DISEÑO DEL CONTROL DE POSICIÓN DE UN MOTOR DC CON HARDWARE SIMPLIFICADO

DESIGN OF A DC MOTOR POSITIONING CONTROL BY SIMPLIFIED HARDWARE

Jorge Gustavo Butler Blacker[1], Luis Leoncio Figueroa Santos[2], Trini Castillo Belsuzarri[3], Jorge Luis Inca Rodríguez[4]

RESUMEN

En el presente trabajo se presentan los detalles necesarios para el diseño del control de posición de un motor DC, teniendo en cuenta los parámetros del motor dados por el fabricante. Explicaremos los métodos para hallar los parámetros reales del motor, con los cuales se hallan los parámetros del controlador, además de las constantes de Kalman y de control utilizados en los programas tanto en lenguaje C++ como ensamblador del DSP que queramos utilizar, los datos de la señal de referencia, del encoder de posición, de la señal de control y de la señal de error que se pueden obtener como resultado del programa en tiempo real con lenguaje C++, dichos datos de salida se dibujan con software de Matlab. Con los datos obtenidos en tiempo real, se pueden crear los dos archivos de entrada de señal de referencia y del encoder de posición al programa en lenguaje ensamblador del DSP seleccionado, y para producir la señal de control que también es obtenida como resultado del programa en lenguaje C++, finalmente dibujamos los resultados del programa en lenguaje ensamblador de DSP seleccionado con software de Matlab.

Palabras claves.- Identificación, SISO, Matlab, Señal de referencia, Señal de control, Señal de error, Lenguaje ensamblador, DSP, Encoder.

ABSTRACT

In this work is presented the required details for the design a DC motor positioning control, having into account the parameters of the motor given by the manufacturer. It is explain in detail the method used to find their operation parameters, with the controller parameter, the Kalman and control constants are calculated. All of those used in the C++ language as well as in DSP assembler. With the data collected in real time is produced the control signal that also is obtained from C++, finally we drew the results in DSP assembly language.

Keywords.- Identification, SISO, Matlab, Reference signal, Control signal, Error signal, Assembly language, DSP, Encoder.

INTRODUCCIÓN

En este artículo se da a conocer los resultados de la investigación sobre el diseño del control de posición de un motor con hardware simplificado y; con ello poner de manifiesto las partes y los

---

[1]Ing. Master Docente investigador de la Facultad de Ingeniería Eléctrica y Electrónica de la Universidad Nacional de Ingeniería, [2]Ing. Docente investigador de la Facultad de Ingeniería Eléctrica y Electrónica de la Universidad Nacional de Ingeniería, [3]Ing. Docente investigador de la Facultad de Ingeniería Eléctrica y Electrónica de la Universidad Nacional de Ingeniería, [4]Ing. Docente investigador de la Facultad de Ingeniería Eléctrica y Electrónica de la Universidad Nacional de Ingeniería.

procedimientos utilizados. Primero se realiza la simulación usando el Software de Matlab para establecer una ruta de trabajo y así elaborar programas en lenguaje ensamblador y C++, luego se procede a las pruebas experimentales.

La inestabilidad producida por el sobre-impulso inicial, obliga a ver la forma de cómo llegar a la posición deseada en un tiempo prudencial y corto, tal que esta posición deseada nunca sea superada y se busque la forma adecuada de lograr el éxito del control de posición, además de la posibilidad de trabajar digitalmente si se cuenta con un DSP evitando el ruido causado por señales que pudieran interferir.

En este trabajo se tendrá un análisis simplificado de la etapa del generador PWM, que deberá ser realizado por software dentro del DSP.

Las dificultades, bondades y limitaciones como las que son presentadas [1], son tomadas como conocidas y el DSP elegido es el TMS320LF240X o el dsPIC33FJ128MC706.

El trabajo realizado se puede mostrar en la Fig. 1, donde el controlador digital que utiliza para su programación en lenguaje C++ o ensamblador del DSP para obtener la señal de control, recibe como entradas la señal de referencia y los datos producidos por el encoder.

