

UNIVERSIDAD COMPLUTENSE DE MADRID
FACULTAD DE CIENCIAS FÍSICAS



SISTEMAS DINÁMICOS Y REALIMENTACIÓN

CONTROL DE UN MOTOR DE CORRIENTE CONTINUA

Mario de Miguel Domínguez
19 de enero de 2024

Índice

1	Manejo del motor de continua en lazo abierto	1
1.1	Construcción del modelo	1
1.1.1	Identificación de las partes del motor	1
1.1.2	Modelo de Simulink para el manejo en lazo abierto	1
1.2	Identificación del motor	2
1.2.1	Cálculo de la funciones de respuesta	2
1.2.2	Estimación de los valores de k_e y p del motor	3
1.2.3	Comparación de los datos reales con el modelo de Simulink . .	3
2	Control del motor por realimentación de estados estimados	4
2.1	Modelo completo del motor	4

Introducción

El presente documento es un recopilatorio de las respuestas a las cuestiones planteadas en todas las secciones del guion de la práctica de control del motor de Sistemas Dinámicos y Realimentación. Todos los datos experimentales y los modelos se han obtenido o probado con el motor 02.

1 Manejo del motor de continua en lazo abierto

El propósito de esta sección consiste en la elaboración de un modelo del motor que permita leer los valores esperados de la posición y velocidad a través de encoders, en función de una tensión fija aplicada.

1.1 Construcción del modelo

1.1.1 Identificación de las partes del motor

A continuación se incluye una imagen en la que se muestran los diferentes componentes del motor con el que se va a trabajar a lo largo de todas las prácticas.

1. Motor EMG30
2. Placa de desarrollo
3. Placa de expansión
4. Fuente de alimentación

1.1.2 Modelo de Simulink para el manejo en lazo abierto

Para este modelo se empleará un bloque que simulará el motor, que recibirá dos entradas: una analógica representando la tensión que recibe el motor, y una digital que controlará el sentido de giro del motor. A su vez, el motor devolverá dos salidas, representativas de los encoders de posición y velocidad, respectivamente.

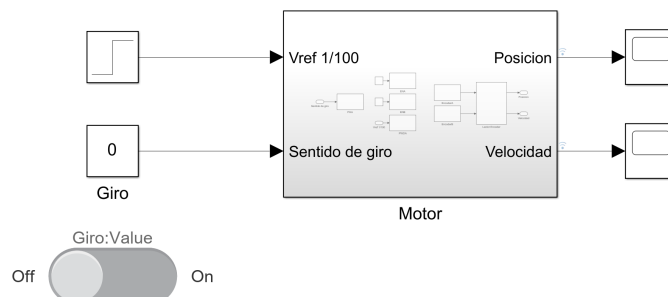


Figura 1: Modelo completo del motor en Simulink

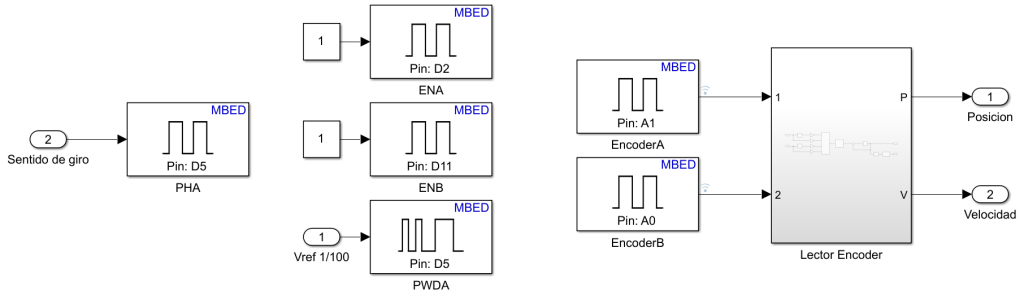


Figura 2: Bloque del motor

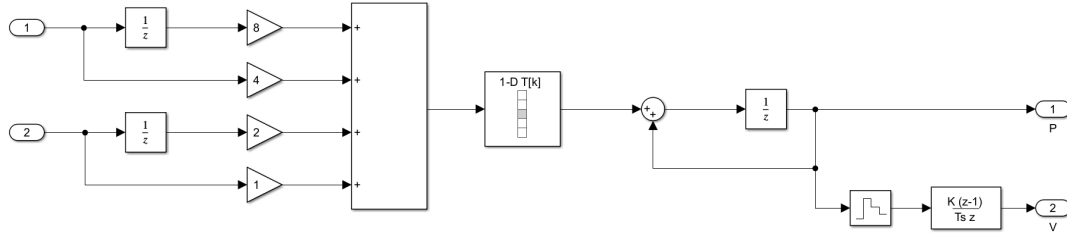


Figura 3: Bloque de lectura de los encoders

1.2 Identificación del motor

1.2.1 Cálculo de la funciones de respuesta

A continuación se calcularán las funciones de respuesta de posición y velocidad del motor a partir del modelo en variables de estado y de la función de transferencia. Para el modelo en variables de estado, se tiene que

