

Simulación y control digital de un motor DC con Scilab-XCOS

G. Olivares⁽¹⁾, F.Gómez⁽²⁾, M.Damas⁽³⁾

Departamento de Arquitectura y Tecnología de Computadores.
ETSI Informática y de Telecomunicación. Universidad de Granada
Granada, España

⁽¹⁾gonzalo@ugr.es; ⁽²⁾fgomez@ugr.es; ⁽³⁾mdamas@ugr.es

Resumen. Se describe en este artículo un conjunto de técnicas de control discreto de un motor DC empleando Scilab y XCOS como alternativa de software libre. El conjunto de experiencias propuestas se están empleando en las prácticas de las asignaturas impartidas por el Área de Ingeniería de Sistemas y Automática y el Departamento de Arquitectura y Tecnología de Computadores de la Universidad de Granada con resultados satisfactorios.

Palabras Claves: Control digital, Scilab-XCOS, Motor de corriente continua. PID, LQR, LQI.

Abstract. This paper describes a set of discrete control techniques of a DC motor using Scilab and XCOS as a free software alternative. The set of experiences proposed is being used in the practices of the subjects taught by the Systems Engineering and Automation Area and the Department of Architecture and Computer Technology of the University of Granada with satisfactory results.

Keywords: Digital-Control, Scilab-XCOS, DC motor.

1 Introducción

Matlab-Simulink es sin duda la herramienta de software de simulación más utilizada en la enseñanza de sistemas de control. No obstante, debido a su coste, no todas las universidades disponen de esta herramienta, o bien, puede haber dificultades puntuales de acceso a los computadores de los laboratorios que disponen de las licencias de uso, tal y como ha ocurrido durante periodos de confinamiento por pandemia, en los cuales los estudiantes no han podido usarlo.

Tras analizar varias posibilidades de software libre, hemos empezado a usar Scilab-XCOS como alternativa viable para el diseño y simulación de sistemas de control, y también como herramienta de control en tiempo real de sistemas mecatrónicos básicos, en los laboratorios de prácticas de algunas asignaturas de grado y master impartidas por el Departamento de Arquitectura y Tecnología de Computadores de la Universidad de

Granada (Informática Industrial, Sistemas de Control, Control Digital de Sistemas Mecatrónicos, etc.).

El motor de corriente continua es uno de los principales componentes que se utilizan en sistemas mecatrónicos. Conocer bien su comportamiento, modelado y métodos de control es básico para avanzar y abordar más adelante el control de sistemas mecánicos más complejos. Además, controlar la posición angular del eje de un motor (servomotor) o la velocidad angular del mismo, permite de una forma sencilla, experimentar y asimilar correctamente un conjunto amplio de conceptos relacionados con el desarrollo de sistemas de control, tales como: funciones de transferencia continuas y discretas, estabilidad, controlabilidad, observabilidad, control PID, control LQR, representación en el espacio de estados, control por ubicación de polos, controladores-observadores, y también experimentar con filtros digitales, sistemas con ruido, etc. En su día ya preparamos un conjunto de guiones y ejercicios prácticos de control de una maqueta de motor DC a realizar con Matlab y Simulink [1].

Ahora presentamos algunos ejemplos de ejercicios prácticos para la docencia de sistemas de control digital realizados con Scilab-XCOS y con un equipamiento hardware distinto.

2 Descripción del equipamiento hardware

Para el desarrollo de las prácticas utilizamos un kit compuesto por un motor DC [2], con una relación de engranajes de 20:1, que dispone a su vez de un codificador incremental, en cuadratura, de 13 ranuras, que suministra 1040 pulsos por vuelta. El controlador es una tarjeta Arduino Mega 2560 [3] e incluye una tarjeta de potencia con el adaptador Toshiba TB6612 [4].

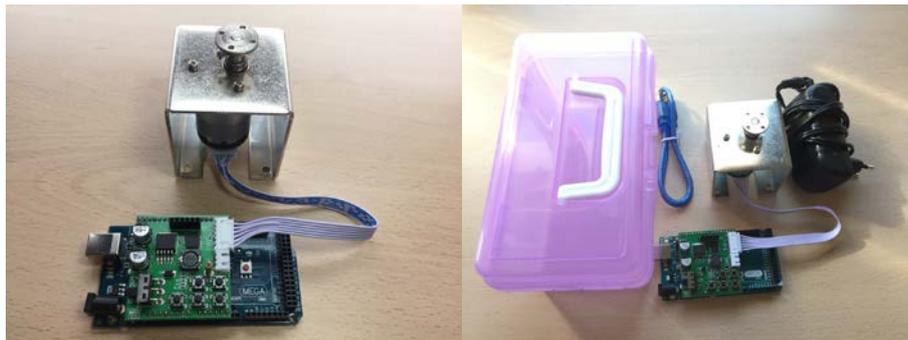


Fig 1. Motor conectado a tarjeta controladora y fotografía del kit completo

3. Scilab

Scilab [5] es un paquete de software libre diseñado para la simulación de sistemas en general, y de sistemas de control en particular, que ha sido desarrollado en Francia por el INRIA (*Institut National de Recherche en Ciencias et Technologies du Numérique*). Al contrario que otras herramientas de software como Octave, la lista de comandos de Scilab no es compatible con Matlab, aunque siempre dispone de funciones equivalentes. Sin embargo, Scilab incluye XCOS [6], que permite programar en forma gráfica en forma similar a Simulink, y dispone de una amplia biblioteca (denominada paleta) con todos los módulos necesarios, tal y como se puede apreciar en la figura 2.

