

¿El uso de microcontroladores de 8-bits es una opción práctica en sistemas de control automático relativamente complejos?

Bautista-Quintero Ricardo¹, Domínguez Rodríguez Ricardo Ernesto¹ y Domínguez Lozoya José Carlos¹

¹Juan de Dios Bátiz S/N, Col. Guadalupe, Culiacán, Sinaloa, México, C.P. 80220.
rbautista@itculiacan.edu.mx, rdoming72@hotmail.com, jcdguez@hotmail.com

Teléfono: 01(667)7-13-38-04, Fax 01(667)7-16-96-47

Resumen—El uso de tarjetas de adquisición y control de datos de alta capacidad de memoria y desempeño del CPU ha simplificado la integración de sistemas de control automático. Las interfaces visuales incorporadas en dichas tarjetas permiten programación de alto nivel reduciendo los tiempos de configuración y ajuste de los algoritmos de control. Sin embargo, el alto costo y la portabilidad para algunas aplicaciones hacen impráctico el uso de dichas tarjetas. Paralelamente los microcontroladores de 8-bits usados en aplicaciones de control (sistema de control embebido o ECS) desde su aparición en el mercado han incrementado su capacidad y su costo ha caído dramáticamente. A pesar de estas atractivas características relacionadas a su costo, investigadores y desarrolladores prefieren sistemas comerciales mas sofisticados. En este contexto, el presente trabajo ejemplifica con un sistema de control basado en un microcontrolador de 8-bits de fácil acceso en México como puede ser una opción práctica para controlar plantas de relativa complejidad (no-lineal SIMO).

Para fines de comparación, el sistema es implementado también con dispositivos de mayor capacidad computacional, (microcontrolador industrial de 32 bits y una tarjeta de control DS1104). Como conclusión se da una respuesta a la pregunta planteada en el título de este artículo.

Palabras clave: computadora de control, algoritmo de control, sistema embebido, no-lineal, SIMO.

I. INTRODUCCIÓN

La construcción de sistemas de auto-regulación basados en retroalimentación –mejor conocido como control automático– (Lewis, 1992) tiene una larga historia, sin embargo desde la aparición de las computadoras digitales y particularmente las de propósito de control específico (computadoras embebidas para control o ECS) en los años 70s han revolucionado el modo de vivir de la humanidad. Las aplicaciones se han extendido a un nivel exponencial, de hecho el 98% de las computadoras digitales en el mundo son usados en sistemas embebidos (Gaetano y Roy, 2000) y existen en promedio 3 de estas computadoras por cada ser humano (WSTS, 2005).

La complejidad y rápido crecimiento del hardware de los ECS crece basado en la predicción propuesta por Gordon Moore en 1979 (las computadoras duplicaran su desempeño cada 2 años aproximadamente) (Moore, 1979), sin embargo

el software desarrollado para las computadoras embebidas ha quedado rezagado y no ha alcanzado el nivel de complejidad para ser usado eficientemente con el hardware existente (Napper, 1998). Esto genera una disyuntiva para el desarrollador entre emplear una computadora de propósito general o un ECS para estabilizar plantas de relativa complejidad.

El desarrollo de tarjetas de adquisición y control que son compatibles con computadoras de propósito general han creado un paradigma atractivo para el desarrollador que diseña e implementa sistemas de control automático (Mrad et al., 2000). Las interfaces visuales incorporadas en dichas tarjetas permiten programación de alto nivel reduciendo el de configuración y ajuste de los algoritmos de control. Sin embargo, en aplicaciones que demandan rentabilidad dichos sistemas resultan imprácticos. La alternativa de usar un ECS de bajo costo puede resultar una opción atractiva para muchas aplicaciones de control automático.

En este contexto, el presente artículo centra su atención en microcontroladores de 8-bits (ampliamente usados en control secuencial) para ser empleado en la adquisición de datos para la caracterización y control en tiempo-real basado en un arquitectura “time-triggered” (TT).

Se presenta un caso de estudio (robot subactuado de una entrada y varias salidas) para validar experimentalmente algunos de los conceptos mostrados y se compara con un sistema de mayor capacidad computacional con el fin de mostrar que tan efectivos son los microcontroladores actuales de 8-bits cuando se implementan algoritmos de control multivariable en plantas no-lineales.

