



FACULTAD DE CIENCIAS EXACTAS, FÍSICAS y NATURALES

# **Informe de Actividad N°1**

Representación de Sistemas y Controladores

**Facundo Nahuel Galvagno**

Sistemas de Control II  
Universidad Nacional de Córdoba  
2024

## Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Desarrollo</b>	<b>2</b>
2.1. Caso de estudio 1. Sistema de dos variables de estado . . . . .	2
2.2. Caso de estudio 2. Sistema de tres variables de estado . . . . .	8
<b>3. Conclusiones</b>	<b>19</b>
<b>4. Lecciones aprendidas</b>	<b>20</b>
<b>5. Recursos</b>	<b>20</b>

## 1. Introducción

En este documento se detalla el desarrollo de la primera actividad práctica propuesta por el profesor Julián Pucheta. Para el procesamiento computacional se utilizó MATLAB R2018a. La primicia del trabajo es la de utilizar fundamentos de control sobre sistemas reales sobre los cuales se realizaron mediciones para obtener parámetros y controlarlos.

La metodología para la resolución de los problemas se basó en el código y la teoría provistos en los apuntes de clase.

## 2. Desarrollo

### 2.1. Caso de estudio 1. Sistema de dos variables de estado

Sea el sistema eléctrico de la Fig. 1-1, con las representaciones en variables de estado

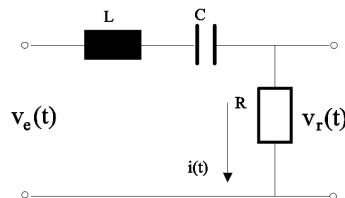


Figura 1: Circuito RLC

$$\dot{X} = Ax(t) + Bu(t)$$

$$y = C * x(t)$$

donde las matrices contienen a los coeficientes del circuito,

$$A = \begin{bmatrix} -R/L & -1/L \\ 1/C & 0 \end{bmatrix}, B = \begin{bmatrix} 1/L \\ 0 \end{bmatrix}, C = [R \ 0] \quad (1)$$

**Ítem (1)** Asignar valores a  $R = 47\Omega$ ,  $L = 1\mu H$  y  $C = 100nF$ . Obtener simulaciones que permitan estudiar la dinámica del sistema, con una entrada de tensión escalón de 12V, que cada 1ms cambia de signo.

Para realizar la simulación del circuito RLC es indispensable conocer la dinámica del mismo. Podemos obtener el valor de la dinámica más rápida y más lenta en base a los autovalores de la matriz A, que son los polos de la función de transferencia del sistema.

```

1 R=47; L=1e-06; C=100e-09; %Parametros del circuito RLC
2 A1 = [-R/L -1/L; 1/C 0];
3 B1 = [1/L; 0];
4 C1 = [R 0];
5 C2 = [0 1];
6 D1 = [0];
7
8 eig_val = eig(A1)
```

Entonces, los autovalores de la matriz A son:

$$\lambda_1 = -4,57 \times 10^7, \lambda_2 = -2,13 \times 10^5$$

Para calcular el tiempo de integración y simulación, podemos encontrar el tiempo al que corresponde el 95 % de la dinámica más rápida a partir de los polos de la función de transferencia del sistema. Notar que el tiempo de simulación debe ser el suficiente como para simular de correcta forma la entrada requerida, por lo tanto, en lugar de observar la dinámica más lenta del sistema, simularemos el tiempo suficiente para que la entrada de tensión conmute.

$$t_r = \frac{\log(0,95)}{\lambda_1}$$

$$t_i \approx t_r/10 = 1 \times 10^{-10}$$

$$t_s = 5 \times 10^{-3}$$

```

1 %% Tiempo de integracion y simulacion
2 tr=log(0.95)/eig_val(1);
3
4 %Tomamos ti = 1e-10 (10 veces menor)
5 ti = 1e-10
6
7 %Tomamos ts = 5e-03 debido a que buscamos simular la
8 %respuesta con una entrada que conmuta
9 ts = 5e-03
10
11 N = ts/ti;
```

La cantidad de muestras de nuestro vector tiempo es:

$$N = 5 \times 10^7$$

Procedemos a simular el comportamiento del sistema mediante integración por Euler.

```

1 %% Entrada y vector de tiempo
2 t = linspace (0, ts, N);
3 u = linspace (0, 0, N);
4
5 %Funciones i_c, v_c
6
7 i_c=zeros(1,N);
8 v_c=zeros(1,N);
9
10
11 x = [0 0]'; %Cond. iniciales nulas
12 vin = 12;
13 ij = 0;
14
15 for ii=1:N-1
16     ij=ij+ti;
17     if (ij>=1e-3)
18         ij=0;
19         vin=vin*-1;
20     end
21     u(ii)=vin;
22     xp=A1*x+B1*u(ii);
23     x=x+xp*ti;
24     Y=C1*x;
25     y(ii+1)=Y(1);
26     i_c(ii+1)=x(1);
```

```

27     v_c(ii+1)=x(2);
28 end
29
30 figure('Name','1.1')
31 subplot(3,1,1);plot(t,i_c);grid on; title('Corriente, I');
32 subplot(3,1,2);plot(t,v_c);grid on; title('Tension del capacitor, V_c');
33 subplot(3,1,3);plot(t,u);grid on; title('Tension de entrada, V_e');

