Rightarrow \quad \begin{aligned} D_i^{(B)}(z) &= \frac{1}{B_i z + G_i^{(A)}(z)} \quad \text{con } i = 1, 2, \dots, p-1 \\ D_i^{(A)}(z) &= \frac{1}{A_i + G_{i+1}^{(B)}(z)} \quad \text{con } i = 1, 2, \dots, p-1 \\ D_p^{(B)}(z) &= \frac{1}{B_p z + \frac{1}{A_p}} \end{aligned}$$

$$D(z) = A_0 + \frac{1}{B_1 z + \frac{1}{A_1 + \frac{1}{B_2 z + \frac{1}{\vdots}}}}} + \frac{1}{A_{p-1} \frac{1}{B_p z + \frac{1}{A_p}}}$$

Ejemplo de programación en escalera (I)

$$D(z) = A_0 + \frac{1}{B_1 z + \frac{1}{A_1 + \frac{1}{B_2 z + \frac{1}{A_2}}}} \quad \text{p=2} \quad \Rightarrow \quad D_p^{(B)}(z) = \frac{1}{B_p z + \frac{1}{A_p}}$$

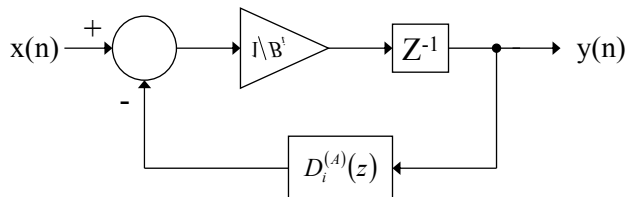
$$D(z) = A_0 + \frac{1}{B_1 z + \frac{1}{A_1 + D_2^{(B)}(z)}} = A_0 + \frac{1}{B_1 z + D_1^{(A)}(z)} = A_0 + D_1^{(B)}(z)$$

$D_i^{(B)}(z)$ se puede escribir como:

$$D_i^{(B)}(z) = \frac{Y(z)}{X(z)} = \frac{1}{B_i z + D_i^{(A)}(z)} \quad \Rightarrow \quad X_i(z) - D_i^{(A)} Y_i(z) = B_i z Y_i(z)$$

Ejemplo de programación en escalera (II)

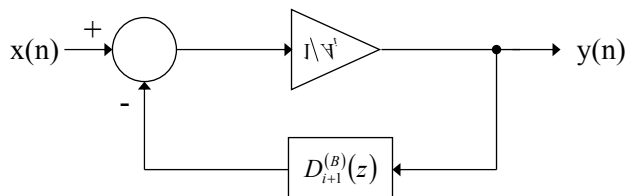
$$X_i(z) - D_i^{(A)} Y_i(z) = B_i z Y_i(z)$$



Ejemplo de programación en escalera (III)

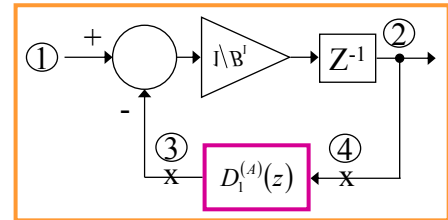
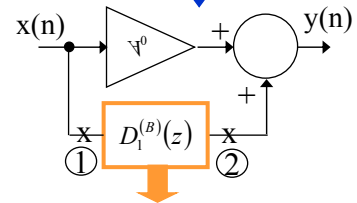
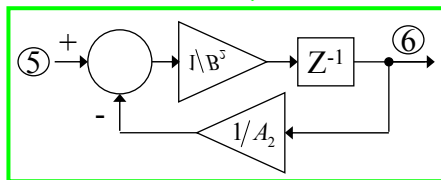
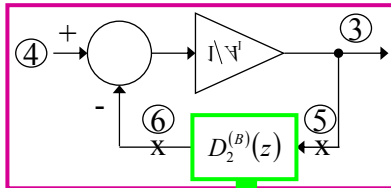
$D_i^{(A)}(z)$ se puede escribir como:

$$D_i^{(A)}(z) = \frac{Y(z)}{X(z)} = \frac{1}{A_i + D_{i+1}^{(B)}(z)} \quad \Rightarrow \quad X_i(z) - D_{i+1}^{(B)} Y_i(z) = A_i Y_i(z)$$



Ejemplo de programación en escalera (IV)

$$D(z) = A_0 + \frac{1}{B_1 z + \frac{1}{A_1 + D_2^{(B)}(z)}} = A_0 + \frac{1}{B_1 z + D_1^{(A)}(z)} = A_0 + D_1^{(B)}(z)$$

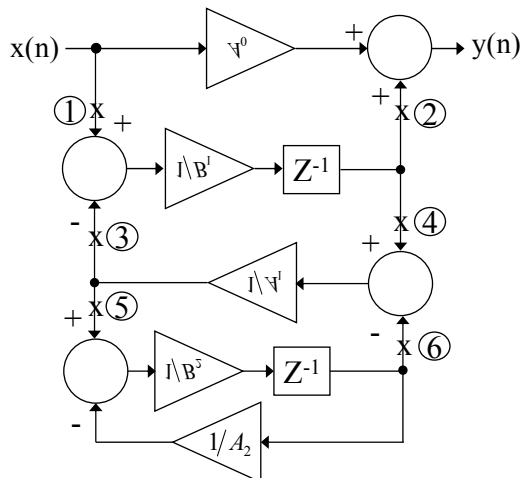


Tema 6: Sistema Digitales de Control en Tiempo Discreto

67

Ejemplo de programación en escalera (V)