$$\begin{bmatrix} \dot{\theta} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -p \end{bmatrix} \cdot \begin{bmatrix} \theta \\ \omega \end{bmatrix} + \begin{bmatrix} 0 \\ k_e \end{bmatrix} V, \quad (1)$$

de donde se deducen las ecuaciones diferenciales

$$\dot{\theta}(t) = \omega(t), \quad (2)$$

$$\dot{\omega}(t) = -p\omega(t) + k_e V. \quad (3)$$

Nótese que hemos definido la tensión de entrada $e(t)$ como una entrada escalón de valor V . Basta con resolver las ecuaciones anteriores imponiendo como condición inicial que $\omega(0) = 0, \theta(0) = 0$ para obtener las expresiones de respuesta:

$$\omega(t) = V \frac{k_e}{p} - \frac{k_e}{p} e^{-pt}, \quad (4)$$

$$\theta(t) = V \frac{k_e}{p^2} e^{-pt} + V \frac{k_e}{p} t - V \frac{k_e}{p^2}. \quad (5)$$

Inmediatamente se comprueba que la expresión 5 es la misma que la ecuación 3.14 del guion. Para calcularlo a partir de la función de transferencia de la posición

$$\theta(s) = \frac{k_e}{s(s+p)} e(s) \quad (6)$$

primero hay que calcular la transformada de Laplace de la entrada $e(t) = V$. Como V es una constante, fácilmente se obtiene $e(s) = \mathcal{L}(V) = \frac{V}{s}$. Una vez aplicada esta transformación, se pueden calcular las funciones de respuesta, resultando

$$\theta(t) = \mathcal{L}^{-1}[\theta(s)] = \frac{Vk_e}{p}t - \frac{Vk_e}{p^2} + \frac{Vk_e}{p^2}e^{-pt}, \quad (7)$$

$$\omega(t) = \dot{\theta}(t) = \frac{Vk_e}{p} - \frac{Vk_e}{p}e^{-pt}, \quad (8)$$

que son expresiones idénticas a las obtenidas a partir del modelo en variables de estado (ecuaciones 5 y 4).

1.2.2 Estimación de los valores de k_e y p del motor

A continuación se va a tratar de obtener los parámetros del motor utilizando el modelo de Simulink de lazo abierto (Ver sección 1.1.2).

1.2.3 Comparación de los datos reales con el modelo de Simulink

Se presenta en la siguiente figura el modelo desarrollado en Simulink, incluyendo los parámetros p y k_e , construido a partir de las ecuaciones 2 y 3.

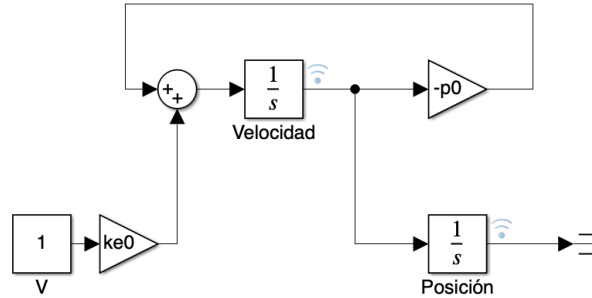


Figura 4: Modelo de Simulink para comprobar los valores obtenidos de k_e y p

Del anterior modelo se observan las señales de salida de los integradores “Velocidad” y “Posición”. Seguidamente se grafican estas señales junto a aquellas obtenidas inicialmente del motor para ciertos valores de tensión, por ejemplo 2, 6 y 12 V, a fin de observar discrepancias entre ellas.

Como se puede apreciar, tanto las señales del modelo como las medidas con el motor real coinciden, dando a entender que los parámetros estimados son adecuados para el motor.

2 Control del motor por realimentación de estados estimados

2.1 Modelo completo del motor

En primer lugar, se procede a construir un modelo más completo que el modelo de motor ideal elaborado al final de la sección anterior, que incluya también un modelo para simular los encoders y la alimentación con señales de PWM. Para la lectura de los encoders, se reutilizará el bloque de lectura empleado para el manejo en lazo abierto.

