

Construcción de un Controlador Digital PID para Enseñanza Empleando un Dispositivo Chipbas8

A. Salvá, V. Sánchez, S. Ortega, L. Altamirano
Facultad de Ingeniería-UNAM
DF 04510, México
salva@dctrl.fi-b.unam.mx
Teléfono: (52)-55-56223108

Resumen—Los controladores de tipo PID tienen un amplio espectro de aplicación en la industria. En este trabajo se presenta una realización física de los mismos, empleando para ello un dispositivo Chipbas8. Estos son microcontroladores de las familias 68HC908 y MC9S08 de FREESCALE, a los que se les ha agregado software de base que habilita la posibilidad de ser programados, para fines de diseño y prueba de aplicaciones, tanto en lenguaje ensamblador como en BASIC. Este proceso se efectúa desde una computadora (PC), mediante un software manejador denominado PUMMA08+, el cual incluye: un ensamblador cruzado denominado ENS08 y un compilador cruzado de BASIC estructurado denominado MINIBAS8A. PUMMA08+ también incorpora facilidades para cargar y depurar programas ejecutables en el dispositivo Chipbas8 empleado. Dependiendo del tipo de microcontrolador empleado, la denominación de los dispositivos aquí mencionados contendrá un posfijo que denota el miembro específico de la familia HC(S)08 implicado. Para este trabajo, se usó el dispositivo Chipbas8gp32, que está basado en el MCU 68HC908GP32 de FREESCALE. El concepto Chipbas8, el software de base, el software PUMMA08+, el ensamblador ENS08 y el compilador MINIBAS8A fueron desarrollados en el Departamento de Control y Robótica de la Facultad de Ingeniería de la UNAM.

Palabras clave: Controlador PID, Sistemas de Lazo Cerrado, Ecuación en Diferencias, Control Digital, Control Basado en una Microcomputadora.

INTRODUCCIÓN

Como es bien conocido en el medio, para un controlador de tipo PID presente en un sistema de lazo cerrado, como el mostrado en la figura 1, la relación entre las señales $u(t)$ y $e(t)$ está dada por la siguiente ecuación genérica:

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \dots\dots\dots (1)$$

Donde dependiendo de los valores relativos de las constantes K_p , K_i y K_d , el controlador podrá ser de tipo P, PI, PD o bien PID.

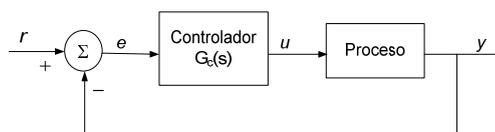


Figura 1. Sistema genérico de lazo cerrado con control PID

La función de transferencia $G_c(s) = U(s)/E(s)$ asociada con la expresión (1) está dada por:

$$G_c(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) \dots\dots\dots (2)$$

donde $T_i = K_p/K_i$ y $T_d = K_d/K_p$

Para realizar físicamente un controlador PID mediante el empleo de un microcontrolador, fundamentalmente se requiere lo siguiente:

- Obtener la función de transferencia (FT) $H_c(z)$ que resulta al discretizar la FT $G_c(s)$ dada en (2). Para ello se emplea un mapeo adecuado entre el plano S y el plano Z; esto bajo un periodo de muestreo (T) coherente con las características de temporalidad de la planta a controlar.
- Desarrollar una rutina, ejecutable en el microcontrolador empleado, que realice la ecuación en diferencias asociada con la FT $H_c(z)$ obtenida en el paso anterior. Para fines notacionales, en este trabajo a esta rutina se le denomina hcz2.
- Con el sistema de temporización propio del microcontrolador empleado, configurar éste de modo que la rutina hcz2 sea invocada a intervalos iguales al periodo de muestreo usado.
- Diseñar y construir físicamente el adecuador de entrada requerido. Para esto se emplea un circuito analógico que transforme el rango de la variable de entrada $e(t)$ que es bipolar, al rango unipolar presente típicamente en los convertidores analógico-digital propios de los microcontroladores comerciales.
- Como interfaz de salida se emplea un convertidor digital analógico, el cual se conecta a un puerto de salida binario del microcontrolador empleado. Por sus siglas en inglés, en este trabajo a este bloque funcional se le denomina DAC.
- Implementar un filtro analógico antialias, el cual debe ser de tipo paso bajas, de modo que componentes espúreas, con frecuencias mayores que la mitad de la frecuencia de muestreo sean eliminadas. Este bloque funcional debe filtrar la señal analógica a procesar por el microcontrolador antes de que ésta sea discretizada.