II. MICROCONTROLADORES DE 8-BITS USADOS PARA CONTROL AUTOMÁTICO

El control automático empleando sistemas embebidos está basado en una combinación sinérgica entre ingeniería y ciencias computacionales. El control en tiempo real “duro” es una característica fundamental que dota al algoritmo de confiabilidad en su ejecución y sincronización con los eventos de adquisición, control y actuación (Butazzo, 2005). La relativa corta historia de estos sistemas se remonta a los primeros microprocesadores de 4-bits diseñados por equipo de Federico Faggin a principios de los 70s. El propósito del primer microprocesador comercial –el de Intel 4004– reconocido como el primer procesador para sistemas embebidos (Pratt y Brown, 1975), el cual fue ampliamente



usado en aplicaciones de control secuencial y cálculos básicos para calculadoras, impresoras y otros aparatos de oficina que no incorporaban técnicas de control automático. Sin embargo, según (Faggin et al. 1996) el 4004 estaba planeado para aplicaciones de control automático. Las 94,000 instrucciones que ejecutaba el 4004 (casi 5000 veces más lento que un procesador moderno) no permitían resolver problemas de control para muchas aplicaciones. La inclusión de varios 4004 para lograr mayor capacidad de procesamiento fue lograda hasta mediados de los 70s y fue basada en un desarrollo previo de intel (Goksel et al., 1975). Los microcontroladores de mayor capacidad como 8008 de 8-bits se perfilaban como los dispositivos que sustituirían los tradicionales controles secuenciales con complejas redes de alambres y dispositivos de memoria basados en relevadores. La era de los PLCs que por su capacidad a soportar un amplio rango de temperatura e inmunidad al ruido los convirtieron en elementos de control por excelencia para la industria (Erickson, 1996).

La combinación de los avances de hardware de 8-bits con el desarrollo de los lenguajes de programación (especialmente C), los depuradores así como los algoritmos de despacho de tareas los hicieron cada vez más prácticos y accesibles (Donaghey, 1976).

A finales de los 70s las aplicaciones de control más avanzado fueron apareciendo con este incremento en el desempeño, por ejemplo en el control auto-contenido de misiles (Mersten, 1979), prótesis de rehabilitación y sustitución de miembros superiores (Graupe et al., 1978) y el control de herramientas de alta precisión (Lamb y Whelan, 1976). En estas aplicaciones la inclusión de sistemas embebidos fue la única opción práctica en términos de espacio y portabilidad que sustituyó al control analógico con los beneficios que el control digital ofrece (Nakamura et al., 2004).

Los primeros algoritmos de control usados en arquitecturas de 8-bits tenían algunas restricciones para ser implementados en recursos limitados de memoria. De acuerdo a (Ahson et al., 1983) sugiere algunas estrategias para implementar algoritmos de control clásico en 1K de memoria. Similarmente, el clásico control Proporcional Integral Derivativo (PID) ha sido desde entonces el pilar de diversas aplicaciones exitosas (Ellis, 2004). Sin embargo, algoritmos más complejos –en términos de memoria y tiempo de procesamiento– aparecieron en aplicaciones comerciales de 8-bits a principios de los 80s, por ejemplo el algoritmo de control no-lineal error cuadrático integral con zona muerta variable (IESVDZ) descrito en una aplicación en (Lipták, 2006) mostró que la tecnología estaba lista para incrementar la complejidad de los algoritmos para aplicaciones donde emplear control clásico no era un opción practica. De la misma forma y con la evolución de los microprocesadores a los microcontroladores que integraban diversos periféricos en un solo chip (Schlett, 1998) permitieron reducir las dimensiones del sistema de control y aumentaron rentabilidad de las aplicaciones. Otros algoritmos no-lineales usando microcontroladores empezaron a ser reportados e implementados en aplicaciones industriales. Por ejemplo: sólo en Estados Unidos en esa