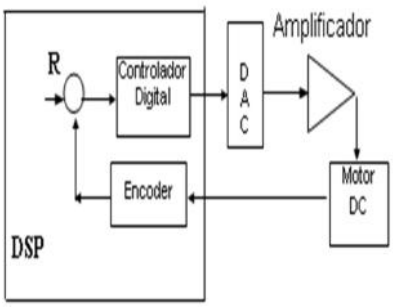

Fig. 1 *El DSP utilizado para controlar un motor de corriente continua.*

El diseño se fundamenta en el criterio de optimabilidad de los controladores que consiste en producir una señal de control adecuada para el proceso del control de posición, pues, es esta señal la que va al motor. El modelo de programa seguido ya con los recursos de mínimos cuadrados como las ecuaciones utilizadas son las del modelo en Simulink [2].

IDENTIFICACIÓN

Implementando la planta del motor DC con los parámetros JM y bM dados por el fabricante (Pittman) se establece la salida para una entrada de escalón unitario [2], se tiene la siguiente ecuación:

$$\frac{\omega(0)}{u(s)} = \frac{nKtKac}{beffR + nEKt} \quad (1)$$

De donde obtenemos beff pues Kt, Kacson las ganancias necesarias del amplificador para activar al motor DC, n es la reducción del motor, E, JM y bM son datos del fabricante para el primario del motor y debemos utilizar Jeff y beff en los programas.Luego, con la ubicación del polo a 3db.

$$\omega_{3db} = \frac{beff}{Jeff} + \frac{nEKt}{RJeff} \quad (2)$$

En la respuesta en frecuencia, leemos la ubicación del polo con el modelo en software de Matlab de la Fig. 2.

Empleando la ecuación anterior, hallamos que Jeff son los datos que podemos emplearlo en el programa de control.

Es así que hemos identificado los parámetros de la planta.

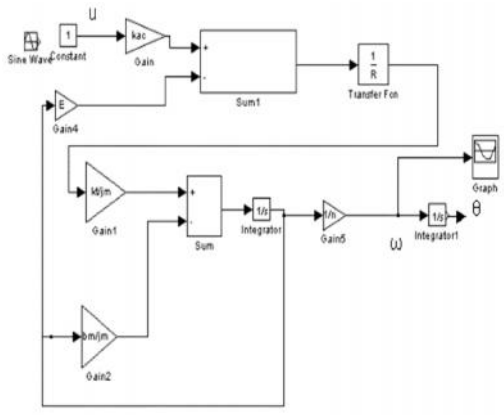

Fig. 2 *Modelo en simulink del motor DC con secundario usado para la identificación de los parámetros reales.*

PARÁMETROS DEL SISTEMA

Para los parámetros de la matriz que representa la planta en el controlador se utiliza un programa en Matlab considerando los datos hasta aquí obtenidos. Las instrucciones de dicho programa se muestran a continuación:

```
n=red= 19.741; R= 7.38;  Kt= 31.071e-3; E=Ke= 31.0352e-3;  Kac =A= 14.9; Jeff= 5.63e-5; beff= 7.05e-5; M= red*Jeff;  B= red*beff;  N= 0;
%
F=100, T=1/F
 %estos valores de M, B, N: H(s)= K/s(s+a)
%
K= A*red*Kt/(M*R);  a= (B*R+red*red*Kt*Ke)/(M*R);
numc= [ K ];  denc= [ 1 a 0 ]; [numd, dend]= c2dm(numc, denc, T, 'zoh');
a1= dend(2);  a2= dend(3);  b1= numd(2);  b2= numd(3);
%
```

Obteniendo por resultado:

a1 = -1.6246, a2 = 0.6246, b1 = 0.0479, b2 = 0.0410

CONSTANTES DE LAS ETAPAS DEL PROGRAMA

Procederemos a calcular las constantes que identifican las etapas del programa con el empleo de software de Matlab.

DISEÑO ESTIMADOR CUADRÁTICO LINEAL Y DISCRETO

Se utiliza el comando DLQE de Matlab para obtener la solución de la ecuación de Ricatti de orden 2, la ganancia llamada de Kalman por ser obtenida con el filtro estacionario de Kalman discreto siendo de orden 2*1, permite actualizar las variables del observador. Para obtener la ganancia vemos que:

$$[M,P,Z,E] = DLQE(G,H,C,Q,R) \qquad (3)$$

Donde: G=[0 1; -0.6246 1.6246], H=[0;1], Q=0.01, R=0.04.