En las secciones siguientes se detalla cómo se desarrollaron los bloques más relevantes, mencionados en los párrafos anteriores, empleando las herramientas de software y hardware propias de un dispositivo Chipbas8gp32.

II. DISCRETIZACIÓN DE $G_c(s)$

El proceso de discretización de la FT del controlador consiste básicamente en aplicar a la ecuación (1) una aproximación numérica para la derivada e integral implicadas. Lo anterior conduce a una ecuación en diferencias que representa la relación entrada salida de un sistema discreto, que aproxima al sistema analógico cuya relación entrada salida está dada por la ecuación (1). A la FT del sistema discreto mencionado se le denomina en este trabajo como $H_c(z)$. Las aproximaciones numéricas de la derivada y la integral pueden ser, entre otras, de tipo rectangular, o bien, de tipo trapezoidal. Como se sabe, la FT $H_c(z)$ puede obtenerse, dependiendo del tipo de aproximación empleada para la derivada e integral, mediante una de las siguientes dos ecuaciones:

Para la aproximación rectangular:

$$H_c(z) = G_c \left(\frac{1-z^{-1}}{T} \right) \dots \dots \dots (3)$$

Para la aproximación trapezoidal:

$$H_c(z) = G_c \left(\frac{2}{T} \cdot \frac{1-z^{-1}}{1+z^{-1}} \right) \dots \dots \dots (4)$$

Donde T es el periodo de muestreo empleado.

Dependiendo del hecho de que el controlador sea de tipo PID, PI, PD o P, y de acuerdo con el uso de las ecuaciones (3) ó (4) empleadas para la obtención de $H_c(z)$, ésta presentará las formas que se describen a continuación:

$H_c(z)$ para un controlador PID

En este caso $H_c(z)$ tiene la siguiente forma genérica:

$$H_c(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \dots \dots \dots (5)$$

Empleando la ecuación (3) para la obtención de la FT $H_c(z)$, los coeficientes del denominador y el numerador son:

$$a_1 = -1 \dots \dots \dots (6)$$

$$a_2 = 0 \dots \dots \dots (7)$$

$$b_0 = K_p \left(1 + \frac{T}{T_i} + \frac{Td}{T} \right) \dots \dots \dots (8)$$

$$b_1 = -K_p \left(1 + 2 \frac{Td}{T} \right) \dots \dots \dots (9)$$

$$b_2 = K_p \frac{Td}{T} \dots \dots \dots (10)$$

Si se emplea la ecuación (4) para la obtención de $H_c(z)$, los coeficientes del numerador y denominador son:

$$a_1 = 0 \dots \dots \dots (11)$$

$$a_2 = -1 \dots \dots \dots (12)$$

$$b_0 = K_p \left(1 + \frac{T}{2T_i} + 2 \frac{Td}{T} \right) \dots \dots \dots (13)$$

$$b_1 = K_p \left(\frac{T}{T_i} - 4 \frac{Td}{T} \right) \dots \dots \dots (14)$$

$$b_2 = K_p \left(\frac{T}{2T_i} + 2 \frac{Td}{T} - 1 \right) \dots \dots \dots (15)$$

$H_c(z)$ para un controlador PI

En este caso, $H_c(z)$ presenta la siguiente forma:

$$H_c(z) = \frac{b_0 + b_1 z^{-1}}{1 + a_1 z^{-1}} \dots \dots \dots (16)$$

Si se usa la ecuación (3) para la obtención de la FT $H_c(z)$, los coeficientes del denominador y el numerador son:

$$a_1 = -1 \dots \dots \dots (17)$$

$$b_0 = K_p \left(1 + \frac{T}{T_i} \right) \dots \dots \dots (18)$$

$$b_1 = -K_p \dots \dots \dots (19)$$

Si se emplea la ecuación 4 para la obtención de la FT $H_c(z)$, los coeficientes del denominador y el numerador son:

$$a_1 = -1 \dots \dots \dots (20)$$

$$b_0 = K_p \left(1 + \frac{T}{2T_i} \right) \dots \dots \dots (21)$$

$$b_1 = K_p \left(\frac{T}{2T_i} - 1 \right) \dots \dots \dots (22)$$

$H_c(z)$ para un controlador PD

En este caso la forma de $H_c(z)$ también puede ser descrita por la ecuación (16). Si se usa la ecuación (3) para la obtención de la FT $H_c(z)$, los coeficientes del denominador y el numerador son:

$$a_1 = 0 \dots \dots \dots (23)$$

$$b_0 = K_p \left(1 + \frac{Td}{T} \right) \dots \dots \dots (24)$$

$$b_1 = -Kp \frac{T_d}{T} \dots\dots\dots (25)$$

En caso de que se emplee la ecuación (4) para la obtención de $Hc(z)$, los coeficientes del numerador y denominador son:

$$a_1 = 1 \dots\dots\dots (26)$$

$$b_0 = Kp(1 + \frac{2T_d}{T}) \dots\dots\dots (27)$$

$$b_1 = Kp(1 - \frac{2T_d}{T}) \dots\dots\dots (28)$$

$Hc(z)$ para un controlador P

En este caso, el controlador es un sistema algebraico y es fácil ver que el coeficiente b_0 de la FT $Hc(z)$ es simplemente el valor de Kp , por lo que

$$Hc(z) = Kp \dots\dots\dots (29)$$

III. REALIZACIÓN CON UN DISPOSITIVO CHIPBAS8GP32

Para realizar físicamente el controlador, fundamentalmente se requiere que la computadora digital que realice esta función, calcule en tiempo real el valor presente de la salida del controlador a partir del valor presente de la entrada al mismo y de valores anteriores, tanto de la entrada como de la salida. Esto debe hacerse a intervalos iguales al periodo de muestreo T . La fórmula de recurrencia correspondiente es simplemente la ecuación en diferencias asociada con la FT $Hc(z)$ que, de acuerdo con el tipo de controlador implicado, podrá ser la expresada por las ecuaciones (5), (16) y (29), para un controlador PID, PI ó PD y P, respectivamente. Es claro que los cálculos implicados, junto con la conversión analógica-digital requerida y la escritura binaria al puerto del MCU ligado a la entrada del DAC, deberán efectuarse en un tiempo sensiblemente menor al periodo de muestreo.

MINICON08A, que está basada en un dispositivo Chipbas8gp32. Se supone que el rango de valores que puede tomar la variable $e(t)$ está comprendido entre -10 y +10 volts. Mediante el bloque de adecuación de entrada (AE), este rango se transforma a otro que va de 0 a 5 volts, el cual es propio de los canales de entrada del convertidor analógico-digital (CAD) presente en el MCU. El AE se realizó empleando un circuito analógico basado en dos amplificadores operacionales.

Ecuaciones de recurrencia

Las ecuaciones de recurrencia varían con el tipo de controlador y tipo de discretización empleada. Éstas se detallan a continuación:

Para el controlador PID la ecuación de recurrencia es:

$$up = -a_1u1 - a_2u2 + b_0ep + b_1e1 + b_2e2 \dots\dots\dots (30)$$

Donde los coeficientes a_1, a_2, b_0, b_1 y b_2 están dados por las ecuaciones (6), (7), (8), (9) y (10) en caso que se haya empleado la ecuación (3) para la obtención de $Hc(z)$. Si para este fin se empleó la ecuación (4) los coeficientes mencionados están dados por las ecuaciones (11), (12), (13), (14) y (15). Las variables $up, u1, u2, ep, e1$ y $e2$ representan valores presentes y anteriores de la entrada y salida del controlador; esto es:

$$up = u(nT) \dots\dots\dots(31)$$

$$u1 = u((n-1)T) \dots\dots\dots(32)$$

$$u2 = u((n-2)T) \dots\dots\dots(33)$$

$$ep = e(nT) \dots\dots\dots(34)$$

$$e1 = e((n-1)T) \dots\dots\dots (35)$$

$$e2 = e((n-2)T) \dots\dots\dots (36)$$

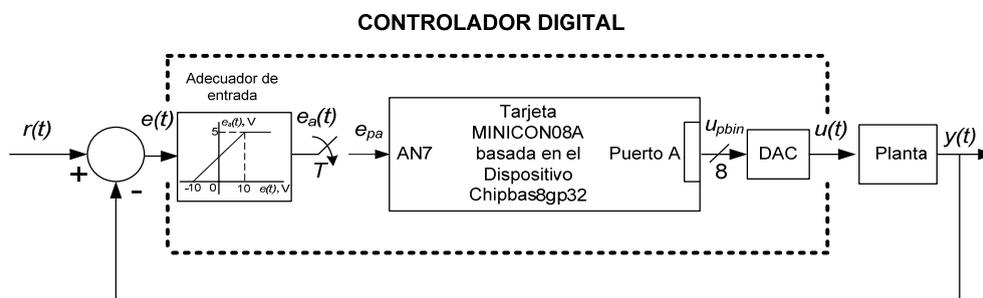


Figura 2. Sistema de control realizado con un dispositivo Chipbas8gp32

En la figura 2 se muestra un esquema a bloques de un sistema de lazo cerrado, donde el controlador es realizado en forma digital con una tarjeta de desarrollo denominada

Donde nT representa al valor presente del instante en que se trabaja con la muestra n -ésima.

Para el controlador PI o PD, la ecuación de recurrencia es:

$$up = -a_1 \cdot u1 + b_0 \cdot ep + b_1 \cdot e1 \dots\dots\dots (37)$$

Donde, para el controlador PI, los coeficientes a_1, b_0 y b_1 están dados por las ecuaciones (17), (18) y (19) en caso de que se haya empleado la ecuación (3) para la obtención de $Hc(z)$. Si para este fin se empleó la ecuación (4), los coeficientes mencionados están dados por las ecuaciones (20), (21) y (22). Para el controlador PD, los coeficientes a_1, b_0 y b_1 se expresan por medio de las ecuaciones (23), (24) y (25), en caso de que se haya empleado la ecuación (3) para la obtención de $Hc(z)$. Si para este fin se empleó la ecuación (4), los coeficientes mencionados están dados por las ecuaciones (26), (27) y (28). Las variables $up, u1, ep$ y $e1$ representan valores presentes y anteriores de la entrada y salida del controlador, y acorde con la notación propia de ecuaciones en diferencias, éstas se representan por las ecuaciones (31), (32), (34) y (35).

Para el controlador P la ecuación de recurrencia simplemente es:

$$up = Kp \cdot ep \dots\dots\dots (38)$$

Flujo de ejecución de la rutina hcz2

Como se ha venido mencionando en este trabajo, la rutina hcz2 es la que realiza propiamente el algoritmo de control implicado, y ésta ha de invocarse a intervalos iguales al periodo de muestreo. En las siguientes líneas se detalla el flujo de ejecución de la rutina hcz2. Para una mejor comprensión de éste, véase la figura 2. Dicho flujo de ejecución está integrado por los siguientes siete pasos:

1. Se invoca una rutina que hace que se efectúe la conversión analógica - digital de la señal e_{pa} retornándose en la variable entera $bycad\%$ el valor entregado por el CAD.
2. A partir de la variable $bycad\%$ se obtiene el valor presente de la señal de error $e(nT)$, asignándose éste a la variable real de precisión sencilla ep , según la siguiente ecuación:

$$ep = .0784313 \cdot bycad\% - 10 \dots\dots\dots (39)$$

3. Mediante una de las tres ecuaciones ((30), (37) ó (38)), según sea el caso, se obtiene el valor presente up de la variable de salida del controlador. Se calcula el valor de la variable entera $bydac\%$, que debe aplicarse a la entrada del DAC, para que la salida de este sea precisamente el valor up calculado en el paso 3. Para esto se emplea la siguiente ecuación:

$$bydac\% = 12.75 \cdot (up + 10) \dots\dots\dots (40)$$

4. En caso de que el valor de $bydac\%$ resulte mayor que 255, se le postasigna el valor 255. Por otro

lado, si $bydac\%$ resulta negativo, se le postasigna el valor cero. De esta forma, el valor de la salida del DAC estará acotado a un rango de valores comprendido entre -10 y 10 .