El sistema de alimentación tratará de simular las señales de PWM (Pulse-Width Modulation) que recibe el motor desde la placa. Entonces, se empieza por construir un bloque que genere una señal con las mismas características. En el modelo final,

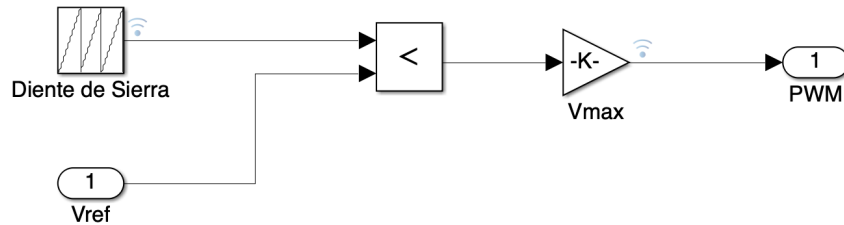


Figura 5: Bloque de generación de señal de PWM

este bloque se encontrará enmascarado y recibirá dos parámetros: T , que dirigirá el periodo del diente de sierra y, en consecuencia, de la señal PWM (1ms) y V_{max} , indicador de la tensión máxima de esta señal (corresponderá con la tensión máxima nominal del motor, 12V).

El sistema de alimentación incluirá también un bloque en el que se ajuste V_{ref} , indicador la tensión continua objetivo del bloque PWM (a menor V_{ref} , menor será el ciclo de trabajo de la señal PWM). Como el ciclo de trabajo se establece a través de una comparación de la tensión de la señal de diente de sierra. Para que el bloque PWM funcione correctamente, se ha de pasar una señal de V_{ref} positiva. Si V_{ref} es negativo, se multiplica por -1 la señal de PWM para que el motor avance en sentido contrario.

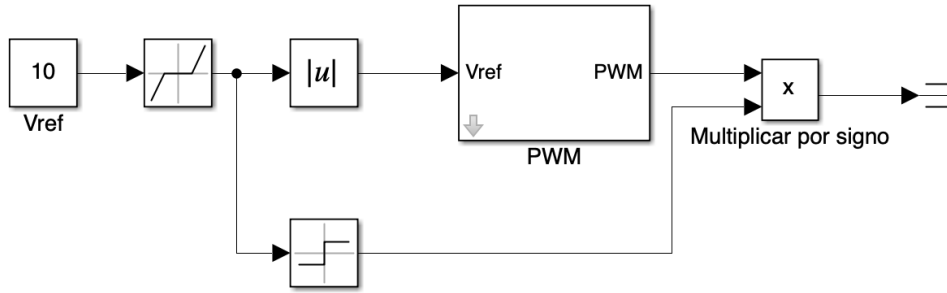


Figura 6: Sistema de alimentación completo

A continuación se añade el modelo del motor ideal desarrollado previamente y se diseñan sus respectivos encoders para poder leer la posición y la velocidad.

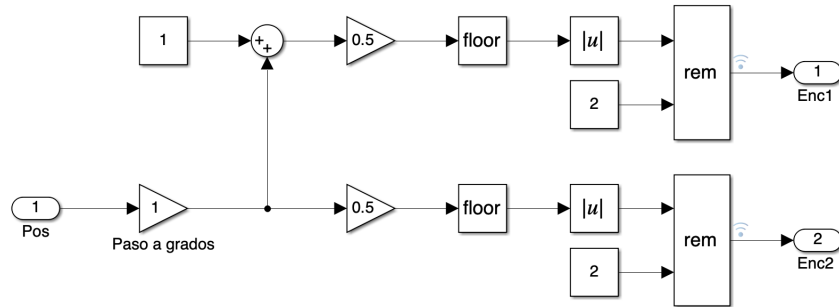


Figura 7: Modelo de los encoders del motor

Por último, añadiendo el modelo reciclado para el lector de los encoders empleado en el lazo abierto, se obtiene el modelo completo del motor real.

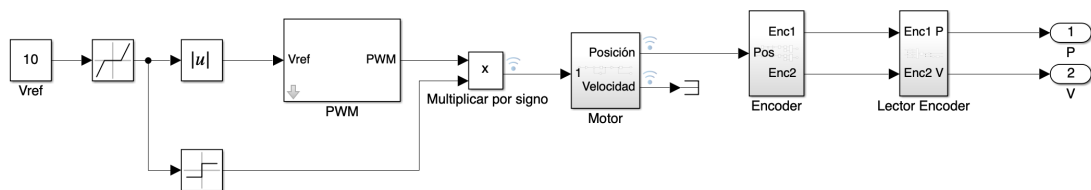


Figura 8: Modelo completo realista del motor

Anexo I: Código empleado para las prácticas

Cálculo de ecuaciones de velocidad y posición del motor

ecs.m

```

1 syms ke s p V w(t) theta(t) E(s) thetalap(t)
2 %Defino las ecuaciones diferenciales para resolver w (
   velocidad),
3 %theta (posición) y las resuelvo
4 %Primero la velocidad
5 eqvel = diff(w, t) == -p*w + ke * V
6 wsol = dsolve(eqvel, w(0) == 0)
7
8 %Y luego la posición
9 eqpos = diff(theta, t) == wsol
10 thetasol = dsolve(eqpos, theta(0) == 0)
11
12 %Calculo la transformada de laplace de e(t) = V
13 E(s) = laplace(V, t, s)
14 %Calculo la transformada inversa de la función de
   transferencia Th(s)
15 Th = ke / (s * (s + p)) * E(s)
16 thetalap(t) = ilaplace(Th)
17
18 wlap = diff(thetalap(t), t)