```

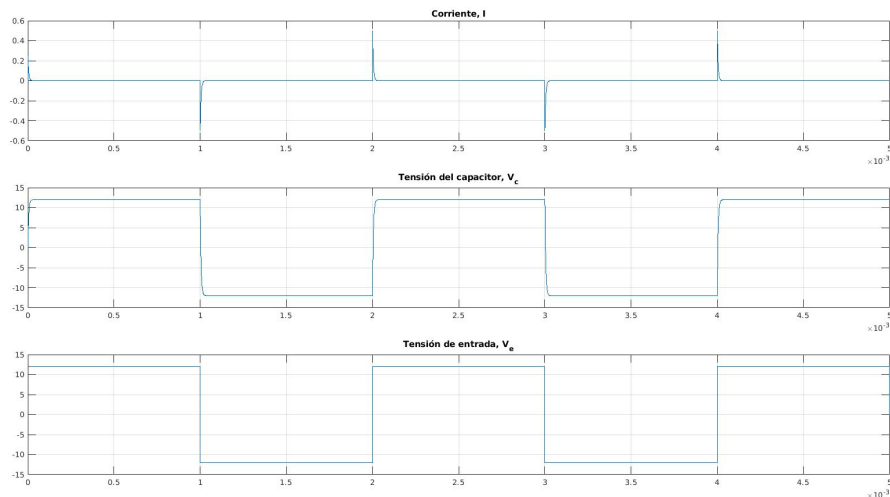


Figura 2: Respuesta del circuito RLC

**Ítem (2)** En el archivo *Curvas\_Medidas\_RLC.xls* (datos en la hoja 1 y etiquetas en la hoja 2) están las series de datos que sirven para deducir los valores de  $R$ ,  $L$  y  $C$  del circuito. Emplear el método de la respuesta al escalón, tomando como salida la tensión en el capacitor.

```

1 datos = xlsread('Curvas_Medidas_RLC_2024.xlsx');
2 t = datos(:,1);
3 i = datos(:,2);
4 v_c = datos(:,3);
5
6 figure('Name','1.2')
7 subplot(2,1,1);plot(t,i);grid on; title('Corriente');
8 subplot(2,1,2);plot(t,v_c);grid on; title('Tension en el capacitor, V_c1');

```

En condiciones iniciales nulas, al energizar el circuito el capacitor y el inductor se comportarán como un corto circuito durante un breve periodo hasta que finalice el transitorio. Es por esto que podemos realizar una primera aproximación de la resistencia del circuito observando la amplitud del primer pico de corriente y la tensión de alimentación.

Esta primera aproximación da como resultado:

$$R_{aprox} = \frac{V_{max}}{I_{max}} = 335\Omega$$

Según las ecuaciones matemáticas que modelan al sistema RLC,

$$\frac{di}{dt} = -\frac{R}{L}i - \frac{1}{L}v_c + \frac{1}{L}v_e$$

$$\frac{di}{dt} = \frac{1}{C}i$$

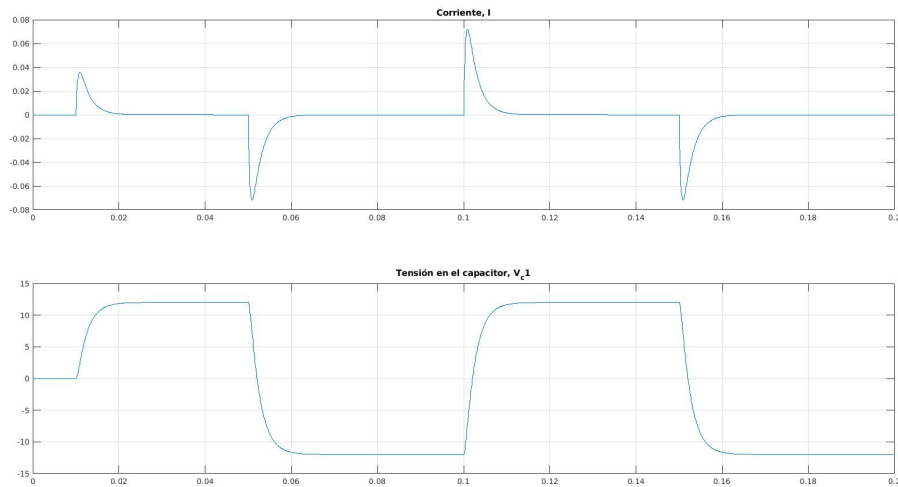


Figura 3: Datos medidos del circuito RLC

podemos obtener la función de transferencia del sistema

$$G(s) = \frac{I(s)}{V_e(s)} = \frac{1}{LCs^2 + CRs + 1}$$

Se logra observar que la función de transferencia del sistema es de segundo orden.

Para obtener el valor de  $L$  y  $C$  es necesario identificar la función de transferencia del sistema medido. Se implementa el método de Chen tomando tres puntos equidistantes que abarquen la mayor parte del transitorio.

```

1  % Determinamos el modelo del circuito RLC a partir del metodo desarrollado
2  % por Chen
3
4  % td: tiempo entre puntos equidistantes para el algoritmo de chen
5  % t1: tiempo inicial para el algoritmo
6  td = 0.01;
7  t1 = 0.002;
8  k=max(v_c);
9  [val lugar] = min(abs((t1+td)-t))
10
11  k1 = v_c(lugar)/k-1
12  [val lugar] = min(abs((2*t1+td)-t))
13  k2 = v_c(lugar)/k-1
14  [val lugar] = min(abs((3*t1+td)-t))
15  k3 = v_c(lugar)/k-1
16
17  b=4*k1^3*k3-3*k1^2*k2^2-4*k2^3+k3^2+6*k1*k2*k3;
18  alfa1 = (k1*k2+k3-sqrt(b))/(2*(k1^2+k2));
19  alfa2 = (k1*k2+k3+sqrt(b))/(2*(k1^2+k2));
20  beta= (2*k1^3+3*k1*k2+k3-sqrt(b))/sqrt(b);
21
22  T1 =-t1/log(alfa1);
23  T2 =-t1/log(alfa2);
24  %T3 = beta*(T1-T2)+T1
25

```

```

26 s = tf('s');
27
28 %sys_id = k*(T3*s+1)/(T1*s+1)/(T2*s+1)
29 sys_id = k/(T1*s+1)/(T2*s+1)

```

La función de transferencia del sistema identificado es:

$$G_{id}(s) = \frac{1}{9,865 \times 10^{-7} s^2 + 2,69 \times 10^{-3} + 1}$$

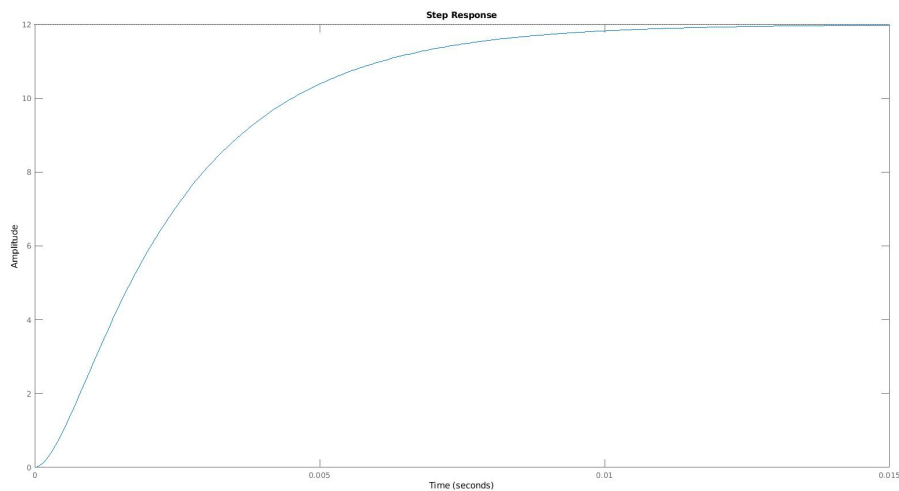


Figura 4: Respuesta al escalón del sistema identificado

A partir de la respuesta al impulso, podemos suponer que posee dos polos reales y distintos y ningún cero, por lo que al realizar el algoritmo de Chen es posible descartar el cero compuesto por  $T_3$ .

Al comparar la función de transferencia del sistema identificado y la obtenida de las ecuaciones matemáticas que lo modelan, se puede despejar los valores restantes.

```

1 [N D] = tfdata(sys_id, 'v');
2
3 C = D(2)/R;
4 L = D(1)/C;

```

Obteniendo los valores de  $R = 335\Omega$ ,  $L = 122,9mH$  y  $C = 8,03\mu F$

**Ítem (3)** Una vez determinados los parámetros R, L y C, emplear la serie de corriente desde 0.05seg en adelante para validar el resultado superponiendo las gráficas.

A partir del modelo de variables de estados realizamos una simulación utilizando *lsim*.

```

1 [u,t] = gensig("square",0.08,0.16,0.00005);
2 u = -(24*u-12);
3 A4 = [-R/L -1/L; 1/C 0];
4 B4 = [1/L; 0];
5 C4 = [1, 0];
6 D4 = [0];
7
8 % Creamos modelo de espacio de estados
9 sys_est = ss(A4,B4,C4,D4);
10

```

```

11 [i_est,t_sim]=lsim(sys_est,u,t);
12
13 figure('Name','1.3')
14 hold on
15 plot(t, i_est)
16 plot(datos(:,1)-0.01,datos(:,2));grid on; title('Corriente');
17 legend('RLC estimado','RLC medido')
18 xlim([0 0.08])
  
```

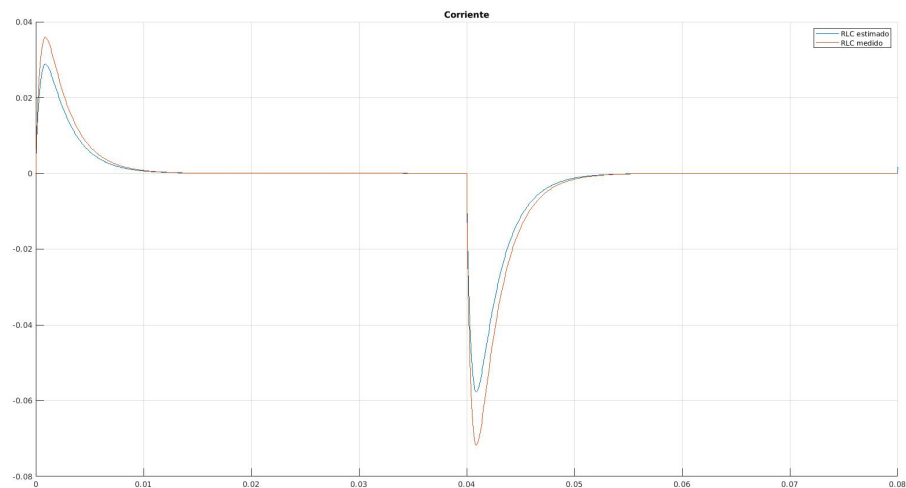


Figura 5: Sistema identificado vs sistema medido

A partir de estos resultados se puede iterar sobre el valor de  $R$  para llegar a una aproximación mas correcta del valor de la misma. Si modificamos  $R = 270\Omega$  y volvemos a calcular los parámetros según el procedimiento anterior, obtenemos:

$$R = 270\Omega$$

$$L = 99mH$$

$$C = 9,96\mu F$$

Obteniendo una mejor aproximación, como se muestra en la Figura 6.



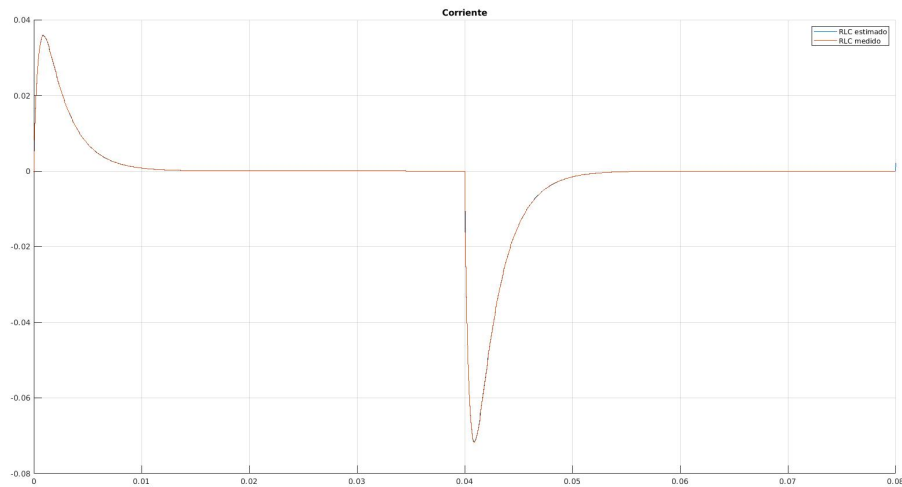


Figura 6: Sistema identificado vs sistema medido, con una mejor aproximación

## 2.2. Caso de estudio 2. Sistema de tres variables de estado

Dadas las ecuaciones del motor de corriente continua con torque de carga  $T_L$  no nulo, con los parámetros  $L_{AA} = 366 \times 10^{-6}$ ;  $J_m = 5 \times 10^{-9}$ ;  $R_A = 55,6$ ;  $B = 0$ ;  $K_i = 6,49 \times 10^{-3}$ ;  $K_m = 6,53 \times 10^{-3}$

$$\frac{di_a}{dt} = -\frac{R_A}{L_{AA}}i_a - \frac{K_m}{L_{AA}}\omega_r + \frac{1}{L_{AA}}v_a$$

$$\frac{d\omega_r}{dt} = \frac{K_i}{J}i_a - \frac{B_m}{J}\omega_r - \frac{1}{J}T_L$$

$$\frac{d\Theta_t}{dt} = \omega_r$$

Implementar un algoritmo de simulación para inferir el comportamiento de las variables interés mediante integración Euler con  $t_s = 1 \times 10^{-7}$  segundos para calcular su operación con un controlador

**Ítem (4)** Obtener el torque máximo que puede soportar el motor modelado mediante las ecuaciones diferenciales cuando se lo alimenta con 12V, graficando para 5 segundos de tiempo la velocidad angular y corriente  $i_a$  para establecer su valor máximo como para dimensionar dispositivos electrónicos.