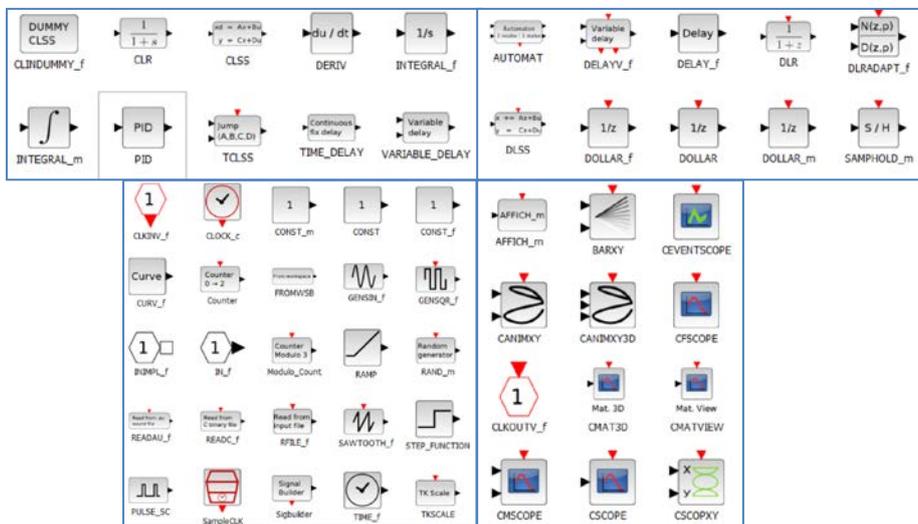


Fig 2. Ejemplo de módulos de XCOS

Scilab se instala muy rápidamente, y requiere muchos menos recursos computacionales que Matlab, lo que es importante para que se pueda usar en la gran mayoría de los computadores de los estudiantes. Dispone además de todas las funciones necesarias para el diseño y la simulación de sistemas de control.

XCOS dispone además de una biblioteca de módulos de interfaz con las tarjetas Arduino Mega y Arduino UNO (ver figura 3), lo que permite realizar aplicaciones de control en tiempo real con dichos dispositivos, siempre que el periodo de muestreo no sea inferior a 0.03 segundos. Esto último hace que esta característica no se pueda usar para controlar algunos sistemas mecatrónicos ampliamente usados en experiencias prácticas docentes (péndulo invertido, sistema de bola en plano, mini-segway, etc.), pero es suficiente para ser utilizada para el control del motor DC, objetivo de nuestro trabajo.

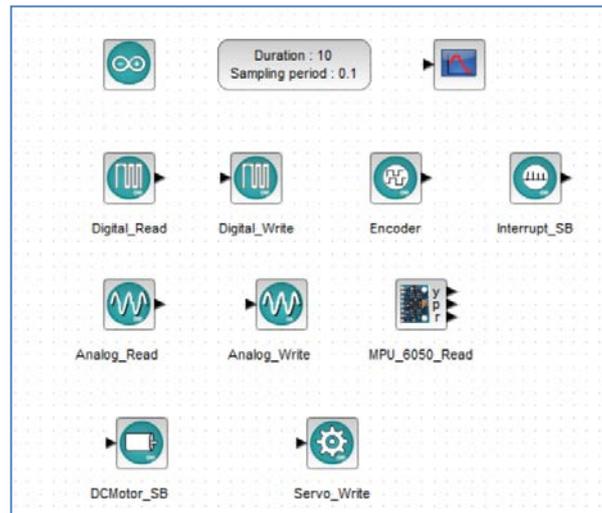


Fig 3. Módulo XCOS para Arduino

4. Listado de ejercicios experimentales ensayados

Presentamos a continuación una lista de tareas experimentales desarrolladas con la herramienta y equipo anteriormente descritos:

1. Control de la velocidad del motor en lazo abierto.
2. Diseño y control PD discreto de posición en tiempo real; comparativa con la simulación mediante la función de transferencia.
3. Diseño y control PI discreto de la velocidad angular; comparativa con la simulación mediante la función de transferencia.
4. Control LQR (comparativa entre simulación y control en tiempo real).
5. Control y simulación LQI (LQR con integrador).
6. Simulación de control con observador discreto.