5. Se coloca el valor de la variable $bydac\%$ en el puerto del MCU cuya salida es la entrada del DAC.
6. Si el controlador es PID, se actualizan los valores anteriores de la entrada y la salida del controlador de acuerdo con las ecuaciones (41), (42), (43) y (44). Por razones obvias, si el controlador es PI o PD, esta actualización se efectúa empleando solamente a las ecuaciones (42) y (44). En caso de que el controlador sea de tipo P, es claro que este paso no es necesario.

$$u_2 = u_1 \dots\dots\dots (41)$$

$$u_1 = u_p \dots\dots\dots (42)$$

$$e_2 = e_1 \dots\dots\dots (43)$$

$$e_1 = e_p \dots\dots\dots (44)$$

7. Se retorna de la rutina hcz2

Experimentalmente se midió el tiempo que dura la ejecución (Te) de la rutina hcz2 para cada una de las acciones de control realizadas con el dispositivo chipbas8gp32. En la tabla 1 se resume esta información. Más adelante se detalla como se hizo esta medición.

| Tipo de control | Te [ms] |
|-----------------|---------|
| P | 1.2 |
| PI | 1.6 |
| PID | 2.0 |
| PD | 1.6 |

Tabla 1. Tiempo de ejecución de la rutina hcz2

De acuerdo con los valores de Te mostrados en la tabla 1 se asumió conservadoramente una cota inferior para el periodo de muestreo del orden de 10 [ms].

Para lograr que la rutina hcz2 se invoque a intervalos iguales al periodo de muestreo, en la fase de inicialización del software ejecutable en el microcontrolador del dispositivo Chipbas8gp32 se configura el temporizador de éste, de modo que se genere por medio del evento de sobreflujo una interrupción periódica a intervalos iguales al periodo de muestreo, y desde la rutina de servicio de interrupción asociada se invoca la rutina hcz2. El código fuente de ésta se muestra a continuación.

Código fuente de la rutina hcz2

```

hcz2:
  iniens
  bset 0, ptb 'ptb0<--1, testigo de inicio de ejecución
  finens

  gosub convad 'Invocación del CAD (paso 1)

  ep=.0784313*bycad%-10. 'Obtención de ep (paso 2)
  ----- (paso 3) -----
  
```

```
up = -a1*u1 -a2*u2 + b0*ep+b1*e1+b2*e2
bydac% = 12.75*(up+10.)
```

------(paso 4)-----

```
if bydac%<0 then
bydac%=0
endif
```

```
if bydac%>255 then
bydac%=255
endif
```

------(paso 5)-----

```
iniens
lda bydac%+1
sta pta@
finens
```

------(paso 6)-----

```
u2 = u1
u1 = up
e2 = e1
e1 = ep
```

```
iniens
bclr 0,ptb 'ptb0<--0, testifica fin de ejecución
finens
```

------(paso 7)-----

```
return
```

Determinación experimental de T_e para la rutina hcz2

El tiempo de ejecución de la rutina hcz2 se midió colocando un nivel alto en el bit ptb0 del MCU al inicio de ésta, regresándose este testigo a nivel bajo justo antes de la instrucción de retorno. Véase el código fuente inmediato anterior. De esta manera, la forma de onda en ptb0 será una señal cuadrada con periodo igual al tiempo de muestreo T y con duración en nivel alto muy aproximada al tiempo T_e que dura la ejecución de la rutina hcz2. En la figura 3 se muestra la señal en ptb0 cuando se realiza un control PI bajo la discretización rectangular. Se aprecia que el tiempo de ejecución T_e es del orden de 1.6 [ms].

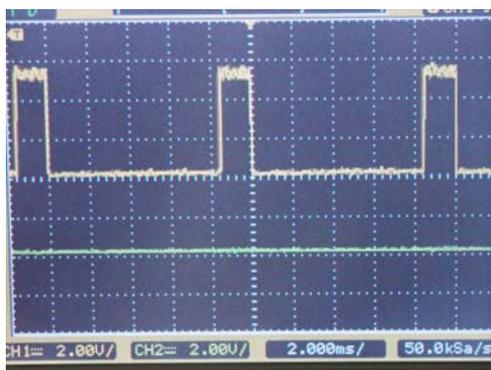


Figura 3. Señal testigo en ptb0 para control PI con discretización rectangular, ($T=10$ [ms])

Circuito restador y generador local de referencia

Para implementar el restador que aparece en la figura 1 se empleó el circuito mostrado en la figura 4. Éste está basado en un amplificador operacional configurado como restador.