época (en los 80s) 80 millones de motores eran usados en aplicaciones nuevas donde se requería control de alta precisión (Bose, 1988) en impresoras, discos duros y dispositivos similares. La investigación de más eficientes algoritmos de control implementados en dispositivos de escasos recursos de hardware estaba en la cúspide (Wang, 1981). Entre los principales alcances logrados en esta área de control de movimiento empleando servomotores fue el control vectorial (Bose, 1988). El uso de observadores y controladores basados en algoritmos cuadrático-lineales (LQ) los cuales fueron usados en productos de alto valor agregado (Korn et al., 1981). El algoritmo de control de modos deslizantes (SMC) también es ampliamente usado en este periodo por sus características de desempeño y pocos recursos de hardware requeridos (Machado y De Carvalho, 1988). El control inteligente basado en redes neuronales (Dicheng et al., 2004), en lógica difusa (Jackson, 1997), y las combinaciones de estos con los métodos tradicionales (Leach, 1995) se convirtieron en alternativas de control que dieron resultados experimentales satisfactorios empleando microcontroladores embebidos.

A medida que se incrementaba la complejidad de los algoritmos nuevas tecnologías de microcontroladores de 16 y 32 bits fueron requeridas. Algoritmos de control adaptivo se implementan en dichas tecnologías y la necesidad de optimizar con modelos de reducción de memoria surgen para hacer posible la implementación de estos métodos en sistemas de pocos recursos (Bruijn et al., 1988).

La era del microcontrolador basados en tecnología Computadora con Conjunto de Instrucciones Reducidas (RISC) donde se podría destacar recientemente a una aplicación basada en control PI empleando un microcontrolador RISC de 8-bits (C. Hernández-Rosales et al., 2006). En versiones posteriores del RISC se tiene el RISC avanzado (ARM) iniciaron importantes simplificaciones en la forma de programar. La combinación de microcontroladores y arreglos de compuertas programables (FPGA), y diversas configuraciones de procesadores en paralelo hicieron posible implementar algoritmos de control automático de mayor carga computacional (Chen et al., 2007), (Nomura et al., 1996), (Sevcik, 2006).

Sin embargo, a medida que se incremento la complejidad del hardware de estos sistemas embebidos el soporte del software quedo rezagado (Napper, 1998). Las diferentes arquitecturas de hardware manejan los datos del programa de acuerdo a sus capacidades de memoria y CPU. El soporte del software para aplicaciones de control automático fue cada vez más especializado. Este hecho fue unos de los factores que orientaron las aplicaciones de control automático a tecnologías centralizadas en computadoras más comerciales, como la PC junto con tarjetas de control y adquisición de propósito general.

El uso de tarjetas de control y adquisición de datos de alta capacidad de memoria y desempeño del CPU han simplificado la integración de sistemas de control automático. Las computadoras comerciales son interfaces

para el empleo de estas tarjetas. Estas permiten al desarrollador hacer aplicaciones de control automático sin conocimiento del hardware de control. Es decir son “cajas negras” que cuentan con soporte de software de alto nivel. La desventaja fundamental de estas tarjetas es el costo y en la siguiente sección se comparará los requerimientos de hardware de un dispositivo microcontrolador de recursos limitados con el de una tarjeta DSpace modelo DS1104 para la caracterización y control automático de un robot subactuado.

III. DESCRIPCIÓN DE LA PLANTA Y DEL SISTEMA EMBEBIDO DE CONTROL

Algunas publicaciones ((Rong-Jong and Li-Jung, 2006), (Magaña and Holzapfel, 1998),) han mostrado lo efectivo que es un robot subactuado (péndulo invertido) en la investigación y en la implementación de dinámica no-lineal. El modelo de la planta considerando la dinámica del robot linealizado alrededor del punto de equilibrio superior y del actuador se puede ver en la Ecuación 1.

$$\begin{aligned} (M + m)\ddot{x} + ml\ddot{\theta} + c_2\dot{x} &= c_1u_o \\ ml\ddot{x} + (J + ml^2)\ddot{\theta} - mgl\theta &= 0 \end{aligned} \quad \text{Ecuación 1}$$

Las matrices A, B y H de la Ecuación 2 se muestran en representación de espacio de estado de la Ecuación 1 y los valores reales de dichos parámetros se pueden ver en la Tabla 1.