Todas estas tareas didácticas han sido planificadas y ensayadas con éxito. A lo largo de este artículo (apartado 5) se detallarán.

5. Modelado del motor DC

A partir de las ecuaciones diferenciales que definen el comportamiento físico de un motor de corriente continua, y suponiendo una autoinducción despreciable ($L=0$), se obtiene la representación en el espacio de estados continuo del motor, con variables de estado (θ, ω) :

$$Ki = J\ddot{\theta} + b\dot{\theta} \Rightarrow \ddot{\theta} = -\frac{b}{J}\dot{\theta} + \frac{K}{J}i \quad [1]$$

$$V - K\dot{\theta} = Ri + Li\dot{=} Ri \text{ ya que } L=0$$

$$\text{Despejando } i: \quad i = \frac{V - K\dot{\theta}}{R}$$

$$\text{Sustituyendo } i \text{ en [1]: } \ddot{\theta} = -\frac{b}{J}\dot{\theta} + \frac{K}{J}i = -\frac{b}{J}\dot{\theta} + \frac{K}{J}\left(\frac{V - K\dot{\theta}}{R}\right)$$

$$\ddot{\theta} = -\left(\frac{b}{J} + \frac{K^2}{RJ}\right)\dot{\theta} + \frac{K}{RJ}V$$

En forma matricial :

$$\begin{pmatrix} \dot{\theta} \\ \ddot{\theta} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & -\left(\frac{b}{J} + \frac{K^2}{RJ}\right) \end{pmatrix} \begin{pmatrix} \theta \\ \dot{\theta} \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{K}{RJ} \end{pmatrix} V$$

$$\dot{\theta} = (1 \ 0) \begin{pmatrix} \theta \\ \dot{\theta} \end{pmatrix}$$

donde J es el momento de inercia del eje, K es la constante de fuerza electromotriz cuyo valor numérico es el mismo que el valor de la constante de par motor ($K=K_t=K_e$), R es la resistencia del bobinado, L la autoinducción, b el coeficiente de fricción, y V la entrada de control (voltaje aplicado al motor).

La función de transferencia con respecto a la posición angular es:

$$\frac{\theta(s)}{V(s)} = \frac{K}{s[JLs^2 + (RJ + Lb)s + (Rb + K^2)]} = \frac{K'}{s(Ts + 1)} \quad (L=0)$$

$$K' = \frac{K}{Rb + K^2} \quad T = \frac{RJ}{Rb + K^2} \text{ constante mecánica}$$

y la función de transferencia con respecto a la velocidad angular:

$$\frac{\omega(s)}{V(s)} = \frac{K}{JLs^2 + (RJ + Lb)s + (Rb + K^2)} = \frac{K'}{Ts + 1}$$

Para el motor utilizado, se obtuvo experimentalmente: $K'=4.56$ y $T=0.07$

6 Resultados obtenidos

A continuación se describen algunos de los módulos experimentales y los resultados obtenidos.

6.1 Control del motor en lazo abierto y lectura del ángulo de giro del motor

En la figura 4 se presenta un módulo realizado con XCOS para el control de velocidad del motor en lazo abierto. Se puede seleccionar el sentido de giro mediante la activación de dos pines de salida digital; asimismo, la velocidad se regula con una salida PWM.

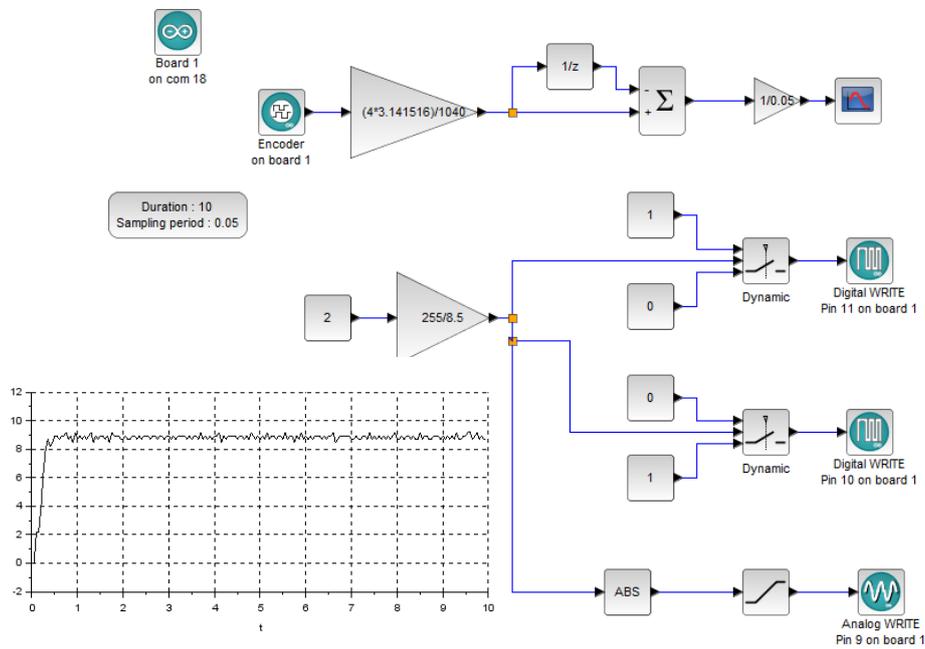


Fig 4. Control de la velocidad y sentido de giro del motor en lazo abierto y lectura del ángulo de giro.