Se realiza una simulación mediante Euler para estudiar la corriente pico de consumo del motor.

```
1 %Asignamos los valores correspondientes al motor
2 clear all;
3 close all;
4 Laa=366e-6;
5 J=5e-9;
6 Ra=55.6;
7 B=0;
8 Ki=6.49e-3;
9 Km=6.53e-3;
10 delta=10e-07;
11 ts=5;
```

```

12
13 %Generamos la funcion de entrada de tension Va y TL para el motor.
14
15 Va=12;
16 t=0:delta:(ts-delta);
17 T_L = t>=0;
18 for i=1:ts/delta-delta
19   T_L(i)=0;
20 end
21
22 %Realizamos la integracion por euler
23 ia = zeros(1,ts/delta);
24 wr = zeros(1,ts/delta);
25 o = zeros(1,ts/delta);
26 ia(1)=0; wr(1)=0; o(1)=0; %Cond. iniciales
27 Va=12;
28
29 for i=2:(ts/delta-1)
30   ia(i)=ia(i-1)+delta*(-Ra*ia(i-1)/Laa-Km*wr(i-1)/Laa+Va/Laa);
31   wr(i)=wr(i-1)+delta*(Ki/J*ia(i-1)-B/J*wr(i-1)-T_L(i-1)/J);
32   o(i) = o(i-1) + delta*wr(i-1);
33 end
34
35 figure('Name','Torque vs Corriente')
36 subplot(2,1,1);plot(t,wr);grid on; title('Velocidad angular, W_r');
37 subplot(2,1,2);plot(t,ia);grid on; title('Corriente de armadura, I_a');
  
```

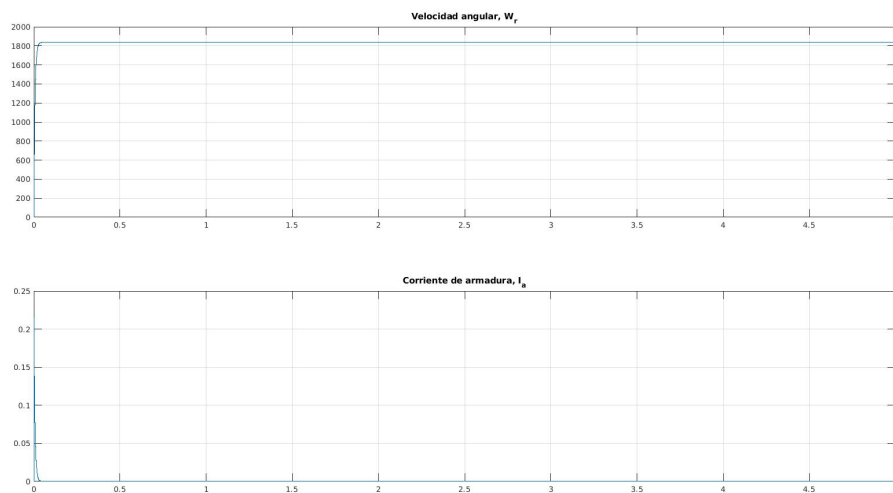


Figura 7: Simulación del motor CC

El consumo máximo de corriente del motor, así como el torque máximo que este nos produce, viene dado por el pico producido en el arranque.

```

1 ia_max = max(ia)
2 wr_max = max(wr)
3 T_L_max = Ki*ia_max-B*wr_max
  
```

$$i_{a,max} = 0,2147A$$

$$T_{L,max} = 0,0014Nm$$

Se puede comprobar que este es el torque máximo si se lo aplica al modelo.

```

1 %Se comprueba que son los valores maximos al aplicar un torque equivalente en
2 %el modelo y observar que la velocidad angular se reduzca a cero.
3
4 T_L = t>=0;
5 for i=1:ts/delta/2
6     T_L(i)=0;
7 end
8 T_L = T_L*0.0014;
9 ia(1)=0; wr(1)=0; o(1)=0; %Cond. iniciales
10 Va=12;
11 for i=2:(ts/delta-1)
12     ia(i)=ia(i-1)+delta*(-Ra*ia(i-1)/Laa-Km*wr(i-1)/Laa+Va/Laa);
13     wr(i)=wr(i-1)+delta*(Ki/J*ia(i-1)-B/J*wr(i-1)-T_L(i-1)/J);
14     o(i) = o(i-1) + delta*wr(i-1);
15 end
16
17 figure('Name', 'W_r al aplicar T_L maximo')
18 plot(t, wr)

```

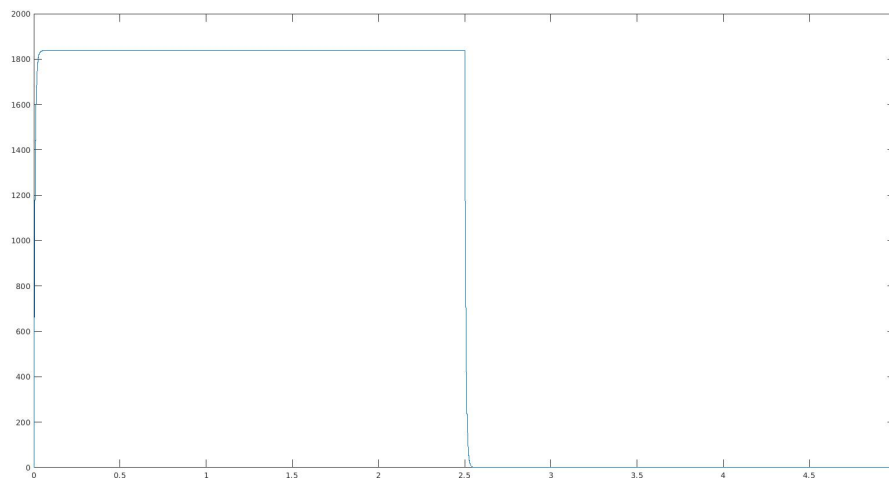


Figura 8: Velocidad angular al aplicar el torque máximo luego de los 2.5 segundos