Combinación de los diagramas de bloques



Tema 6: Sistema Digitales de Control en Tiempo Discreto

68

6.4 Ejemplo de diseño: control de un motor DC mediante el driver L293E

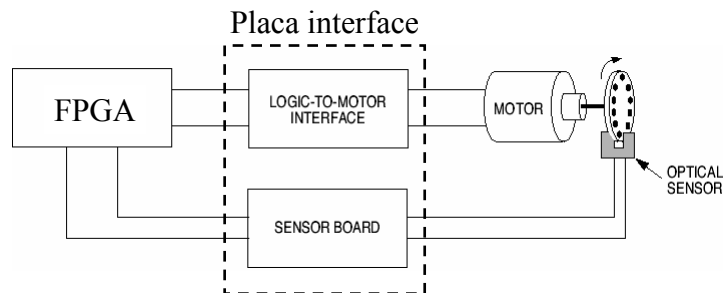
Índice

- 6.4.1. Introducción al sistema de control digital de un motor
- 6.4.2. Modelo equivalente eléctrico del motor DC
- 6.4.3. Interface con el motor: L293E
- 6.4.4. Control en lazo abierto de la velocidad de giro
- 6.4.5. PWM digital
- 6.4.6. Monitorización de la velocidad de giro
- 6.4.7. Regulación de la velocidad de giro

6.4.1 Introducción al sistema de control digital de un motor

71

Diagrama de bloques del sistema de control



Elementos básicos:

- *FPGA*: implementa el sistema digital de control
- *Interface etapa lógica-motor*: etapa de potencia controlada digitalmente que alimenta el motor DC
- *Motor*: sistema a controlar
- *Sensor óptico*: permite medir la velocidad de giro del motor

72

6.4.2 Modelo equivalente eléctrico del motor DC

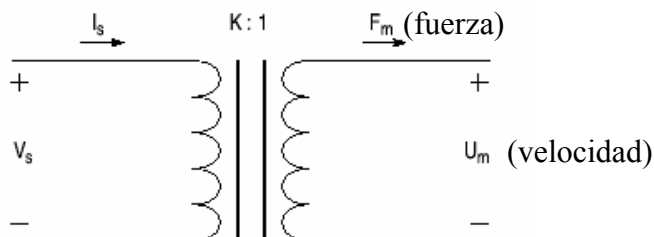
73

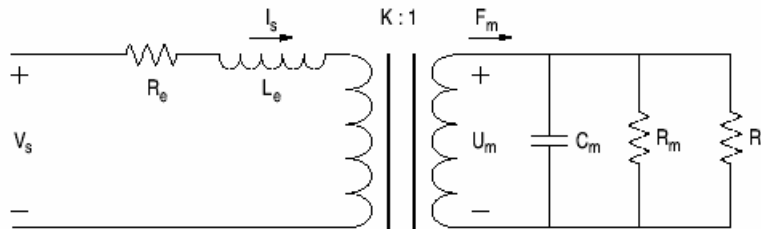
Modelo ideal del motor DC

- Transformador ideal de tensión/corriente en velocidad/fuerza
- Relaciones de transformación:

$$K \cdot I_s = F_m$$

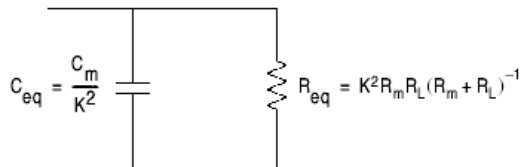
$$V_s = K \cdot U_m$$



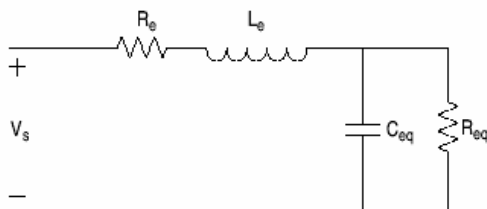


- L_e y R_e : Inductancia y resistencia del devanado del rotor
- C_m : modela la resistencia que presenta el motor a cambios de velocidad
- R_m : resistencia mecánica
- R_L : carga del motor