6.2 Lectura de la velocidad de giro w del motor

A partir de la medida del ángulo de giro en radianes, se lleva a cabo la derivada mediante el método Backward-Euler para obtener la medida de la velocidad angular en radianes/segundo (ver figura 5). El periodo de muestreo seleccionado es de 0.05 segundos.

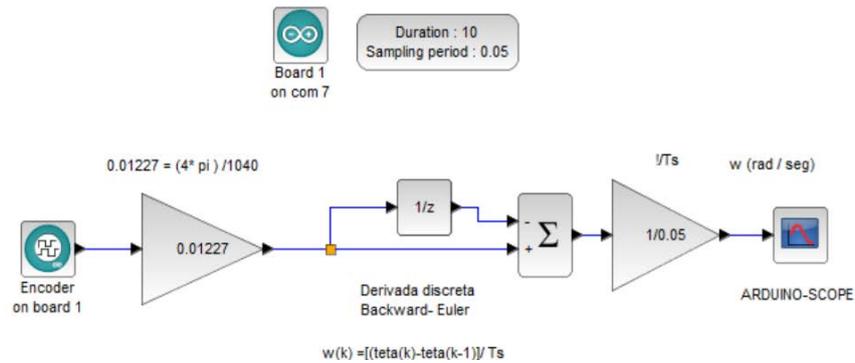


Fig 5. Medida de la velocidad de giro del motor.

6.3 Control PD de la posición angular del motor (servo)

Los parámetros del controlador PD (constantes proporcional y diferencial) se diseñan a partir de la función de transferencia del motor y en función de la respuesta deseada. Se usa el método de diseño por ubicación de los polos de la respuesta final deseada, a partir del sobredisparo máximo (M_p) y el tiempo de asentamiento (t_s) requeridos. Presentamos a continuación un programa Scilab para realizar dicho diseño y representar gráficamente la respuesta del sistema controlado:

```

R=3.8; L=0.0037; K=0.2; b=0.001; T=0.07;
J=(T*(R*b+K^2))/R; Kprima=K/(R*b+K^2);
s=%s;
G=Kprima/(s*(T*s+1));
Ts=0.01; // periodo de muestreo
Mp=0.001; ts=0.5;
xi=abs(log(Mp)/sqrt(%pi^2+log(Mp)^2)); wn=4/(xi*ts);
// Calculo de Kp y Kd Kp=((wn^2)*T)/Kprima;
Kd=(2*xi*wn*T-1)/Kprima;
// Simulacion
fdtPD=Kd*s+Kp; // f.d.t del controlador PD
sysc=K/(J*L*s^3+(R*J+L*b)*s^2+(R*b+K^2)*s);
fdtlazoc=(fdtPD*sysc)/.1; // Realimentación negativa
t=[0:0.01:3];
r=10+0*t; // consigna de ángulo deseado (10 rad)
y=csim(r,t,fdtlazoc);
plot(t,y);

```

Una vez obtenidas las constantes K_p y K_d del controlador PD, este se implementó de nuevo con XCOS, visualizando en tiempo real la respuesta ante un cambio de consigna periódico. Además, en el mismo ejemplo se realizó una comparativa con la respuesta simulada sobre la función de transferencia (ver figura 6).

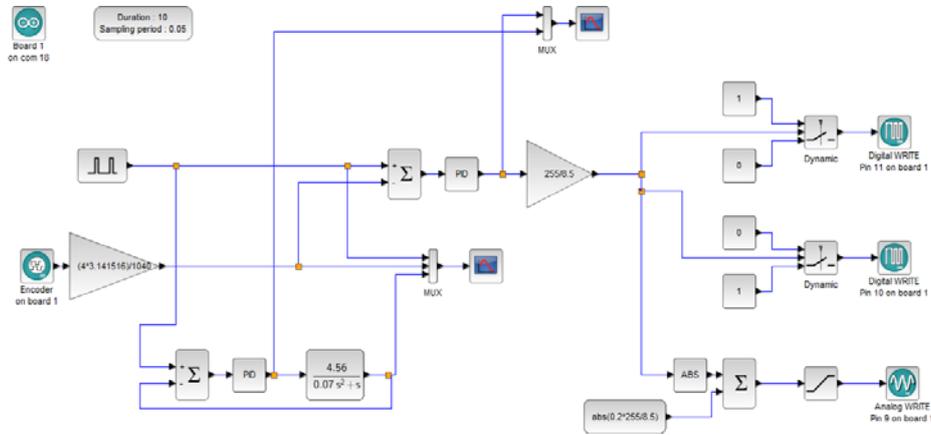


Fig.6 Control PD de la posición angular, realizado con XCOS

Se obtuvieron los siguientes resultados experimentales:

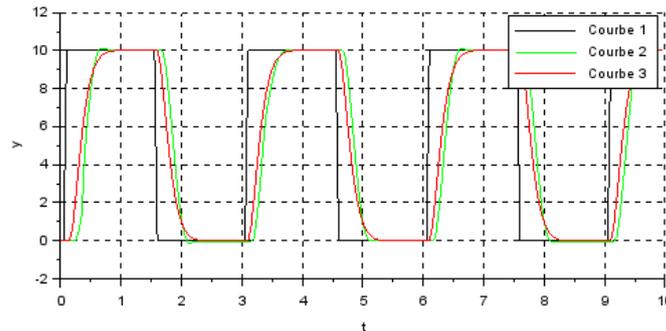


Fig.7 Consigna (en negro), ángulo del eje estimado (en rojo) y ángulo real controlado (en verde)

La respuesta es casi idéntica. Se detecta un pequeño retardo debido al retardo de la comunicación serie con la tarjeta Arduino. Hay que tener en cuenta también que se compara la respuesta teórica continua con la respuesta real del sistema discreto, con una baja frecuencia de muestreo.

6.4 Control PI de la velocidad angular

En la figura 8 se puede apreciar el diseño XCOS del control Proporcional-Integral (PI) de la velocidad w del motor en tiempo real, con un periodo de muestreo de 0.05 seg, así como su simulación empleando la función de transferencia continua del motor.

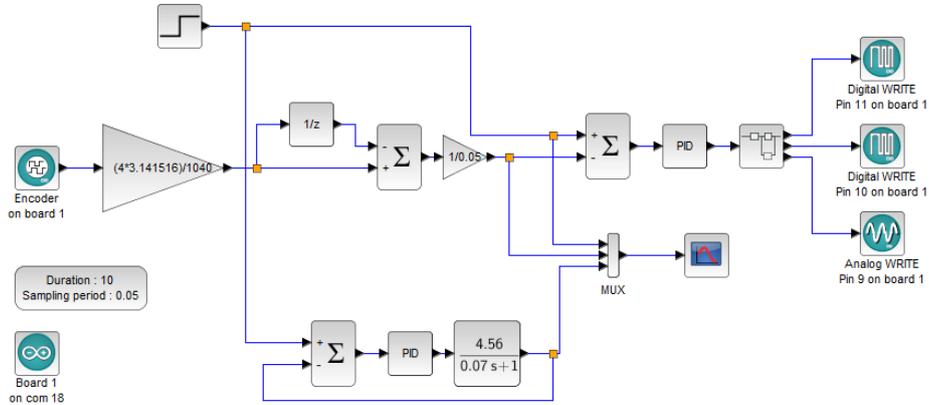


Fig.8 Control y simulación en tiempo real de la velocidad angular del motor

Se puede observar (figura 9), que salvo el ruido registrado en la gráfica experimental, las respuestas en tiempo real (en verde) y simulada (en rojo) son muy similares.

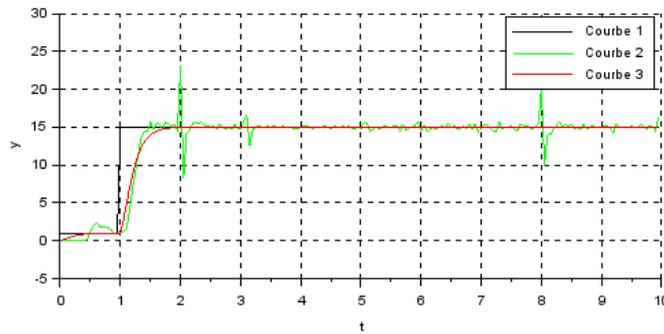


Fig.9 Consigna, señales real y simulada de la velocidad angular controlada

6.5 Control LQR de la posición angular

Con el módulo XCOS de la representación en el espacio de estados (ver figura 10) descrita en el apartado 5, hemos implementado un sistema de control de posición angular por realimentación en el espacio de estados (ver figura 11), donde las constantes de realimentación se calculan mediante un Regulador Cuadrático Lineal (LQR) discreto que optimiza la respuesta de control, minimizando además el coste energético [7].