**Ítem (5)** A partir de las curvas de mediciones de las variables graficadas, se requiere obtener el modelo del sistema considerando como entrada un escalón de 12V, como salida a la velocidad angular, y al torque de carga TL aplicado una perturbación. En el archivo *Curvas\_Medidas\_Motor.xls* están las mediciones, en la primer hoja los valores y en la segunda los nombres. Se requiere obtener el modelo dinámico, para establecer las constantes del modelo

```

1 % Cargamos los datos medidos
2 datos = xlsread('Curvas_Medidas_Motor_2024.xlsx');
3 t = datos(:,1);

```

```

4 w_r= datos(:,2);
5 i_a = datos(:,3);
6 u = datos(:,4);
7 t_l = datos(:,5);
8
9 figure('Name', 'Item 5')
10 subplot(4,1,1);plot(t,i_a);grid on; title('Corriente de Armadura, i_a');
11 subplot(4,1,2);plot(t,w_r);grid on; title('Velocidad Angular, W_r');
12 subplot(4,1,3);plot(t,u);grid on; title('Tension de entrada, V_i');
13 subplot(4,1,4);plot(t,t_l);grid on; title('Torque de carga, T_L');

```

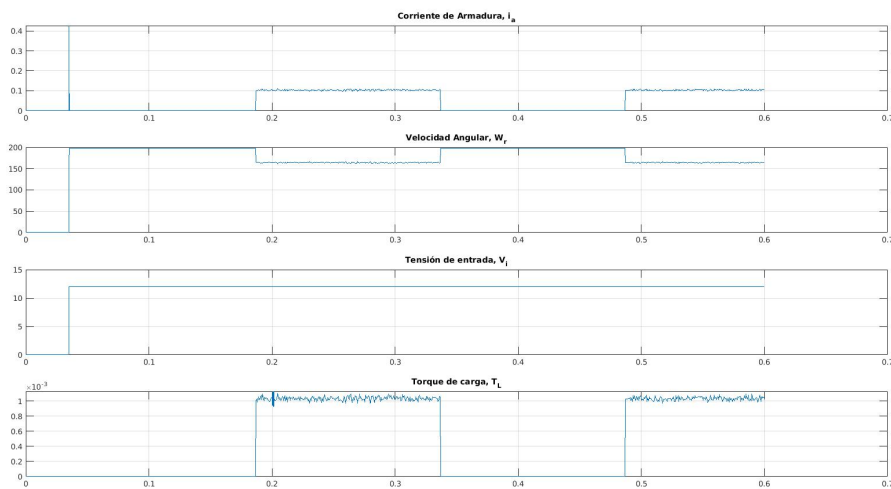


Figura 9:  $I_a$ ,  $\omega_r$ ,  $V_i$ ,  $T_L$  medidos

Para identificar el modelo, primero se obtiene la función de transferencia matemática del motor, a partir de sus ecuaciones diferenciales.

Puesto que  $B_m$  es chico, podemos aproximarlo a 0 para la obtención de los parámetros.

```

1 %% Obtencion de la funcion de transferencia del motor
2 syms V Ra La I Ki w Jm Km Bm Tl s real
3 eq1=s*I==Ra/La*I-Km/La*w+1/La*V;
4 eq2=s*w==Ki/Jm*I-Bm/Jm*w;
5 S1=solve(eq1,eq2,w,V);
6 wr_va=collect(S1.w/S1.V,s);
7
8
9 syms V Ra La I Ki w Jm Km Bm Tl s real
10 eq1=s*I==Ra/La*I-Km/La*w;
11 eq2=s*w==Ki/Jm*I-Bm/Jm*w-Tl/Jm;
12 S1=solve(eq1,eq2,w,Tl);
13 wr_tl=collect(S1.w/S1.Tl,s);
14
15
16 % tomamos Bm = 0
17 wr_va = subs(wr_va, Bm, 0)
18 wr_tl = subs(wr_tl, Bm, 0)
19 pretty(wr_va);

```

```
20 pretty(wr_tl);
```

Las funciones de transferencia que modelan al motor, tomando como salida solamente la velocidad angular  $\omega_r$  son

$$\frac{W_r(s)}{V_a(s)} = \frac{K_i}{J_m L_{AA} s^2 + J_m R_a s + K_i K_m}$$

$$\frac{W_r(s)}{T_L(s)} = \frac{-R_a - L_{AA} s}{J_m L_{AA} s^2 + J_m R_a s + K_i K_m}$$

La ganancia en estado estable de las funciones se puede obtener mediante el teorema del valor final para las mismas. Tomamos los siguientes límites

$$\lim_{s \rightarrow 0} \frac{s W_r(s)}{s V_a(s)} = \frac{1}{K_m}$$

$$\lim_{s \rightarrow 0} \frac{s W_r(s)}{s T_L(s)} = \frac{-R_a}{K_i K_m}$$

```
1 % Segun TVF tenemos
2 limit(wr_va, s, 0)
3 limit(wr_tl, s, 0)
```