Circuito equivalente del secundario visto desde el primario



Modelo eléctrico equivalente de segundo orden



$$\frac{w_A}{V_S} = \frac{1}{s^2 + 2\alpha s + w_o^2}$$

$$w_o = \frac{1}{\sqrt{L_{eq} \cdot C_{eq}}}$$

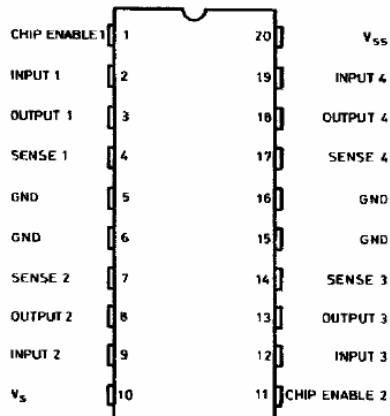
$$V = D \cdot V_{REF}$$

6.4.3 Interfase con el motor: L293E

77

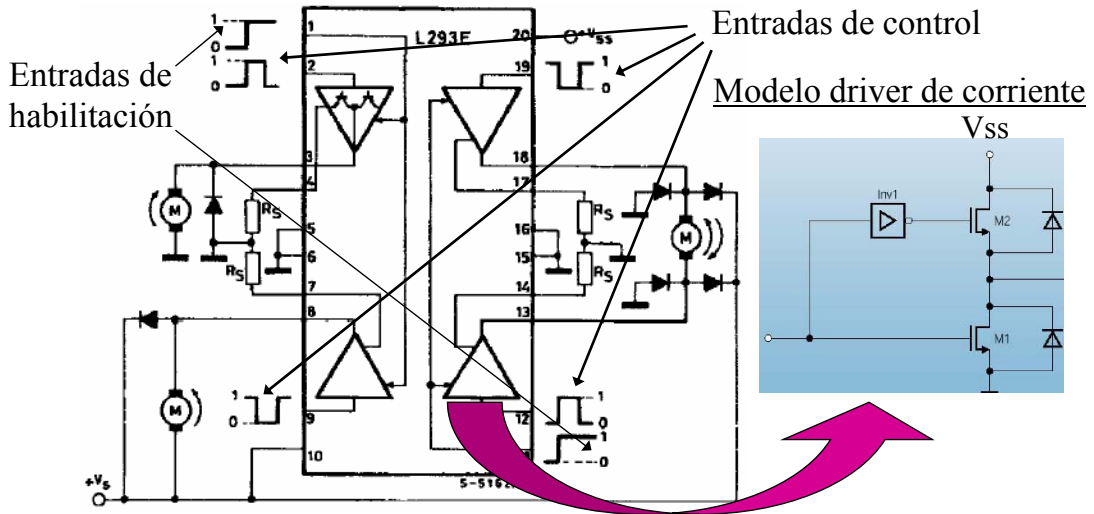
Driver de cuatro canales Push-Pull

POWERDIP (16+2+2) - L293E

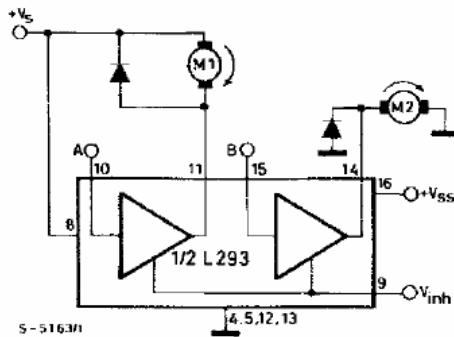


Características:

- I_{out}: 1A por canal
- I_{out} de pico: 2 A por canal
- Entrada de inhibición
- Protección contra sobre-temperatura

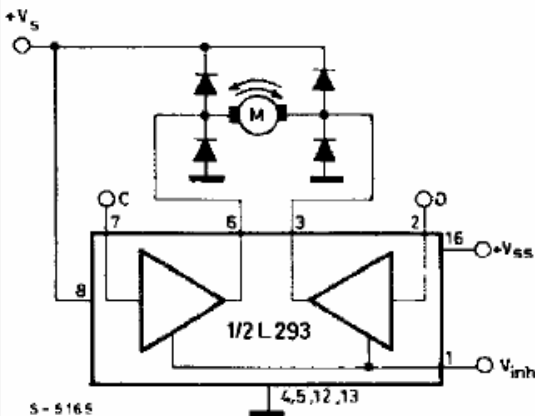
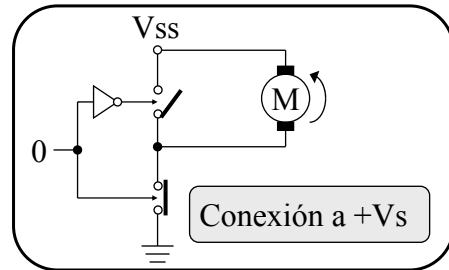
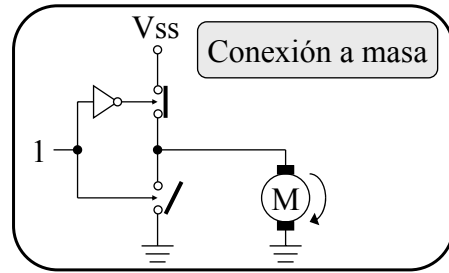
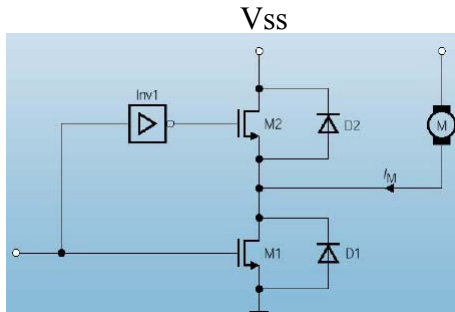


Conexión a +Vs

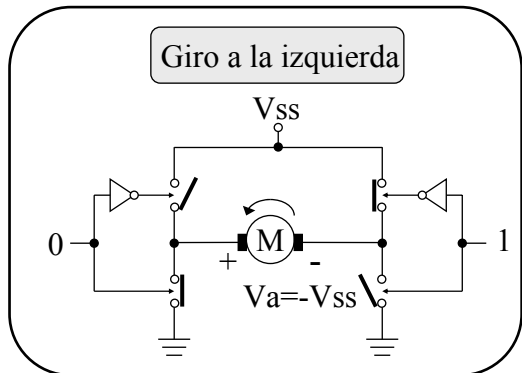
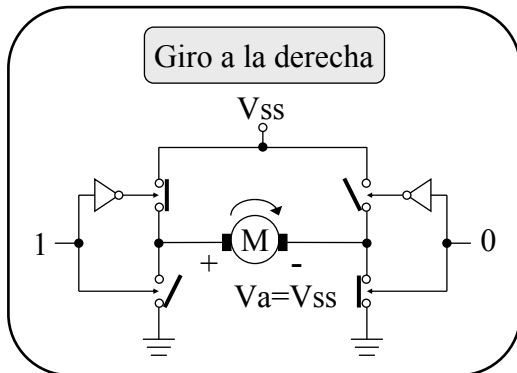


Conexión a masa

V _{inh}	A	M1	B	M2
H	H	Fast Motor Stop	H	Run
H	L	Run	L	Fast Motor Stop
L	X	Free Running Motor Stop	X	Free Running Motor Stop



Inputs	Function	
$V_{inh} = H$	$C = H ; D = L$	Turn Right
	$C = L ; D = H$	Turn Left
	$C = D$	Fast Motor Stop
$V_{inh} = L$	$C = X ; D = X$	Free Running Motor Stop



Inputs	Function	
$V_{inh} = H$	C = H ; D = L	Turn Right
	C = L ; D = H	Turn Left
	C = D	Fast Motor Stop
$V_{inh} = L$	C = X ; D = X	Free Running Motor Stop

6.4.4 Control en lazo abierto de la velocidad de giro

- Variando la tensión de alimentación del motor se varia su velocidad
- Con un modulador PWM se obtiene una tensión variable de valor: $D \cdot V_{CC}$ donde D es el ciclo de trabajo de la señal PWM