El interruptor S1 permite que la señal de referencia sea local o remota. En este último caso la señal de referencia es proporcionada al dispositivo chipbas8gp32 desde otra computadora ligada con éste de alguna forma.

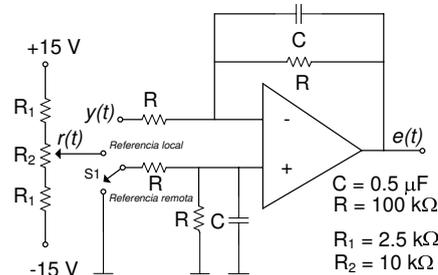


Figura 4. Circuito restador y de generación local de señal de referencia $r(t)$.

En el dominio de Laplace, la relación entrada-salida del circuito mostrado en la figura 4 es la siguiente:

$$\frac{E(S)}{D(S)} = \frac{1}{RCS + 1} \dots\dots\dots (45)$$

Cuando el interruptor S1 está en la posición de referencia local, la señal $d(t)$ es la diferencia entre la señal de referencia local $r(t)$ y la señal $y(t)$ que representa a la variable a controlar. Para S1 en la posición de referencia remota, la señal $e(t)$ es simplemente la señal $y(t)$ con el signo invertido, efectuándose por software en el MCU la suma requerida para la obtención de la señal de error verdadera.

De acuerdo con la ecuación (45), claramente se aprecia un efecto de filtrado analógico paso bajas que realiza el filtro antialias requerido, para fines del procesamiento digital implicado en el controlador digital descrito en este trabajo. Para los valores numéricos de los elementos del restador, la frecuencia de corte del filtro es 3.18 [Hz], lo que para un periodo de muestreo de 10 [ms] implica que para frecuencias mayores o iguales a la mitad de la frecuencia de muestreo, la atenuación del filtro antialias sea mayor que 23.9 dB.

IV RESULTADOS PRÁCTICOS

La funcionalidad del controlador implementado se probó experimentalmente para cada una de las modalidades de control contempladas (P, PI, PD y PID). Para esto se utilizó la planta de segundo orden que se muestra en la figura 5.

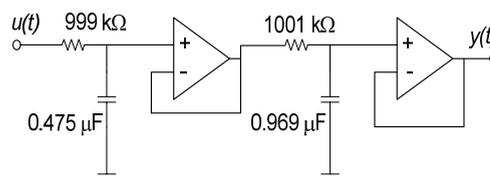


Figura 5. Planta de segundo orden

La FT $Gp(s)$ de la planta es:

$$G_p(S) = \frac{1}{(1+.474S)(1+.97S)} \dots\dots\dots (46)$$

Para fines ilustrativos, aquí se muestran resultados experimentales para el caso de las modalidades PI y P. Para el modo PI, usando el primer método de Ziegler Nichols, los valores de las constantes K_p y T_i son 9.36 y 0.616 [s] respectivamente (Ogata, 2010) y (Verde, 2001). En la figura 6 se muestra una fotografía donde se aprecia la respuesta del sistema de lazo cerrado de la figura 2 para un control PI de la planta empleada bajo una señal de referencia local cuadrada.

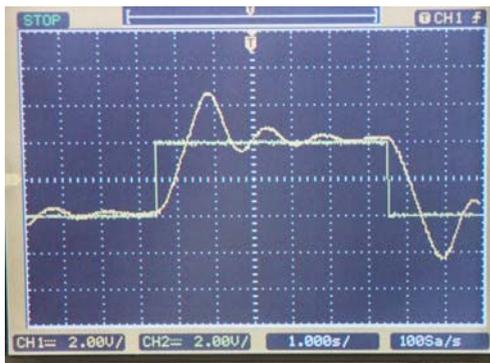


Figura 6. Respuesta a escalón de lazo cerrado para control PI ($K_p = 9.36$, $T_i = .616$ [s], $T = 10$ [ms])

Para el modo P, aplicando nuevamente el primer método de Ziegler Nichols, se obtuvo $K_p = 10.4$ (Ogata, 2010) y (Verde, 2001). En la figura 7 se muestra una fotografía donde se aprecia la respuesta del sistema de lazo cerrado de la figura 2 para un control P de la planta empleada bajo una señal de referencia local cuadrada.