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -m^2l^2g/H & -(j+ml^2)c_2/H & 0 \\ 0 & (M_c + m)mg/H & mlc_2/H & 0 \end{bmatrix} \quad \text{Ecuación 2}$$

$$B = [0 \ 0 \ (j + ml^2)c_1/H \ -mlc_1/H]$$

$$\text{where: } H = (M_c + m)j + M_c ml^2$$

Tabla 1 PARAMETROS DEL PENDULO

Constant e	Símbolo	Valor	Unidades
Masa del carro	M_c	0.78	Kg
Masa del péndulo	M	0.06	Kg
Longitud del péndulo	L	0.14	Metros
Inercia del péndulo	J	0.46e-3	Kg metros ²
Aceleración de gravedad	G	9.81	m/seg ²
Coficiente 1(motor)	c_1	1.56	Nw/Volts
Coficiente 2 (motor)	c_2	3.9	Kg metros/seg

El algoritmo de control empleado es un Regulador Cuadrático Lineal (LQR) en donde las ganancias de control se calculan empleando MATLAB con el siguiente código:

```
Ts=0.01; %Tiempo de muestreo
[Ad, Bd] = c2d(A,B,Ts); % Discretización
e = eig(A); %Calculo de eigenvectores
Uc=ctrb(A,B);%Cal Mat controlabilidad
rank(Uc); %Rango de la mat controlab
%Pesos de la matriz Q y r
Q=diag[120,350,10,12]; r=0.001;
```

```
%funcion para calcular las ganancias
[k, S, e]=dlqr(Ad, Bd, Q, r);
```

Dado A y B de la Ecuación 2 se calcula el equivalente del modelo en tiempo discreto T_s (10 ms). El vector k contiene las ganancias de serán retroalimentadas como lo indica la Figura 1, las cuales son: $k=[k_1, k_2, k_3, k_4]$; $k_1 = -30.066, k_2 = -102.107, k_3 = -28.350$ y $k_4 = -15.619$

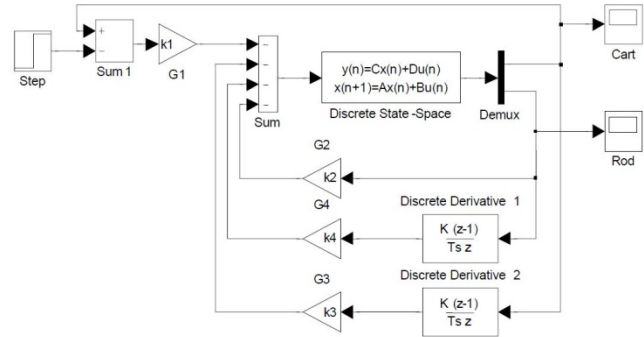


Figura 1 Diagrama simplificado de la planta y el control LQR cuando el sistema tiene el péndulo en la posición vertical superior.

Como se puede ver de la Figura 1 se deriva las posiciones de ángulo del péndulo y posición del carro. Para este proceso se realiza una derivada aproximada y un filtro digital dado por la Ecuación 3.

$$vel = avel_k + (1 - \alpha)vel_{k-1} \quad \text{Ecuación 3}$$

Donde α es el coeficiente del filtro y tiene un valor de 0.9

A. Arquitectura del hardware de control

El microcontrolador empleado es el PIC18F4550 de Microchip. El código fue escrito en lenguaje C empleando el compilador CCS. Aunque el núcleo del 18F4550 tiene un procesador RISC de 8-bits el set de instrucciones es de 16-bits. Con 32K de memoria FLASH para ser usada en el programa de control y 1048 bytes de SRAM para memoria de datos. Contiene 31 terminales de entrada/salida, 13 de estas 31 terminales pueden ser usadas como entradas analógicas multiplexadas. Diversos periféricos de comunicación (uarts, I2C, SPI, USB), módulos de PWM y 5 timers (2 de 8-bits y 3 de 8-16 bits). La operación del ciclo de operación esta por encima de los 40 MHz empleando un cristal de 10Mhz.

B. Arquitectura del software de control

Una arquitectura TT (Pont, 2001) fue empleada para sincronizar el algoritmo de control en tiempo real. El despachador de funciones TT se encarga de 3 tareas fundamentales: adquisición de datos, ejecución del algoritmo de control y actuación de la señal de control. Las cuales son ejecutadas periódicamente como se puede ver en la Figura 2. Un algoritmo de control acciona al actuador del robot (motor que permite el movimiento translacional) de tal forma que

produce una oscilación creciente hasta hacerlo llegar a la posición vertical invertida. Una vez vertical, el algoritmo de control LQR permite controlar el péndulo. El actuador le proporciona la energía al péndulo de forma indirecta para lograr que el péndulo se estabilice en cualquier posición dada por el sp (posición deseada) del carro.