Obteniendo: $M = Kb = \begin{bmatrix} 1.10 \\ 1.16 \end{bmatrix}$

Hallándose además la solución de la ecuación de Riccati que es una matriz de orden 2, a la que llamamos PB cuyos elementos son agregados al programa como constantes, dicha matriz es:

$$P = PB = \begin{bmatrix} 0.5386 & 0.5605 \\ 0.5605 & 0.5986 \end{bmatrix}$$

DISEÑO DEL SISTEMA REGULADOR LINEAL CUADRÁTICO Y DISCRETO

Se utiliza el comando DLQR de Matlab para obtener la solución de la ecuación de Ricatti de orden 3, la ganancia llamada de control que se obtiene con realimentación que reduce al mínimo la función del costo sujeta a la ecuación de diferencias.

Para obtener la ganancia de control tenemos la ecuación (4) de la siguiente columna:

$$[K,S,E] = DLQR(GE,HE,Q,R) \qquad (4)$$

Donde GE, HE, Q y R son como se muestra a continuación:

$$GE = \begin{bmatrix} 0 & 1 & 0 \\ -0.6246 & 1.6246 & 0 \\ -0.041 & -0.0479 & 1 \end{bmatrix},\ HE = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

$$Q = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.1 \end{bmatrix};\ R = 0.05$$

Obteniendo la matriz de la constante de ganancia de la etapa de control K de orden 1*3 que es:

$$K = \begin{bmatrix} -0.5215 & 1.3460 & -0.5128 \end{bmatrix}$$

PROGRAMA DE PRUEBA EN MATLAB

El programa cuenta con todas las etapas necesarias, la salida sigue a la entrada en cada pasada del programa. Se utilizan todos los datos hasta aquí vistos. La sintaxis del programa llamado sesenta por que la posición deseada es casi 1 cercano a 60º en radianes, se tiene a continuación:

```
%
y=0;x=0;z=0;v=0.2;T=0.05;Nsimul=40;
for k=0:Nsimul
t=k*T
if(t<=1.0)r=0.4112+0.5788*t;
elseif(1.0<t<20.0)          r=0.99;end
if(t<0.25)                  W=2.432*t;
elseif(0.2<t<=2.)           W=r;end

e=W-r;W

rr=e-0.041*x-0.0479*y;r
x=x+1.1*rr;
y=y+1.16*rr;
z=z-e;e

u=-(-0.5211*x + 1.3371*y - 0.5137*z);
Raux1=y;
y=-0.6246*x+1.6246*y+u;
x=Raux1;
u=abs(u);u
if(u+v>1.4)                 u=1.4;
else
end
end
```

Luego de la corrida se obtienen los datos correspondientes (Anexo 1). Con los datos del Anexo 1 y el siguiente programa se obtiene la gráfica de la señal de control, Fig.3.

```
loadsesenta.out
t=sesenta(:,1);
u=sesenta(:,4);
plot(t,u);
grid;
plot(t,u);
grid;
title ('control(u)');
xlabel('t');
ylabel('u(t)');
```

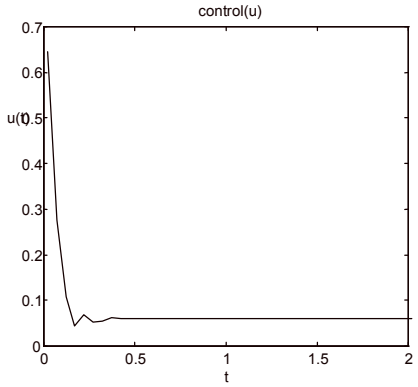

Fig. 3 *Señal de control* ***u*** *para el programa de prueba en Matlab.*

PROGRAMA DE PRUEBA EN LENGUAJE C++

Un programa en lenguaje C++ para ser corrido en tiempo real, debe tener en cuenta toda la estructura de hardware del sistema [2], este programa es importante pues se obtienen los datos de los registros que serán incluidos en el programa de lenguaje ensamblador del DSP como son: la señal de referencia y del encoder de posición.

Es recomendable que el trabajo de obtención de informe sea depurado y fundamentalmente sea sin ruido pues el dispositivo que vamos a utilizar reproduce con fidelidad hasta el mismo ruido. Se recomienda usar un programa adaptivo y de regresión lineal basada en la técnica de mínimos cuadrados que puede reemplazar la etapa de IRLS como en la referencia [1, 2].