```

Estimación de los parámetros k_e y p del motor

caracterizador.m

```

1 close all; clear all;
2
3 %Importo los datos medidos del motor
4 dt1 = importdata("datos/motor01V.mat");
5 dt2 = importdata("datos/motor02V.mat");
6 dt3 = importdata("datos/motor03V.mat");
7 dt4 = importdata("datos/motor04V.mat");
8 dt5 = importdata("datos/motor05V.mat");
9 dt6 = importdata("datos/motor06V.mat");
10 dt7 = importdata("datos/motor07V.mat");
11 dt8 = importdata("datos/motor08V.mat");
12 dt9 = importdata("datos/motor09V.mat");
13 dt10 = importdata("datos/motor10V.mat");
14 dt11 = importdata("datos/motor11V.mat");
15 dt12 = importdata("datos/motor12V.mat");
16

```

```

17 %Defino una lista con todos los datasets sobre la que voy a
    iterar para sacar los datos
18 dt = [dt1, dt2, dt3, dt4, dt5, dt6, dt7, dt8, dt9, dt10,
        dt11, dt12];
19 p = [];
20 ke = [];
21
22 for i = 1:12
23     m = get(dt(i), "Motor:1").Values.Data;
24     t = get(dt(i), "Motor:1").Values.Time;
25     %Para hacer la estimación, voy a empezar a contar justo
        antes de que los datos de posición cambien de 0. Como
        los datos a 6 y 10 V marcan 1 desde el principio,
        resto 1 a la posición para que la búsqueda luego
        funcione bien.
26     if i == 6 || i == 10
27         m(2:end) = m(2:end) - 1;
28     end
29
30     %Encontrar los valores no cero de la posición y empezar
        a contar justo antes
31     a = find(m);
32     st = a(1) - 5
33     mprima = m(st:end);
34
35     %Hago el fit con los datos recortados y traslado los
        tiempos para que vayan de 0 a X
36     pol = polyfit(t(1:length(mprima)), mprima, 1);
37
38     %Estimo los parámetros para este voltaje y los guardo en
        una lista
39     ke(i) = -pol(1)^2 / (pol(2) * i);
40     p(i) = (ke(i) * i)/pol(1);
41
42
43     %figure(i)
44     %plot(t, m); hold on;
45     %plot(t, polyval(pol, t))
46 end
47
48 %Imprimo las listas de valores y las medias
49 p
50 ke
51 p0 = mean(p)
52 ke0 = mean(ke)

```

Controlador por realimentación de estados estimados

realimentador.m

```

1 p = 63.7990;
2 ke = 6.3065e+03;
3 A = [0 1; 0 -p]
4 B = [0; ke] %Solo controlo la velocidad
5 K = place(A, B, [-0.3 * p, -0.4 * p])
6
7 %u = 1 Vref
8 %Diseño del estimador
9 C = [1, 0];
10 L = acker(A', C', [-1.2*p, -1.2*p])'
11
12 Aamp = [A zeros(2, 1); C 0]
13 Bamp = [B; 0]
14
15 Ki = acker(Aamp, Bamp, [-0.5*p, -0.5*p, -0.7*p])

```

Discretización del controlador para el modelo de motor real

discretizador.m

```

1 warning("off")
2
3 %Crear modelo del sistema del motor
4 sys = ss(A, B, C, 0)
5 %Discretizar el modelo: c2d(modelo, T) (T = 10^-h s)
6 T = 1e-4
7 sysd = c2d(sys, T)
8
9 %Crear el control discreto
10 Kd = acker(sysd.A, sysd.B, [exp(-0.3 * p * T), exp(-0.3 * p
    * T)])
11 Ld = acker(sysd.A', C', [exp(-1.2*p * T), exp(-1.2 * p * T)]
    )'
12
13 DAamp = [sysd.A zeros(2, 1); T * C, 1]
14 DBamp = [sysd.B; 0]
15
16 Kid = acker(DAamp, DBamp, [exp(- p * T), exp(-0.5 * p * T),
    exp(-0.7*p*T)])

```