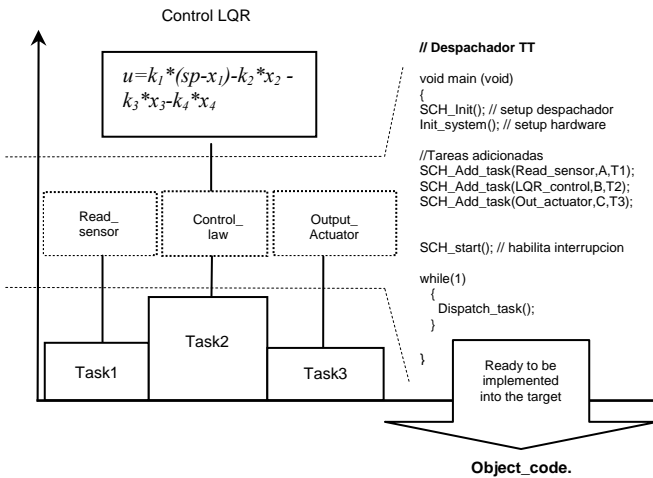


Figura 2 Idea básica de la implementación del algoritmo LQR en un microprocesador de recursos limitados empleando un despachador TT. A, B y C son los “offsets” y T1 T2 y T3 son los números de “Ticks”. Para mayores detalles vea (Pont, 2001)

El segmento de adquisición de posiciones (carro y péndulo) y de estimación de velocidades se muestra a continuación:

```

posicion_carro_pulsos=Encoder(0); //Lee encoder 1
x1=posicion_carro_pulsos*0.000020559; //Pos carro
x3=(x1-xold)/Ts; //Velocidad carro sin filtrar
x3f=alfa*x3f+(1-alfa)*x3; //Vel carro filtrada
xold=x1; // Posicion carro anterior
posicion_pend_pulsos=Encoder(1); // Lee encoder 2
x2=pos_ini+posicion_pend_pulsos*(6.283192/8000);
x4=(x2-teta_old)/Ts; //Vel angulo sin filtrar
x4f=alfa*x4f+(1-alfa)*x4; // Vel angulo filtrada
teta_old=x2; //Poscion angulo anterior

```

El algoritmo de excitación se propuso de la siguiente manera:

```

time=time+Ts;
if(time<0.3)Delta=-6; //Impulso de arranque
else{adicional=X2-3.14159; // teta-π
V_adicional=fabs(V_adicional);
if(V_adicional<=V_adicional_max)
{seno_volt=sin(V_adicional_max);
V_adicional=fabs(seno_volt*seno_volt);
}else
{V_adicional_max=V_adicional;
seno_volt=sin(V_adicional);
V_adicional=seno_volt*seno_volt;
}
V_adicional=V_init*V_adicional;
Volts_in=V_init+V_adicional;
Delta=0;
if(vaf>0)u1=-Volts_in;
else u1=Volts_in;
}
/*Si esta en el cuadrante 1y2 (parte superior) */
if(X1<=1.57 || X1>=4.7124)
u1=u1;
else
u1=-u1;
}
/* Control PI para mantener al centro el carro */

```

```

error2=SP-X1; // setpoint-pos carro
Iz=Izold+error2*Ts; //parte integral
u2=kp*error2+ki*Iz; //control PI
Izold=Iz; // Memoria
/* Señal u (voltaje de salida) */
u=Delta+u1+u2;

```

El algoritmo de control LQR se escribe de la siguiente forma, incluyendo la salida al actuador:

```

u=k1*(SP-X1)-k2*X2-k3*X3f-k4*X4f;
if(u>0)bit_set(portc,0); //mov en CW
else bit_clear(portc,0); // mov en CCW
/* Convierte salida de voltaje a num PWM */
uf=u*volts_2_PWM;
/* verificar los limites de saturación */
if(uf<0)uf=-uf;
if(uf>Vsat)uf=Vsat;
/* Escribir en el registro el valor de salida*/
u_PWM=uf; /* de float a char (8-bits)
set_pwm2_duty(u_PWM); // modulo CCP2