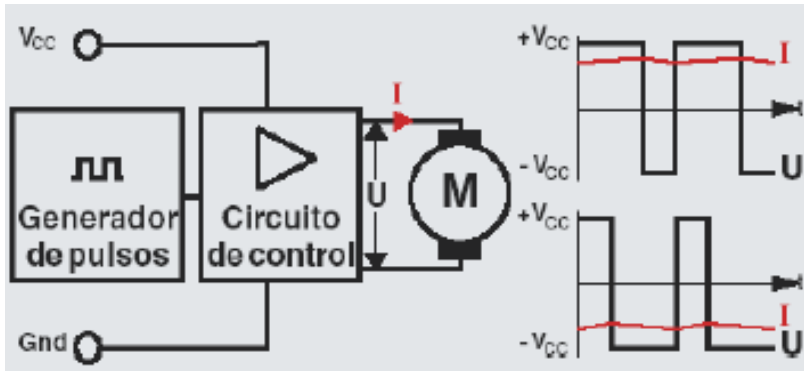
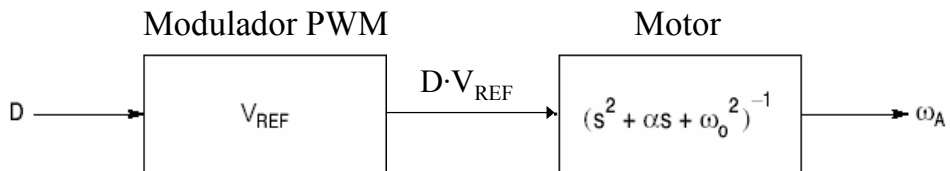
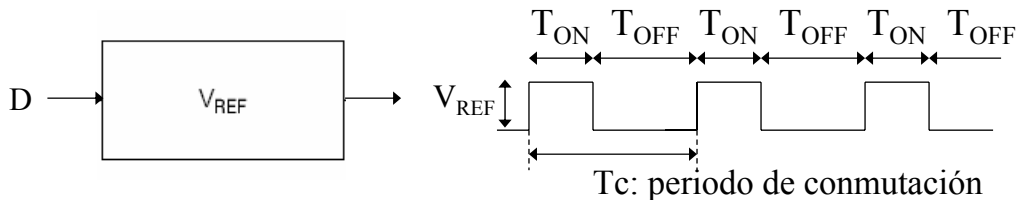


Diagrama de bloques del sistema de control en lazo abierto



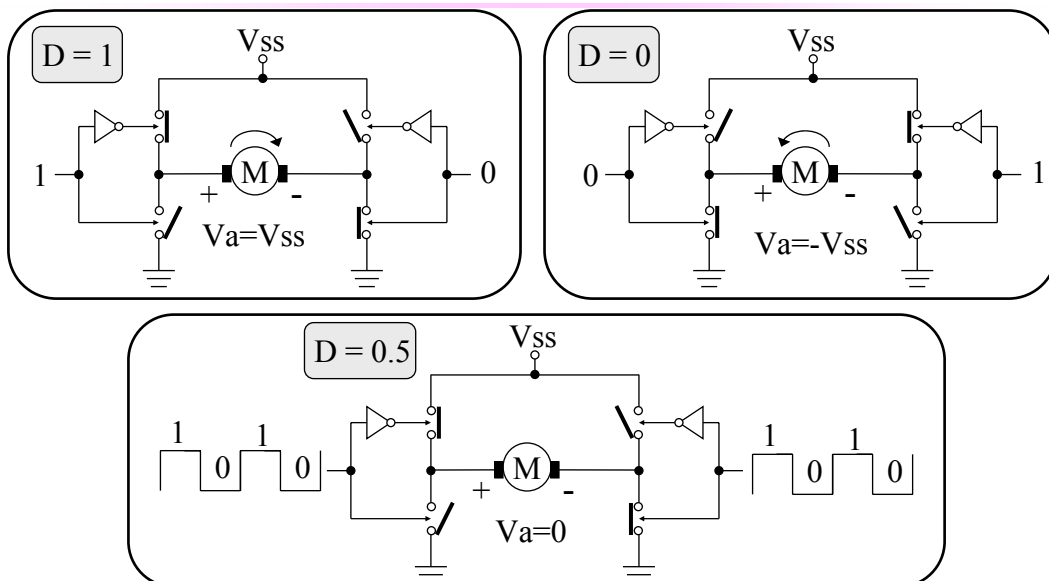
Variando el ciclo de trabajo D se varia la tensión de alimentación del motor y con ello la velocidad de giro ω_A

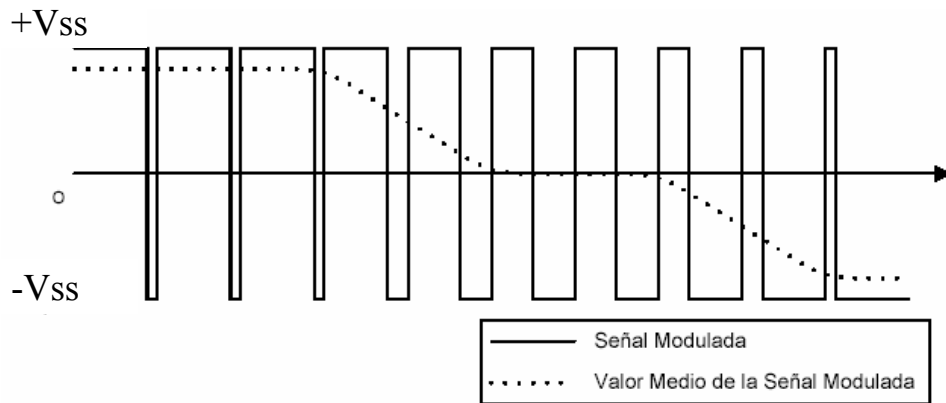
$$\omega_A = D \cdot V_{REF} \cdot \frac{1}{s^2 + 2\alpha s + \omega_o^2}$$