Sin embargo, se pudo traducir al lenguaje ensamblador las etapas del programa en lenguaje C++ que se empezaron a agregar porque fueron posibles.

PROGRAMA DE CONTROL DE POSICIÓN EN C++

El programa consta fundamentalmente de las siguientes etapas:

1. Iniciación
   a. Inicialización de la tarjeta de adquisición de datos.
   b. Inicialización de las variables.
   c. Establecimiento de la posición cero de la varilla.

2. Algoritmo de control
   a. Detección del flanco de subida de clock
   b. Medición de las salidas
   c. Enunciado de la referencia
   d. Estimación de estados
   e. Cálculo y aplicación de la ley de control.

El programa se encuentra en el Anexo 2, luego de la corrida del programa se obtienen los datos mostrados en el Anexo 3. El siguiente programa de Matlab permite graficarel resultado del programa con los datos del Anexo 3.

```
load \comp.out
t=comp(:,1);
r=comp(:,2);
y=comp(:,3);
u=comp(:,4);
e=comp(:,5);
subplot(411);
plot(t,r,t,y);
grid;
title ('referencia (r)')
xlabel('t')
ylabel('r(t)')
subplot(412);
plot(t,u);
grid;
title ('control (u)')
xlabel('t')
ylabel('u(t)')
subplot(413);
plot(t,y);
grid;
title ('posición (y)')
xlabel('t')
```

El gráfico que muestra la respuesta es el de la Fig. 4.

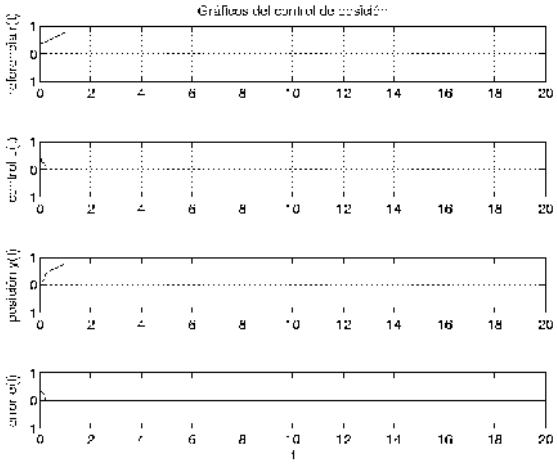

Fig. 4 *Gráficos del control de posición del programa implementado en C++.*

PROGRAMA DE PRUEBA EN LENGUAJE ENSAMBLADOR

Normalmente se tiene datos de un sistema en tiempo real obtenido en una computadora Pentium con el programa en lenguaje C++ (versión utilizada 3.5), de lo contrario la opción es usar los datos de la prueba en software de Matlab como son la referencia y la posición dada por el encoder que serán incluidas en el programa en ensamblador. Pudiendo trabajar con la señal de error, se crea un programa de cabecera y comp1.asm como la diferencia entre la señal y la de encoder de posición para simplificación, pues, el programa utiliza la señal de error; con facilidades de utilizar los dos programas el C++ y el ensamblador, en este último se puede guardar la señal del encoder de posición obtenido de una buena corrida en lenguaje C ++, como una adecuada referencia.

Debido a nuestras limitaciones se creó el programa mencionado de cabecera ycomp1.asm que se incluye en el programa.

PROGRAMA DEL CONTROL DE POSICIÓN EN ENSAMBLADOR DEL DSP

El programa consta fundamentalmente de las siguientes etapas:

1. Iniciación
   a. Inclusión de los programas de datos.

b. Inicialización de las variables.
c. Establecimiento de la posición cero de la varilla.

2. Algoritmo de control
   a. Subrutina para discretizar en el tiempo de muestreo.
   b. Medición de las salidas.
   c. Enunciado de la referencia.
   d. Estimación de estados.
   e. Cálculo y aplicación de la ley de control.