```

IV. RESULTADOS EXPERIMENTALES

El sistema de control y la planta se muestra en la Figura 3, este sistema tiene dos sensores para medir la posición del carro (que por la disposición mecánica es de 12160 pulsos/mm) y del péndulo (2000 pulsos/rev). Cabe mencionar que el conteo emplea una técnica de multiplicación por 4, con el fin de incrementar la resolución de los sensores. Se emplea un contador de 32-bits de dos canales (HCTL-2032) el cual esta diseñado para acondicionar la señal de los encoders y cuenta con un multiplexor para ser transferida la cuenta de manera paralela en 8 bits al microcontrolador. El actuador es un motor de corriente directa de 12 Volts. La señal de salida del controlador es de modulación de ancho de pulso (PWM) de 45 Khz modulada por la variable de control u . El carro tiene un rango de movimiento de 1 metro y el péndulo puede girar libremente alrededor del eje del sensor.

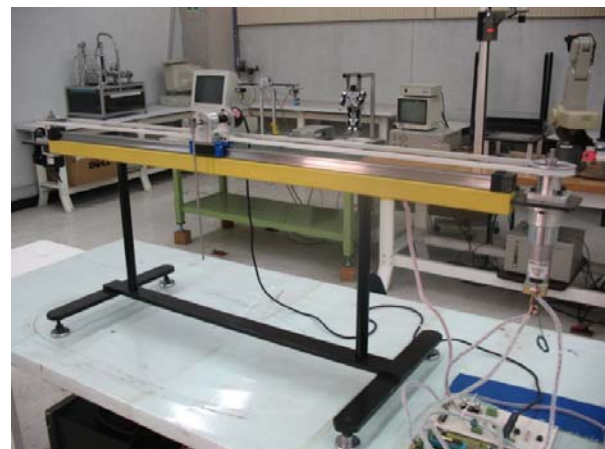


Figura 3 Robot sub-actuado usando el microcontrolador PIC18F4550

En la habilitación del algoritmo fueron medidos los tiempos de ejecución de cada tarea (adquisición, control y actuación).

El código del programa para balancear el péndulo y llevarlo a la vertical mostrado en la llevo el tiempo de ejecución mas largo ya que implica cálculos basados en funciones trigonométricas.

Tabla 2 Datos medidos sobre el tiempo de ejecución de la tarea de adquisición y estimación de velocidades.

Media	Desviación Estandard	Error Estandard	Min	Max
45.4µs	7.1 µs	27.8 x10 ⁻⁶	36.3 µs	55.1 µs

Tabla 3 Datos medidos sobre el tiempo de ejecución de la tarea de balanceo (swing-up) incluyendo la actuación del motor.

Media	Desviación Estandard	Error Estandard	Min	Max
410.2µs	46.3 µs	52.7 x10 ⁻⁵	264.6 µs	610.1 µs

La tarea de control LQR y actuación se muestran en la Tabla 4.

Tabla 4 Datos medidos sobre el tiempo de ejecución de la tarea de control LQR y actuación del motor

Media	Desviación Estandard	Error Estandard	Min	Max
245.6µs	13.1 µs	87.3 x10 ⁻⁷	198.2 µs	269.3 µs

El desempeño de la planta empleando el microcontrolador de 8-bits comparado con la tarjeta DS1104 no tiene una diferencia medible y repetible ya que los dos sistemas están sincronizados con las mismas ganancias de control.

La tarjeta DS1104 permite la programación visual de las rutinas y se transmite el código generado a dicha tarjeta sin usar el lenguaje ensamblador de dicho sistema. Sin embargo el costo de la tarjeta puede no ser justificable para la mayoría de los presupuestos de universidades y centros de educación de posgrado en México.

V. CONCLUSIONES

Este artículo proporcionó una amplia reseña histórica de la evolución de los sistemas embebidos de 8-bits desde la tecnología de microprocesadores hasta una de las tecnologías más usadas en microcontroladores de la actualidad con un enfoque en control automático. Se mostró que dicha tecnología tiene potencial para implementar a un muy bajo costo aplicaciones de control automático de relativa complejidad. La exploración del uso de estos dispositivos en sistemas de control para aplicaciones portátiles puede ser el detonante para implementar sistemas de control complejos sin depender de hardware y software de propósito general (normalmente cajas negras de muy alto costo) que normalmente no se producen en México. Los algoritmos e implementaciones mostradas fueron hechos en un robot sub-actuado (péndulo invertido) diseñado y construido para explorar el uso de control robusto en futuras aplicaciones. Por ejemplo, el sistema está construido para modificar deliberadamente la estructura de la planta y validar experimentalmente el nivel de robustez ante incertidumbres estructurales.