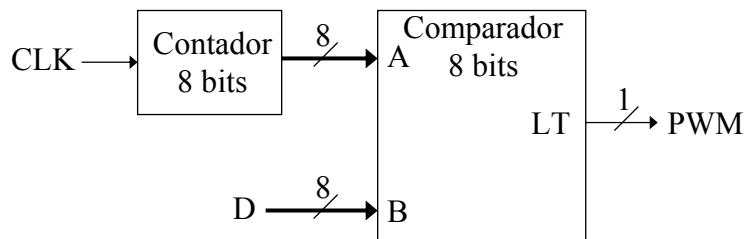
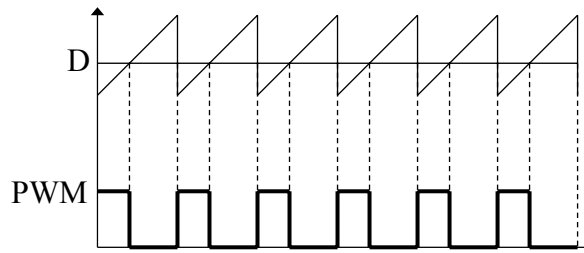
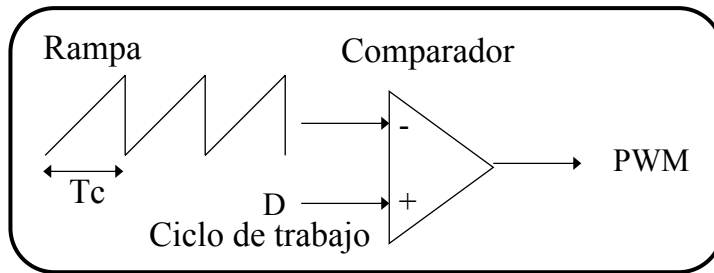
$$D = T_{ON} / T_c \text{ con } 0 \leq D \leq 1$$

- Si el T_c es suficientemente pequeño la tensión equivalente que se aplica al motor es el valor medio de la señal PWM: $D \cdot V_{REF}$