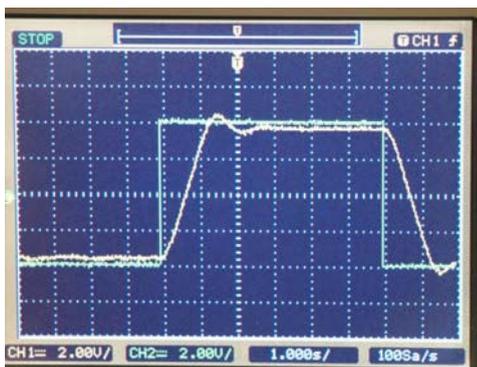


Figura 7. Respuesta a escalón de lazo cerrado para control P ($K_p = 10.4$, $T = 10$ [ms])

Para más detalles acerca de las herramientas de software y hardware empleadas en el diseño y construcción del controlador PID aquí descrito, pueden verse las referencias (Salvá y Altamirano, 2009) y (Salvá, 2011).

V. CONCLUSIONES

Los controladores PID tienen un amplio espectro de aplicaciones en la industria; y de hecho existen diversos fabricantes en el mercado basados en hardware analógico o

digital. En el caso de estos últimos, hay controladores basados en algún tipo de microcontrolador o microprocesador, que junto con algunos otros componentes, integran al controlador como una caja negra. Otra forma de realización de los controladores PID consiste en emplear un módulo de herramientas de instrumentación virtual como lo es LabVIEW de National Instruments. Sin embargo, aún cuando en campo muchas acciones de desarrollo de ingeniería se efectúan integrando bloques funcionales ya probados, desde el punto de vista de la enseñanza de la Ingeniería de Control es muy conveniente que los estudiantes conozcan lo que hay detrás de las cajas negras que realizan los controladores PID.

En cuanto al lenguaje de programación empleado para validar el controlador aquí presentado, éste fue un BASIC estructurado validado por el compilador cruzado MINIBAS8A. Si bien es cierto que el lenguaje C es más popular para la programación de aplicaciones basadas en microcontrolador, las facilidades del compilador MINIBAS8A fueron adecuadas para la realización del controlador PID aquí presentado. El programa fuente completo no se incluyó aquí por razones de espacio. El desarrollo aquí presentado se basó en el Chipbas8gp32, validado por el MCU 68HC908GP32 que ha estado presente en la industria por un largo tiempo. Trasladar el diseño a otra versión del Chipbas8 basada en un MCU más reciente de la familia HC(S)08, no presenta grandes dificultades. Tal es el caso del Chipbas8gt60, que está basado en el MCU MC9S08GT60.

Finalmente se debe enfatizar que el procedimiento usado en la realización de este trabajo, tiene un alto valor agregado desde el punto de vista didáctico, porque permite que el alumno opere puntualmente cada uno de los componentes de un sistema de control digital completo, tanto en hardware como en software. Además, este sistema permite la validación de otros algoritmos de control diferentes al PID clásico, verbigracia control por modos deslizantes o cancelación de polos y ceros, entre otros.

REFERENCIAS

Åström, K. y T. Hägglund (1995). PID Controllers: Theory, Design, and Tuning. ISA.

Ogata, K. (2010). INGENIERÍA DE CONTROL MODERNA. Editorial Pearson.

Salvá, A y L. Altamirano (2009). DISPOSITIVOS CHIPBAS8, MICROCONTROLADORES HC08 PROGRAMABLES EN LENGUAJE BASIC. Memoria del Simposio Anual de Automatización, Electrónica e Instrumentación, (SAAEI 2009), celebrado en julio de 2009 en la Universidad Carlos III, en Madrid, España. El artículo puede verse en el archivo chipbas8_art_saaei09_def.pdf descargable desde <http://dctrl.fi-b.unam.mx/~salva>.

Salvá, A (2011). Manual de usuario del sistema AIDA08. Archivo muaida08ve2011.pdf, descargable desde <http://dctrl.fi-b.unam.mx/~salva>

Verde, C. y R. Carrera (2001). EJERCICIOS RESUELTOS DE CONTROL ANALÓGICO. Editorial Trillas.