El sistema de control basado en el micro de 8-bits tuvo la memoria suficiente para adquirir datos para la caracterización del sistema. No sólo se programó la rutina de control del sistema con el péndulo en posición invertida sino la rutina de auto-excitación (swing-up) para llevarlo a

dicha posición. Los algoritmos diseñados fueron programados con un despachador TT con lo que se dota de robustez en la sincronía de la ejecución del control en tiempo real además de mejorar la predictibilidad de las funciones de adquisición, control y actuación del algoritmo.

La tecnología actual y disponible en México en materia de microcontroladores de 8-bits (como el caso de estudio descrito en este artículo) se puede concluir que cuentan con la capacidad de ejecutar tareas de control más complejas de lo que la percepción general del desarrollador tiene en el caso concreto de aplicaciones de control automático de plantas de relativa complejidad. A pesar de tener la desventaja de no tener soporte gráfico de alto nivel (como la DS1104), los microcontroladores de 8-bits son una alternativa confiable para muchos desarrollos en donde el costo y el volumen exijan rentabilidad.

VI. AGRADECIMIENTOS

Este trabajo fue desarrollado con el apoyo de la COSDAC para el proyecto “diseño y construcción de un robot sub-actuado” desarrollado de Julio a Diciembre del 2009, en el Centro Nacional de Actualización Docente perteneciente a la DGETI. Los becarios participantes fueron los ingenieros: José Juan Varela Herrera, Josefina Hernández Wong, Roberto Moreno Martínez, Julio Cesar Leal Gonzalez y Julio César Aguilar Sánchez.

REFERENCIAS

- Lewis, F.L., 1992. Applied Optimal Control and Estimation. Prentice-Hall.
- Gaetano, B. and Roy, W., 2000. Embedded Computation Meets The World Wide Web. Commun. ACM, 43(5): 59-66.
- WSTS, 2005. Embedded Systems Statistics. In: <http://www.wsts.org/> (Editor). World Semiconductor Trade Statistics
- Moore, G., 1979. VLSI: Some Fundamental Challenges. IEEE Spectrum 16(30).
- Napper, S., 1998. Embedded-System Design Plays Catch-Up. IEEE Computer, 31(8): 120 - 118-19.
- Mrad, F., El-Hassan, N., Mahmoud, S.E., Alawieh, B. and Adlouni, F., 2000. Real-Time Control of Free-Standing Cart-Mounted inverted Pendulum Using Labview RT. In: IEEE (Editor), industry Applications.
- Butazzo, G., 2005. Rate Monotonic Vs. EDF: Judgment Day. Real-Time Systems, 29(1): 5-26
- Pratt, W.C. and Brown, F.M., 1975. Automated Design of Microprocessor-Based Controllers. IEEE Transactions on industrial Electronics and Control instrumentation, IECI-22(3): 273-279
- Faggin, F., Hoff, M.E., Mazor, S. and Shima, M., 1996. The History of The 4004. Micro, IEEE, 16(6): 10-20.
- Goksel, K., Knowles, K.A., Parrish, E.A. and Moore, J.W., 1975. An intelligent industrial Arm Using A Microprocessor. IEEE Transactions on industrial