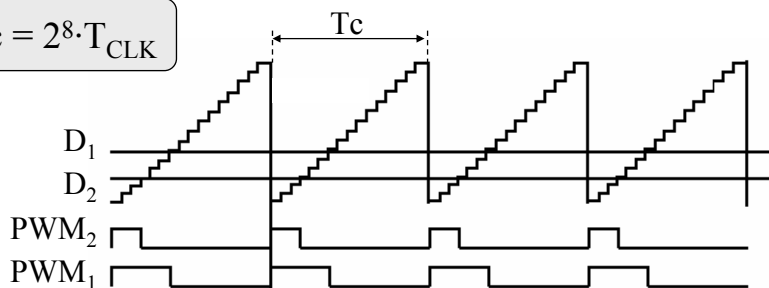


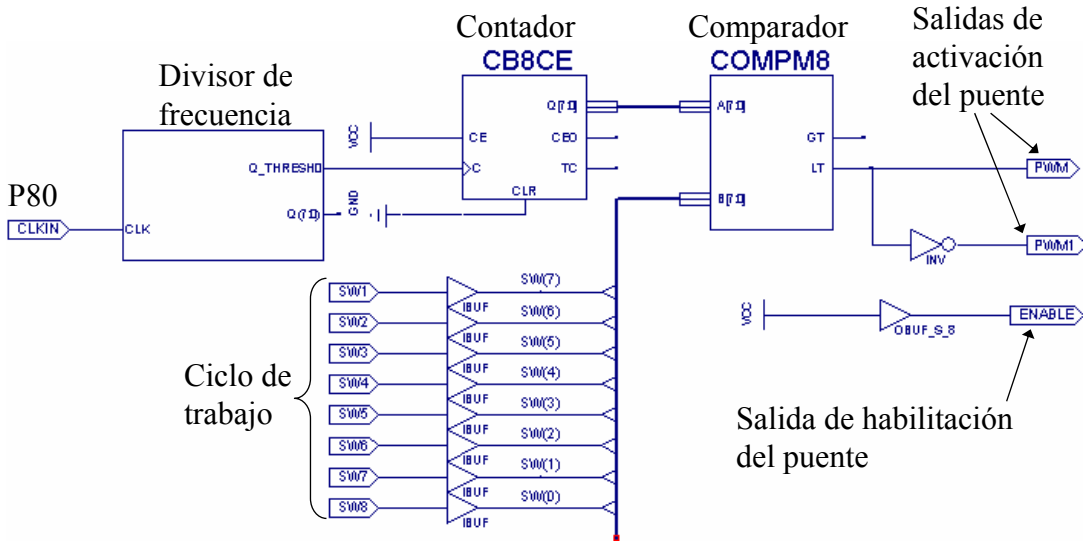


6.4.5 PWM digital

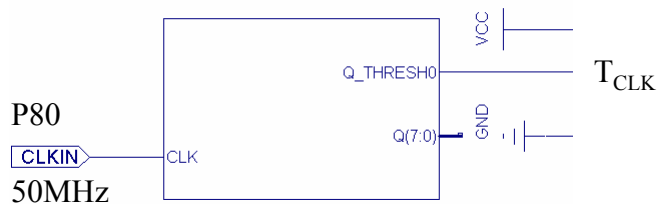


$$T_c = 2^8 \cdot T_{CLK}$$



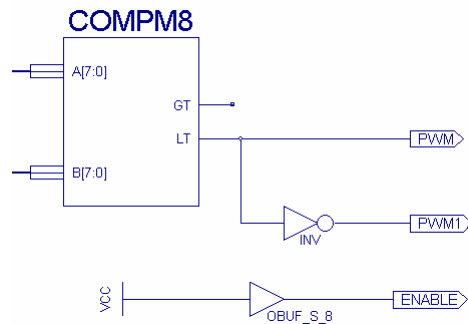
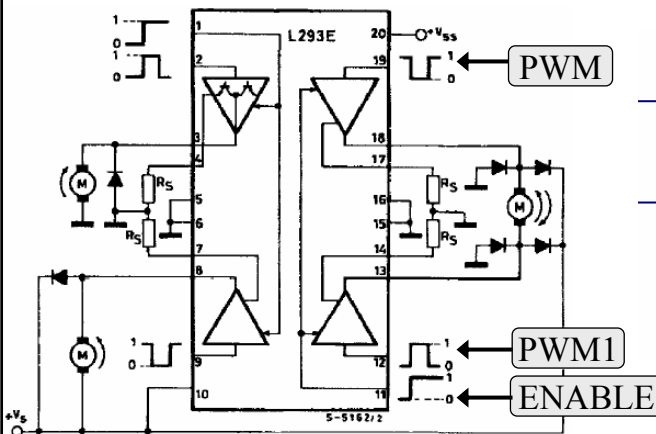
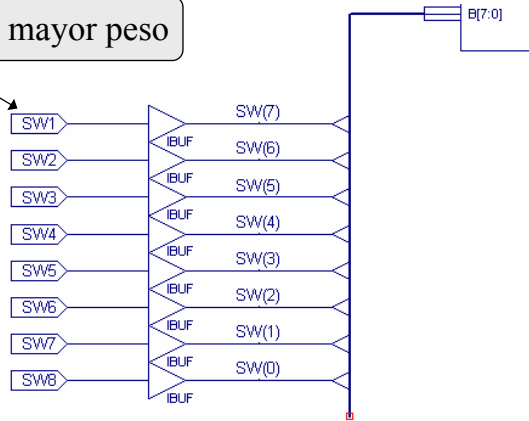


- Diseñar el divisor de frecuencia para que $T_c = 1\text{ms}$
- $T_c = 2^8 \cdot T_{\text{CLK}}$



- El valor del ciclo de trabajo se introduce mediante los switches
- La precisión es de 8 bits

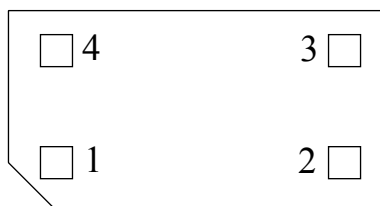
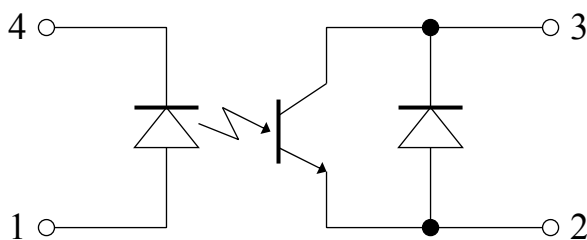
El SW1 es el bit de mayor peso



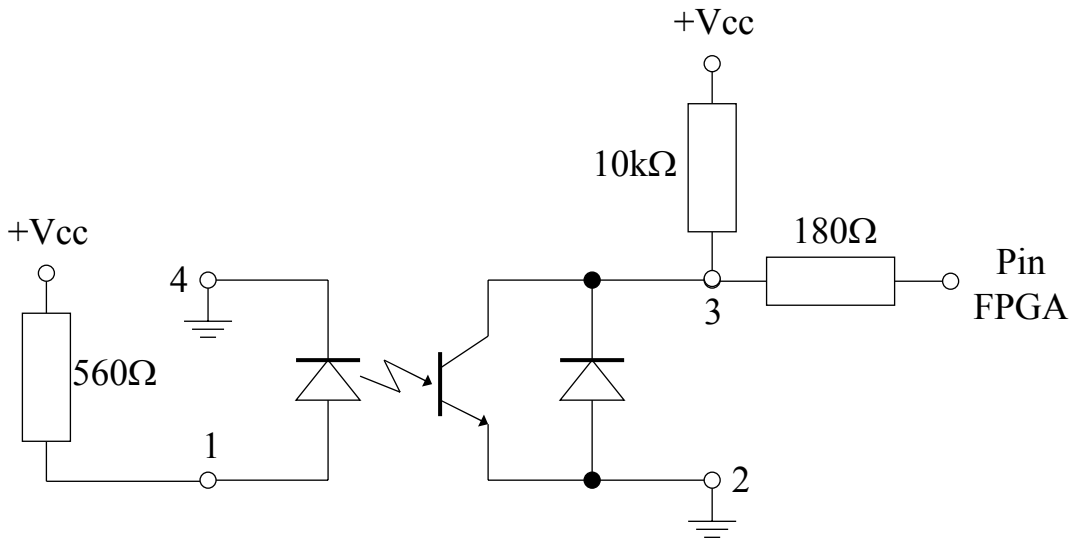
6.4.6 Monitorización de la velocidad de giro

97

Fotointerruptor



Vista inferior



La velocidad se obtiene contando el nº de pulsos por unidad de tiempo:

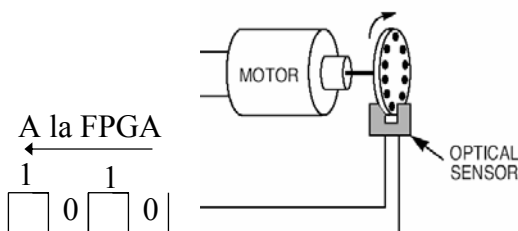
$$v_{rps} = \frac{N_p}{N_v \cdot T}$$

v_{rps} : velocidad en revoluciones por segundo

N_p : número de pulsos contados

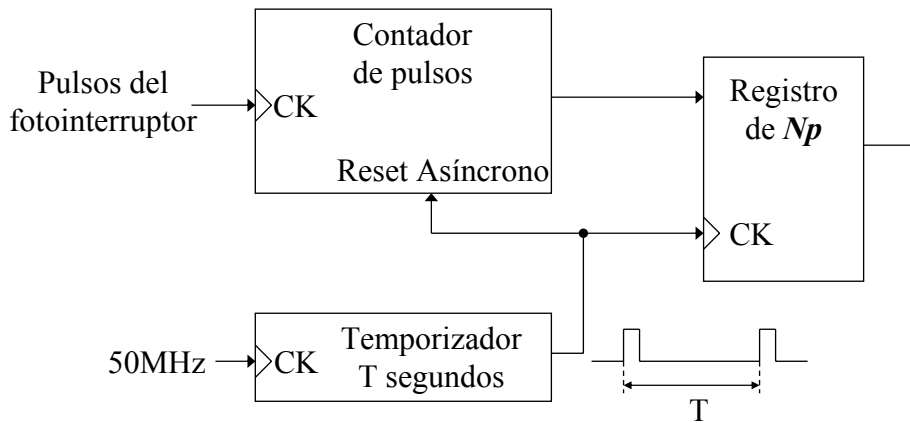
N_v : número de ventanas del disco

T : tiempo de cuenta en segundos

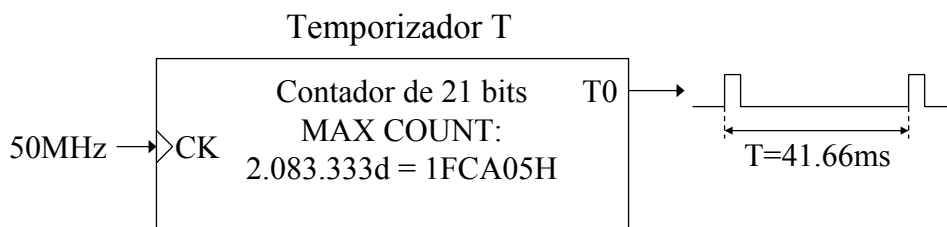


Simplificación del cálculo:

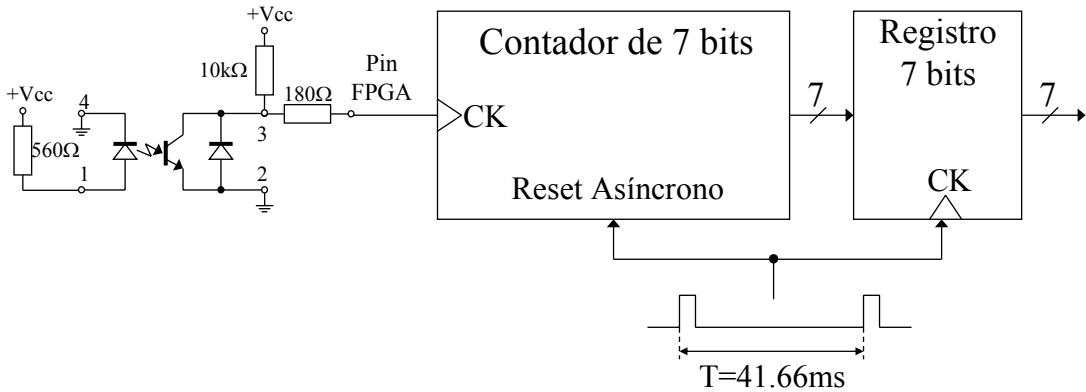
$$\text{Si } T = 1/N_v \Rightarrow v_{rps} = N_p$$



- Para $N_v = 24$ y con $T = 1/N_v \Rightarrow v_{rps} = N_p$
- El valor a temporizar es $T = 41.66\text{ms}$
- Si $f_{\text{CLK}} = 50\text{MHz}$, 41.66ms equivalen a 2.083.333 pulsos



- Si se toma como velocidad máxima $v_{rps_max} = 100d$
 - El contador debe ser de 7 bits
 - El registro también es de 7 bits

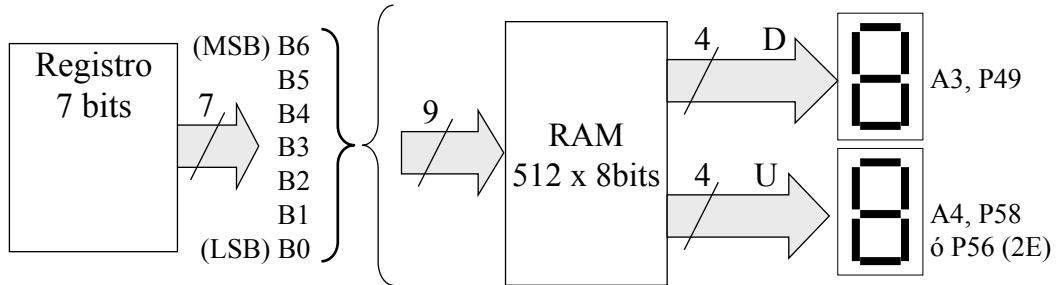


Spartan-II Device	# of Blocks	Total Block RAM Bits
XC2S15	4	16K
XC2S30	6	24K
XC2S50	8	32K
XC2S100	10	40K
XC2S150	12	48K
XC2S200	14	56K

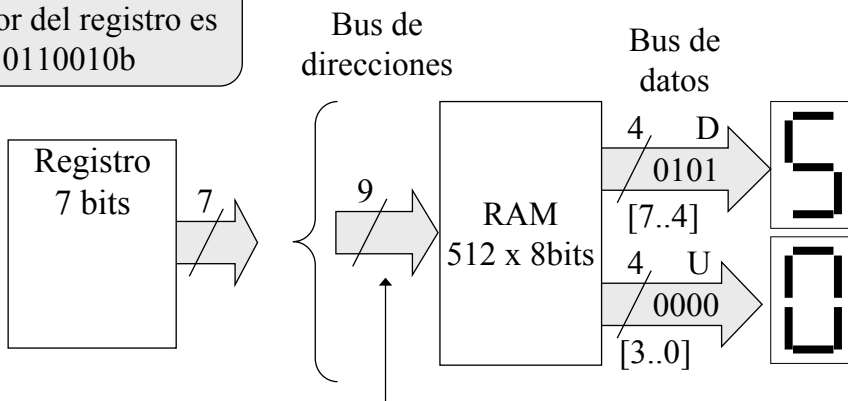
Estructura utilizada

Width	Depth	ADDR Bus	Data Bus
1	4096	ADDR<11:0>	DATA<0>
2	2048	ADDR<10:0>	DATA<1:0>
4	1024	ADDR<9:0>	DATA<3:0>
8	512	ADDR<8:0>	DATA<7:0>
16	256	ADDR<7:0>	DATA<15:0>

- Se debe visualizar valores entre 0 y 99 rps
- Los valores a visualizar se almacenan en la memoria RAM