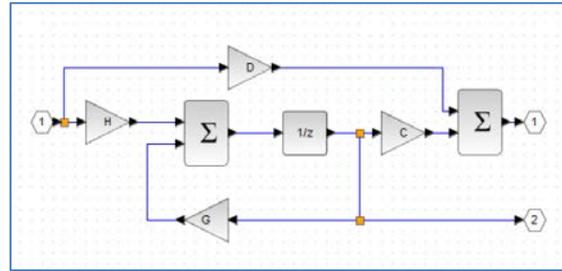


Fig 10 Módulo para la representación de estados discreta en XCOS

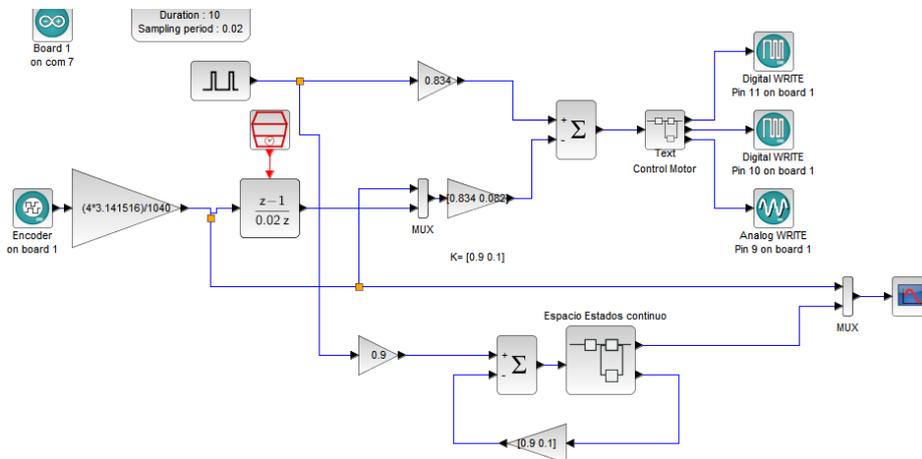


Fig 11 Comparativa entre control LQR del servo motor, real y simulado, usando XCOS .

Se puede apreciar que las respuestas simulada y real se aproximan muy bien (ver figura 12).

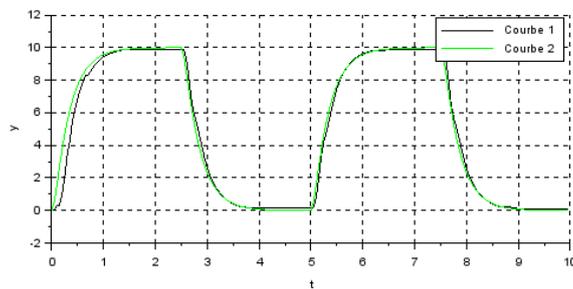


Fig 12. Señales obtenidas en el control LQR del servo motor.

6.6 Control LQI (LQR con Integrador) de la velocidad angular

Dado que la función de transferencia que relaciona la velocidad angular con el voltaje aplicado al motor es de tipo 1 (no tiene integradores), será necesario incorporar una acción integral discreta para conseguir una respuesta adecuada en régimen transitorio. Para ello hemos ensayado con XCOS la configuración de simulación presentada en la figura 13.

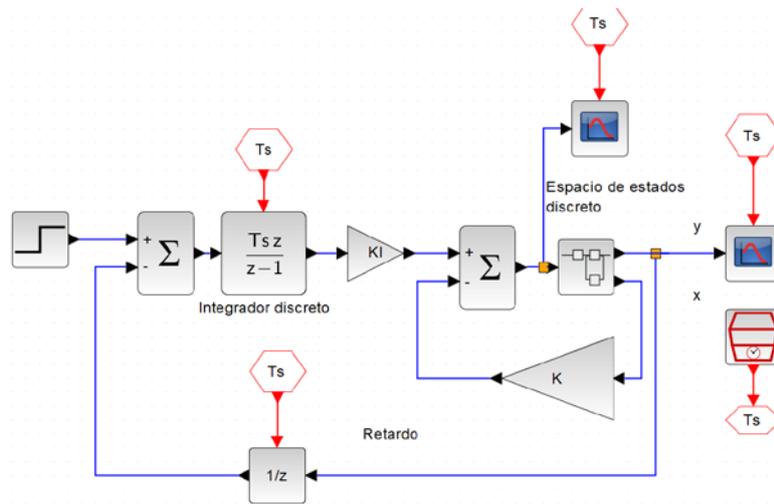
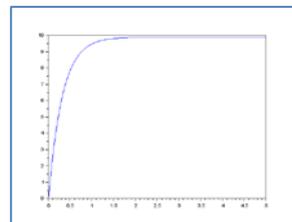


Fig 13. Configuración LQI simulada con XCOS.