Se escribe un programa llamado sec.asm [1], el cual se encuentra en el Anexo 4. Los resultados de la corrida del programa anterior se muestran en el Anexo 5.
El siguiente programa de Matlab permite graficar los resultados del resultado del programa.

```
load \sec.out
t=sec(:,1);
r=sec(:,2);
y=sec(:,3);
u=sec(:,4);
e=sec(:,5);
subplot(411);
plot(t,r);
grid;
title ('referencia (r)')
xlabel('t')
ylabel('r(t)')
subplot(412);
plot(t,u);
grid;
title ('control (u)')
xlabel('t')
ylabel('u(t)')
subplot(413);
plot(t,y);
grid;
title ('posición (y)')
xlabel('t')
```

La grafica obtenida se muestra en la Fig. 5.

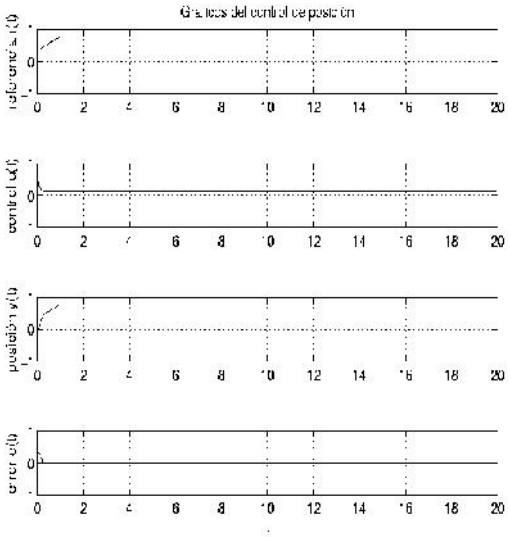

Fig. 5 *Gráficos del control de posición del programa implementado en C++.*

CONCLUSIONES

De las pruebas observamos que el tiempo de proceso puede ser reducido a 2 s, tiempo suficiente para que se llegue al posicionamiento. De este trabajo obtenemos las siguientes conclusiones:

La comunicación ya sea con el amplificador PWM (ahora dentro del DSP) o con la salida del sensor de posición es factible utilizando puertos paralelos, pero presentan ruido posible de eliminar utilizando DSP.

Es posible obtener la identificación de los parámetros relevantes del motor con eje secundario, con los datos que proporciona el fabricante utilizando el modelo del motor en Simulink, del software Matlab y las ecuaciones de la respuesta en frecuencia.

La identificación del sistema con redes neuronales, mediante el entrenamiento proporciona la señal de salida del sistema, lo que importa es la fidelidad de la señal de salida, el método de mínimos cuadrados proporciona una señal continua lo que permite confeccionar la base adecuada de datos que no puede ser manipulada mejor que con un DSP.

El procedimiento de diseño se simplifica obteniendo las ganancias del programa adaptivo con el software de Matlab.

Las ecuaciones del modelo matemático permiten tener la simulación del sistema tanto en continua como en discreto.
Es posible tener un programa en lenguaje ensamblador del DSP que cumpla las mismas funciones que el programa en C++ a pesar que se tiene que respetar las características de programación de punto fijo en el DSP.

Con el empleo del pll del DSP se incrementa la frecuencia de muestreo en diez lo que permite tener todas las instrucciones en un ciclo de programa.

La subrutina de espera a interrupción permite la adecuada discretización de la señal de control que es renovada cada ciclo debido a la función de la subrutina de espera a interrupción.

El DSP produce señales de control semejantes a las producidas por la computadora con programas en C++ pero las del DSP son menos ruidosas.

Es posible, emular los resultados del sensor mediante instrucciones del programa de control en el lenguaje ensamblador del DSP con archivos de datos cargados como cabeceras y actualizados cada ciclo de programa, dicha secuencia de posición es posible obtenerla en corridas con programas en C++.

La subrutina de espera a interrupción permite la adecuada discretización de la señal de control que es renovada cada ciclo debido a la función de la subrutina de espera a interrupción.

REFERENCIAS

1. **Inca Rodríguez, J. L., Gutiérrez, A., Figueroa, L.,** "Diseño de un Sistema de Control para un Motor DC Usando DSP56002evm". Revista TECNIA. Julio-Diciembre 2003 Vol. 13 Nº2.
2. **Inca Rodríguez, J. L., "**Diseño de un Sistema de Control de Posición de un motor DC usando Procesador Digital de Señales". Tesis de Título Profesional UNI. 2002.

Correspondencia: jbutler@iciuni.edu.pe