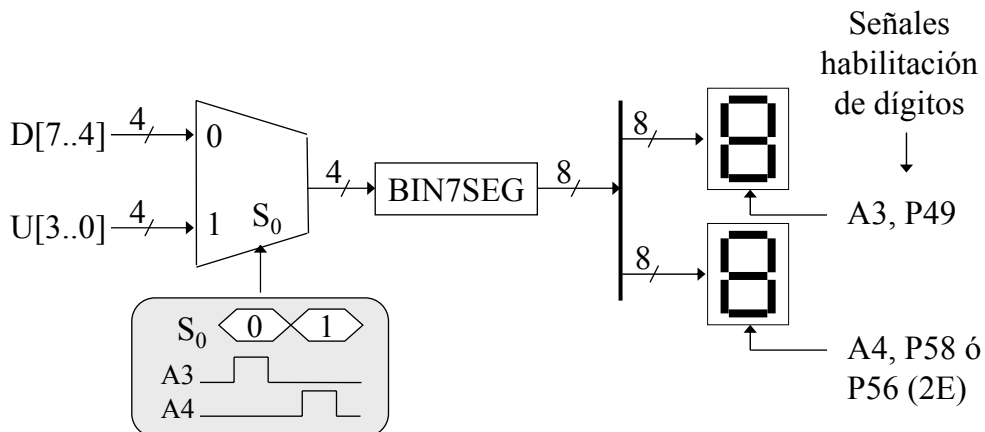
Para $N_p=50d$,
el valor del registro es
0110010b



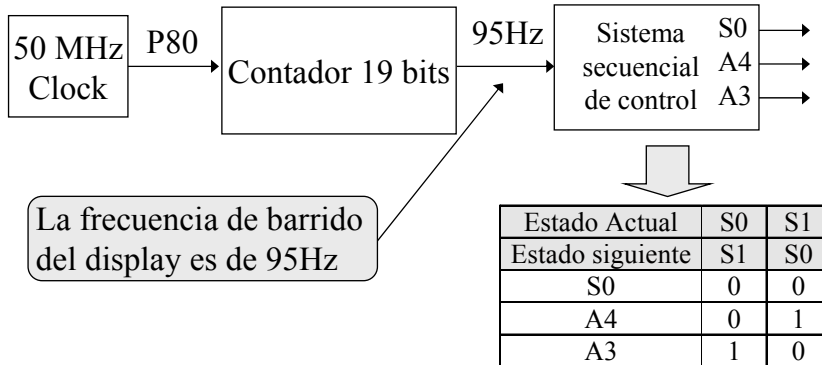
Los dos bits de mayor peso del bus de direcciones valen siempre 0

```

INIT_00=3130292827262524232221201918171615141312111009080706050403020100
INIT_01=6362616059585756555453525150494847464544434241403938373635343332
INIT_02=9594939291908988878685848382818079787776757473727170696867666564
INIT_03=00000000000000000000000000000000000000000000000000000000099989796
INIT_04=0000000000000000000000000000000000000000000000000000000000000000
INIT_05=0000000000000000000000000000000000000000000000000000000000000000
INIT_06=0000000000000000000000000000000000000000000000000000000000000000
INIT_07=0000000000000000000000000000000000000000000000000000000000000000
INIT_08=0000000000000000000000000000000000000000000000000000000000000000
INIT_09=0000000000000000000000000000000000000000000000000000000000000000
INIT_0A=0000000000000000000000000000000000000000000000000000000000000000
INIT_0B=0000000000000000000000000000000000000000000000000000000000000000
INIT_0C=0000000000000000000000000000000000000000000000000000000000000000
INIT_0D=0000000000000000000000000000000000000000000000000000000000000000
INIT_0E=0000000000000000000000000000000000000000000000000000000000000000
INIT_0F=0000000000000000000000000000000000000000000000000000000000000000
  
```

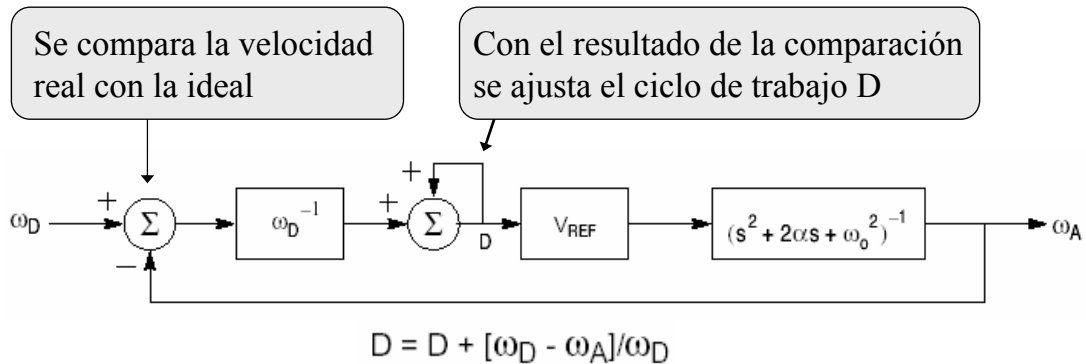


El control de activación se puede realizar mediante un sistema secuencial



6.4.7 Regulación de la velocidad de giro

El sistema de control ajusta el valor de D para que la velocidad real, w_a se mantenga igual a la velocidad deseada w_d



www.ti.com

- **Application Report SPRA083:** "Implementation of PID and Deadbeat Controllers with the TMS320 Family," Irfan Ahmed
- **Application Report SPRA009:** "Control System Compensation and Implementation with the TMS32010," Charles Slivinsky



www.motorola.com

- **AN1213/D:** "16-Bit DSP Servo Control With the MC68HC16Z" David Wilson
- **AN1249/D:** *Bruhed DC Motor Control Using the MC68HC16Z1*
- **AN1712:** "Get Your Motor Running" with the MC68HC708MP16

- *Katsuhiko Ogata, “Discrete-Time Control Systems,” Ed. Prentice Hall*



- *L293B/L293E Data Sheet*