Para el diseño de las constantes K, y de la ganancia KI, hemos desarrollado el siguiente programa Scilab:

```
// LQI-Integrador discreto w motor
R=3.8; L=0.0037; K=0.2; b=0.001; T=0.07; J=(T*(R*b+K^2))/R;
Kprima=K/(R*b+K^2);
//Modelo del motor simplificado (L=0)
A = [0 1; 0 -(b/J)-(K^2)/(R*J)]; B = [0; K/(R*J)]; C = [0 1]; D = [0];
sysc=syslin('c',A,B,C,D); Ts=0.001;
sysd=dscr(sysc,Ts); // discretizamos el sistema
G = sysd.a; H = sysd.b; C = sysd.c; D = sysd.d;
//matrices discretas para calculo posterior de K
GG=[G zeros(2,1);-Ts*C 1]; HH=[H;0]; CC=[C 0]; DD=[0];
sysGG=syslin('d',GG,HH,CC,DD);
Q=diag([0.1,1,10]); R=0.1;
KK=-lqr(sysGG,Q,R); // KK=lqi(sysd,Q,R) OPCIONAL
KI=-KK(3); K=KK(1:2)-(KI*Ts*C); // KK=[(K+KI*Ts*C) -KI]
// Matrices para simulación
```



```

GGG=[G-H*(K+KI*Ts*C) H*KI;-Ts*C 1]; HHH=[H*KI*Ts;Ts]; CCC=[C 0]; DDD=[0];
x0 = [0;0;0]; // 3 Condiciones iniciales. Añadimos integrador
sys_fd=syslin('d',GGG,HHH,CCC,DDD,x0); t=[0:Ts:2];
r=10+0*t; // Consigna de 10 rad /seg
y=dsimul(sys_fd,r); // y discreto
x=ltitr(GGG,HHH,r,x0); // x discreto
plot(t,y); // dibujar la salida y
u=-K*x((1:2),:)+KI*x(3,:);
plot(t,u); // dibujar la señal de control
    
```

Con los valores de las constantes de realimentación obtenidas, hemos realizado además un control LQI en tiempo real (ver figura 14).

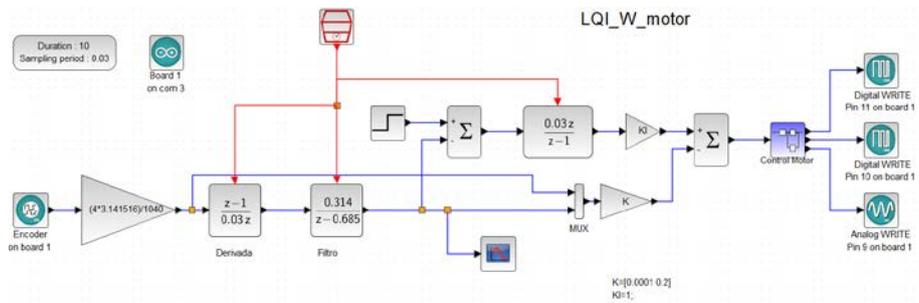


Fig 14. Configuración LQI de control de w en tiempo real

En la figura 15 se puede apreciar las respuestas simulada y real para una velocidad angular deseada de 10 rad/seg. La respuesta real tiene mucho ruido, a pesar del filtrado pasa baja realizado. Se debe principalmente a la baja velocidad de muestreo que permite la comunicación XCOS con la tarjeta Arduino Mega.

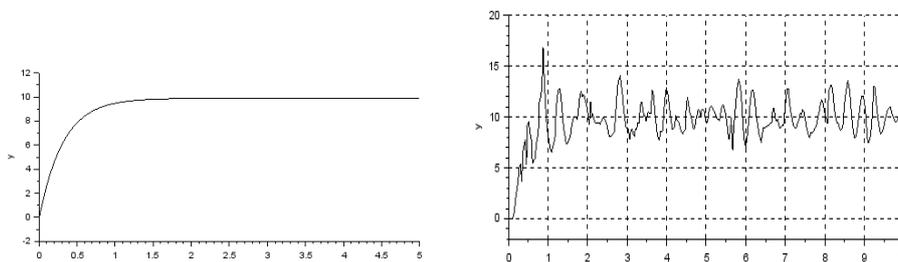


Fig 15. Respuesta simulada y real

6.7 Controlador-Observador

Cuando no puede medirse con precisión alguna de las variables de estado, o hay mucho ruido en las medidas se puede utilizar un observador de estados discreto (siempre que el sistema sea observable) [8], a partir del modelo de estados y de la salida, que en nuestro caso es la posición angular del eje del motor.

El diseño de las constantes L de realimentación del observador y las constantes K de realimentación del controlador se realiza previamente mediante un programa Scilab.

Aunque con nuestra herramienta la programación en tiempo real del controlador-observador no puede llevarse a cabo con frecuencias de muestreo bajas, hemos desarrollado un esquema XCOS para la simulación de este tipo de configuración, aplicada al motor, en el que se ha añadido ruido a la medida de la salida (ver figura 16).