A partir de los datos medidos, se puede aproximar  $R_a$  con un método similar al usado para el circuito RLC. Puesto que en el arranque, con condiciones iniciales nulas, el bobinado del motor se comporta como un circuito casi resistivo, la aproximación

$$R_a = V_a / I_{a,max}$$

es válida. Por lo tanto,

```
1 Ra = max(u)/max(i_a)
```

$$R_a = 28,13\Omega$$

A su vez, el valor de la velocidad angular  $\omega_r$  en estado estacionario está determinado por la constante del motor  $K_m$  para la función de transferencia  $\frac{W_r(s)}{V_a(s)}$ , por lo que podemos calcular  $K_m$  a partir del valor de  $\omega_r$  para un torque de carga nulo  $T_L = 0$ .

```
1 Km = max(u)/max(w_r)
```

$$K_m = 60,5 \times 10^{-3}$$

Al aplicar el torque  $T_L$ ,  $\omega_r$  cae de  $198 rad/s$  a  $164 rad/s$ . Por linealidad, se puede inferir que esto es producto de sumar el aporte de la función de transferencia  $\frac{W_r(s)}{T_L(s)}$ , que relaciona la velocidad con el torque. Entonces, la ganancia en estado estacionario de  $\frac{W_r(s)}{T_L(s)}$  tiene signo negativo y se puede calcular a partir de los parámetros obtenidos anteriormente. Debido a que la velocidad disminuye, la ganancia tendrá signo negativo y su valor es

$$K_{ss} = \frac{-(192 - 164)}{1,05 \times 10^{-3}} = 3,23 \times 10^4$$

con la cual es posible despejar

$$K_i = \frac{R_a}{K_m K_{ss}} = 14,4 \times 10^{-3}$$

Usando el algoritmo de Chen se logra identificar la función de transferencia  $\frac{W_r(s)}{V_a(s)}$

```

1 %% Determinamos las funciones de transferencia a partir del metodo desarrollado
2 % por Chen
3
4 % td: tiempo de delay de entrada
5 % t1: tiempo inicial para el algoritmo
6
7 td = 0.0351;
8 t1 = 0.0001;
9 %k=max(w_r);
10 k = 198
11 [val lugar] = min(abs((t1+td)-t))
12
13 k1 = w_r(lugar)/k-1;
14 [val lugar] = min(abs((2*t1+td)-t));
15 k2 = w_r(lugar)/k-1;
16 [val lugar] = min(abs((3*t1+td)-t));
17 k3 = w_r(lugar)/k-1;
18 %k1 = 135.58/k-1
19 %k2 = 191.60/k-1
20 %k3 = 198.30/k-1
21
22 b=4*k1^3*k3-3*k1^2*k2^2-4*k2^3+k3^2+6*k1*k2*k3;
23 alfa1 = (k1*k2+k3-sqrt(b))/(2*(k1^2+k2));
24 alfa2 = (k1*k2+k3+sqrt(b))/(2*(k1^2+k2));
25 beta= (2*k1^3+3*k1*k2+k3-sqrt(b))/sqrt(b);
26
27 T1 =-t1/log(alfa1);
28 T2 =-t1/log(alfa2);
29 %T3 = beta*(T1-T2)+T1
30 s = tf('s');
31 G1=(k)/(T1*s +1)/(T2*s +1); %G1 normalizada.
32 G1=G1/12;
33
34 figure
35 hold on
36 step(G1*12,5e-4, 'r') %Simulo aplicando 12V a la entrada.
37 plot(datos(:,1)-0.0351,datos(:,2));grid on; title('Velocidad Angular W_r');
38 legend('FdT estimado','FdT medido')
39 hold off

```

Así, la funcion de transferencia que relaciona la velocidad angular con la identificada es

$$\frac{W_r(s)}{V_a(s)} = \frac{16,5}{2,23 \times 10^{-9}s^2 + 8,43 \times 10^{-5}s + 1}$$

Para obtener los parámetros del motor restante, se puede comparar la función de transferencia identificada con la función obtenida mediante sus ecuaciones diferenciales, tal que

$$\frac{W_r(s)}{V_a(s)} = \frac{16,5}{2,23 \times 10^{-9}s^2 + 8,43 \times 10^{-5}s + 1} = \frac{K_i}{J_m L_{AA}s^2 + J_m R_a s + K_i K_m}$$

Como el valor de  $K_i$  es conocido, se puede multiplicar y dividir la función de transferencia identificada para lograr que ambas estén en la misma magnitud.

$$\frac{K_i/16,5}{K_i/16,5} \frac{16,5}{2,23 \times 10^{-9}s^2 + 8,43 \times 10^{-5}s + 1} = \frac{0,0144}{1,941 \times 10^{-12}s^2 + 7,34 \times 10^{-8}s + 8,707 \times 10^{-4}}$$

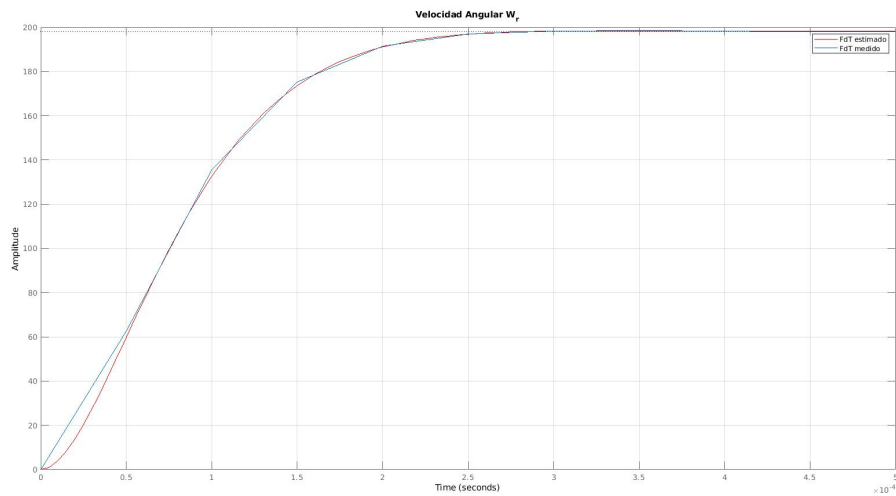


Figura 10:  $\frac{W_r(s)}{V_a(s)}$  medido vs identificado

Despejando las constantes,

```
1 [N D] = tfdata(G1, 'v')
2 %Normalizamos el numerador y el denominador
3 fac = Ki/N(3)
4 N = N*fac
5 D = D*fac
6
7 B = 0;
8 J = D(2)/Ra
9 Laa = D(1)/J
```

obtenemos:

$$J_m = 2,61 \times 10^{-9}$$

$$L_{AA} = 7,43 \times 10^{-4}$$

Se puede comprobar si las constantes son correctas si integramos las ecuaciones diferenciales que modelan al motor mediante Euler.

```
1 %Verificamos el modelo aproximado vs las mediciones
2 delta=1e-6;
3 ts=0.6;
4
5 %Generamos la funcion de entrada de tension Va y TL para el motor.
6 t_s = 0:delta:(ts-delta);
7 ia = zeros(1,ts/delta);
8 wr = zeros(1,ts/delta);
9 o = zeros(1,ts/delta);
10 va = zeros(1,ts/delta);
11 T_L = zeros(1,ts/delta);
12 ia(1)=0; wr(1)=0; o(1)=0; %Cond. iniciales
13
14 for i=1:(ts/delta)
15     if i*delta < 0.0351
```

```

16     va(i) = 0;
17     else
18         va(i) = 12;
19     end
20 end
21
22
23 for i=1:(ts/delta)
24     if i*delta < 0.1863
25         T_L(i) = 0;
26     elseif i*delta < 0.3372
27         T_L(i) = 1.03e-3;
28     elseif i*delta < 0.4866
29         T_L(i) = 0;
30     else
31         T_L(i) = 1.03e-3;
32     end
33 end
34
35 for i=2:(ts/delta-1)
36     ia(i)=ia(i-1)+delta*(-Ra*ia(i-1)/Laa-Km*wr(i-1)/Laa+va(i-1)/Laa);
37     wr(i)=wr(i-1)+delta*(Ki/J*ia(i-1)-B/J*wr(i-1)-T_L(i-1)/J);
38     o(i) = o(i-1) + delta*wr(i-1);
39 end
40
41 figure('Name','Torque vs Corriente')
42 subplot(3,1,1);
43 plot(t_s,wr);grid on; title('Velocidad angular, W_r');
44 hold on
45 plot(t,w_r);
46 legend('Aprox', 'Medido')
47 hold off
48
49 subplot(3,1,2);plot(t_s,ia);grid on; title('Corriente de armadura, I_a');
50 hold on
51 plot(t,i_a);
52 hold off
53
54
55 subplot(3,1,3);plot(t_s,T_L);grid on; title('Torque de carga, T_L');
56 hold on
57 plot(t,t_l);
58 hold off

```

Notar que existe un error en estado estacionario en la corriente de armadura  $i_a$ . Esto es provocado por la forma en que se calculo  $R_a$  (es solo una aproximación, no es exacto).

Si modificamos el valor de  $R_a$  de forma iterativa, para aproximar mejor el primer pico, obtenemos

$$R_a = 20\Omega$$

Debido a que  $R_a$  cambió, debemos volver a calcular el valor de las constantes. Realizando el procedimiento de calculo nuevamente obtenemos las constantes finales del motor:

$$R_a = 20\Omega$$



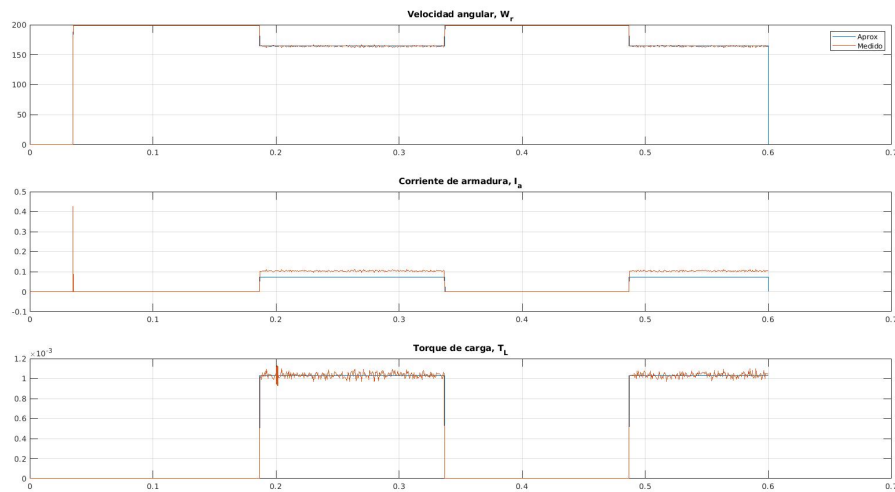


Figura 11: Modelo medido vs modelo aproximado

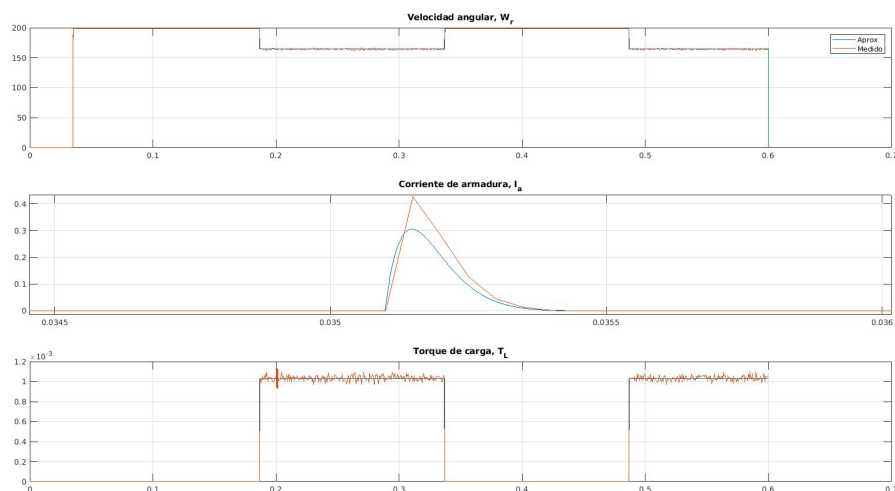


Figura 12: El pico en la corriente en el arranque no coincide de manera exacta debido al valor incorrecto de  $R_a$

$$K_m = 60,5 \times 10^{-3}$$

$$K_i = 10,2 \times 10^{-3}$$

$$B = 0$$

$$J_m = 2,61 \times 10^{-4}$$

$$L_{AA} = 5,28 \times 10^{-4}$$

Al simular nuevamente con los valores nuevos notamos que el error en estado estable de la corriente de armadura desapareció, según la figura 13.

**Ítem (6)** Implementar un PID en tiempo discreto para que el ángulo del motor permanezca en una referencia de 1radian sometido al torque descrito en la entrada.