- Electronics and Control instrumentation, IECI-22(3): 309-314.
- Erickson, K.T., 1996. Programmable Logic Controllers. *IEEE Potentials*, 15(1): 14-17.
- Donaghey, L.F., 1976. Microcomputer Systems For Chemical Process Control. *Proceedings of The IEEE*, 64(6): 975-987.
- Mersten, G.S., 1979. Microprocessors in Aerospace Applications. In: *IEEE (Editor), Comcon* pp. 264-269.
- Graupe, D., Magnussen, J. and Beex, A., 1978. A Microprocessor System For Multifunctional Control of Upper-Limb Prostheses Via Myoelectric Signal Identification. *IEEE Transactions on Automatic Control*, 23(4): 538 - 544.
- Lamb, J.W. and Whelan, H.A., 1976. Microcomputer-Based Electronic Error Corrector Bar For A Numerically-Controlled Machine Tool, United States, pp. Pages: 10.
- Nakamura, M., Goto, S. and Kyura, N., 2004. *Mechatronic Servo System Control*. Springer.
- Ahson, S.I., Lamba, S.S., Chaube, M.C. and Chandra, N., 1983. A Microprocessor-Based Multiloop Process Controller. *IEEE Transactions on Industrial Electronics*, IE-30(1): 34-39.
- Ellis, G., 2004. *Control Systems Design Guide, Using Your Computer To Understand and Diagnose Feedback Controllers*. Elsevier Academic Press.
- Lipták, B.G., 2006. *Instrument Engineerings' Handbook: Process Control and Optimization*. ISA.
- Schlett, M., 1998. Trends in Embedded-Microprocessor Design. *Computing*, 31(8): 44-49.
- Bose, B.K., 1988. Technology Trends in Microcomputer Control of Electrical Machines. *IEEE Transactions on industrial Electronics*, 35(1): 160-177.
- Wang, H.-C., 1981. Sampling Period and Stability Analysis For The Microcomputer-Based Motor Control Systems. *IEEE Transactions on industrial Electronics and Control instrumentation IECI-28(2)*: 98-102.
- Korn, J.E., Carroll, E., Johnson, R. and Kullack, B., 1981. Microcomputer Implementation of Control Algorithms For Weapon Pointing and Stabilization, *Proceedings of The 1981 Army Numerical Analysis and Computers*.
- Machado, J.A.T. and Galhano, A.M.S.F., 1995. Benchmarking Computer Systems For Robot Control. *IEEE Transactions on Education*, 38(3): 205-210.
- Dicheng, L., Qing, W., Jiangfeng, Z. and Zhi, L., 2004. The Study of Microcomputer Adaptive Control System of Controlled Rectifier Device. In: *IEEE (Editor), The 4th international Power Electronics and Motion Control Conference*, pp. 704-707.
- Jackson, A., 1997. A New Microcontroller With Fuzzy inference instructions Simplifies Controller Designs. In: *IEEE (Editor), Aerospace Conference*, pp. 491-503.
- Leach, R., 1995. PID Control With 32-Bit RISC Microprocessors. In: *IEEE (Editor), IEEE Technical Applications Conference and Workshops Northcon95*, pp. 178-182.
- Bruijn, P.M., Roodink, P.A.M. and Verbruggen, H.B., 1988. A Simple Implementation of Adaptive Predictive Controllers. In: *IEEE (Editor), international Conference on Control* pp. 517-522.
- C. Hernández-Rosales, R. Femat, G. Quiroz, 2006 "Make a Discrete-Time PI controller on an 8 bit Microcontroller". *Embedded Systems Design*, Vol. 19, No.1, p.p 28-43.
- Chen, X., Pu, H., Wang, X., Sun, Y. and Jia, W., 2007. Control System of A Modular and Reconfigurable Multilegged Robot. In: *IEEE (Editor), international Conference on Mechatronics and Automation*, pp. 1926-1931.
- Nomura, H., Hayashi, I. and Wakami, N., 1996. A Learning Method of Fuzzy inference Rules By Descendent Method. In: *IEEE (Editor), Fuzzy Systems, IEEE international Conference* pp. 203-210.
- Sevcik, R., 2006. Future FPGA Technologies, in Partnership With Universities, *Proceedings of The 19th international Conference on VLSI Design Held Jointly With 5th international Conference on Embedded Systems Design*. IEEE Computer Society.
- Rong-Jong and W. C. Li-Jung, "Adaptive stabilizing and tracking control for a nonlinear inverted-pendulum system via sliding-mode technique," *IEEE Transactions on Industrial Electronics*, vol. 53, pp. 674 - 692, 2006.
- Magana, M.E. and Holzapfel, F., 1998. Fuzzy-Logic Control of An inverted Pendulum With Vision Feedback. *IEEE Transactions on Education*, 41(2): 165 - 170.
- Pont, M.J., 2001. Patterns for time-triggered embedded systems, *Building reliable applications with 8051 family of microcontrollers*.