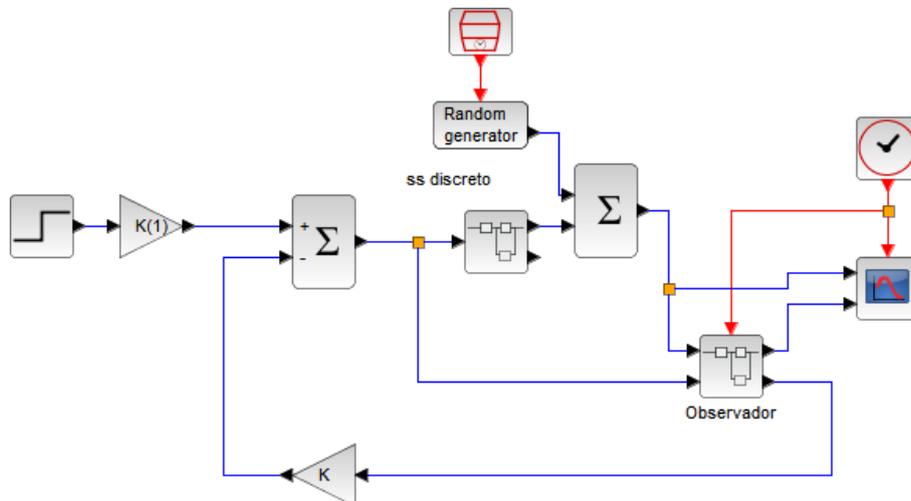


Fig 16. Controlador-observador de la velocidad angular

En la figura 17 se presentan los resultados de la señal de salida, con ruido y filtrada mediante el observador.

7 Conclusiones

Se ha diseñado con Scilab y XCOS un conjunto de experiencias prácticas para el estudio de las principales técnicas de control digital de la posición y velocidad de un motor de corriente continua.

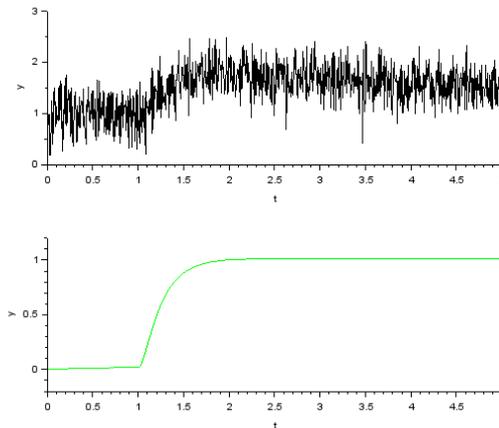


Fig 17. Velocidad angular con ruido y filtrada, simulada con el controlador-observador

Scilab y XCOS son herramientas de libre disposición, con la ventaja que ello conlleva. Su uso es muy intuitivo y la ejecución es muy ágil, prácticamente en cualquier tipo de computador, lo que es ideal para ser utilizado por alumnos. XCOS dispone de librerías para la lectura y el control (vía puerto serie) de entradas y salidas de las tarjetas Arduino más populares (UNO y Mega). De esta manera hemos podido implementar en tiempo real los distintos procedimientos de control discreto, sin apenas problemas.

La principal desventaja de nuestra solución está relacionada con las limitaciones del periodo de muestreo, que como máximo es de 0.03 segundos. Esta limitación no afecta al control PID, LQR o LQI del motor, pero no permite realizar control en tiempo real con observadores o controladores LQG con filtros de Kalman, por ejemplo. Además, tampoco puede usarse para el control real del equilibrio con otros dispositivos mecatrónicos ampliamente utilizados en experiencias didácticas, tales como péndulo invertido, segway, motocicleta autónoma, etc.

Sin embargo, hemos comprobado que los principales fundamentos de la teoría de control sí pueden ser verificados con la solución propuesta, por lo que pretendemos seguir usando este procedimiento en el laboratorio para las asignaturas de control de primer nivel.

8 Referencias

- 1 G. Olivares, A. Olivares, F. Gómez, M. Damas. "Sistema de control de bajo coste de un motor de corriente continua para usos didácticos". Enseñanza y Aprendizaje de Ingeniería de Computadores (2017). ISSN 2173-8688.
2. Kit motor. Accedido el 9/10/2020: <https://es.aliexpress.com/item/32807128872.html?spm=a2g0s.8937460.0.0.58cd2e0eapellz>
3. Arduino. <https://www.arduino.cc>. Accedido el 9/10/2020

4. Driver Toshiba TB6612. Accedido el 9/10/2020:
<https://toshiba.semicon-storage.com/ap-en/semiconductor/product/motor-driver-ics/brushed-dc-motor-driver-ics/detail.TB6612FNG.html>
5. Scilab. <https://www.scilab.org/>. Accedido el 9/10/2020.
6. XCOS. <https://www.scilab.org/software/xcos> . Accedido el 9/10/2020
7. Ogata, N. “Ingeniería de Control Moderna”. ISBN: 9788483226605
8. Bishop. R.H: “Sistemas de Control Moderno”. ISBN-10: 8420544019. 2005.