A partir de las ecuaciones diferenciales que modelan el comportamiento del motor, obtenemos su modelo en espacio de estados con las siguientes matrices

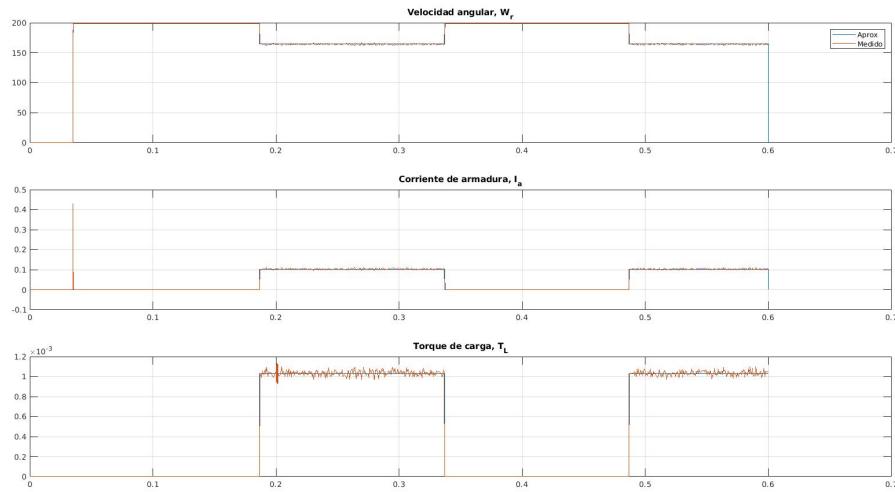


Figura 13: Modelo medido vs modelo aproximado con el valor de  $R_a$  correcto

$$A = \begin{bmatrix} -R_a/L_{AA} & -K_m/L_{AA} & 0 \\ K_i/J_m & -B_m/J_m & 0 \\ 0 & 1 & 0 \end{bmatrix}, B = \begin{bmatrix} 1/L_{AA} & 0 \\ 0 & -1/J_m \\ 0 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

A partir de las matrices de la ecuación de estado y salida, simulamos el sistema mediante Euler, implementando un PID discreto para controlar la posición y colocando una entrada de torque a partir de los 300ms.

Las matrices de entrada y salida son:

$$Y = \begin{bmatrix} i_a \\ \omega_r \\ \theta_r \end{bmatrix}, U = \begin{bmatrix} V_a \\ T_L \end{bmatrix}$$

Las constantes del PID se determinaron a prueba y error, resultando  $K_p = 80$ ,  $K_i = 500$  y  $K_d = 0,01$

```

1  Laa=5.28e-04;
2  J=2.61e-09;
3  Ra=20;
4  B=0;
5  Ki=0.0102;
6  Km=0.0605;
7
8  A = [-Ra/Laa -Km/Laa 0; Ki/J -B/J 0; 0 1 0];
9  B = [1/Laa 0; 0 -1/J; 0 0];
10 C = [1 0 0; 0 1 0; 0 0 1];
11 D = [0 0; 0 0; 0 0]
12 U = [12; 0]
13
14 X = [0; 0; 0];
15 Xp = [0; 0; 0];
16 ii = 0;
```

```

17 t_etapa = 2e-7;
18 tF = 600e-3;
19 ref = 1;
20
21 %Constantes del PID
22 %Kp=0.1;Ki=0.01;Kd=5;color_='r';
23 Kp=80;Ki=500;Kd=0.01;color_='b';
24 Ts=t_etapa;
25 A1=((2*Kp*Ts)+(Ki*(Ts^2))+(2*Kd))/(2*Ts);
26 B1=(-2*Kp*Ts+Ki*(Ts^2)-4*Kd)/(2*Ts);
27 C1=Kd/Ts;
28
29 N = tF/t_etapa
30 omega=zeros(1,N);
31 theta=zeros(1,N);
32 ia=zeros(1,N);
33 e=zeros(1,N);
34 acc=zeros(1,N);
35 ij = 0;
36
37 for ii= 1:N-1
38     ij=ij+t_etapa;
39     if (ij>=300e-3)
40         U(2)=1e-3;
41     end
42     k = ii+2;
43     Xp=A*X+B*U;
44     X=X+Xp*t_etapa;
45     Y=C*X+D*U;
46     %e(k) = ref-Y(3);
47     %U(1) = U(1)+A1*e(k)+B1*e(k-1)+C1*e(k-2);
48
49     acc(ii+1)=U(1);
50     ia(ii+1)=Y(1);
51     omega(ii+1)=Y(2);
52     theta(ii+1)=Y(3);
53
54 end
55
56 t=t_etapa:t_etapa:tF;
57 subplot(2,1,1);hold on;
58 plot(t,theta,color_);title('Salida: Posicion, \theta_t');
59 subplot(2,1,2);hold on;
60 plot(t,omega,color_);title('Salida: Velocidad, \omega_t');
61 xlabel('Tiempo [Seg.]');

```

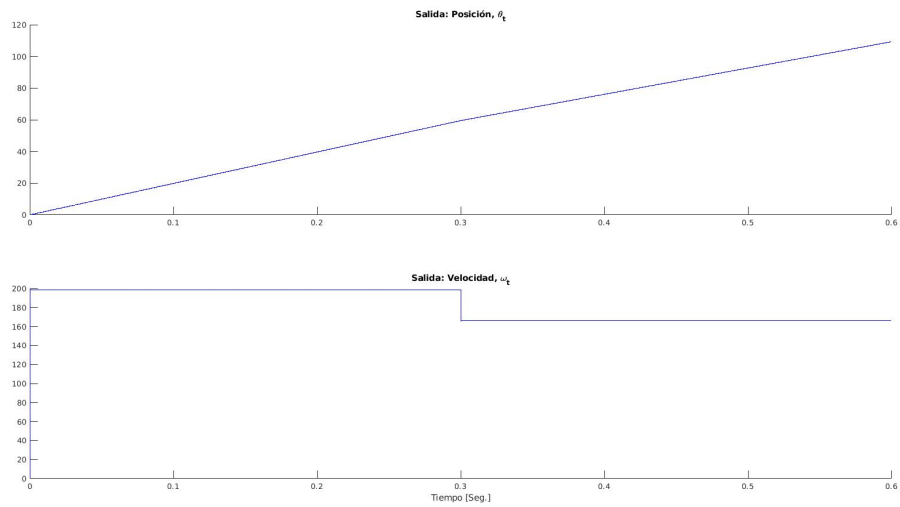


Figura 14: Sistema sin compensar

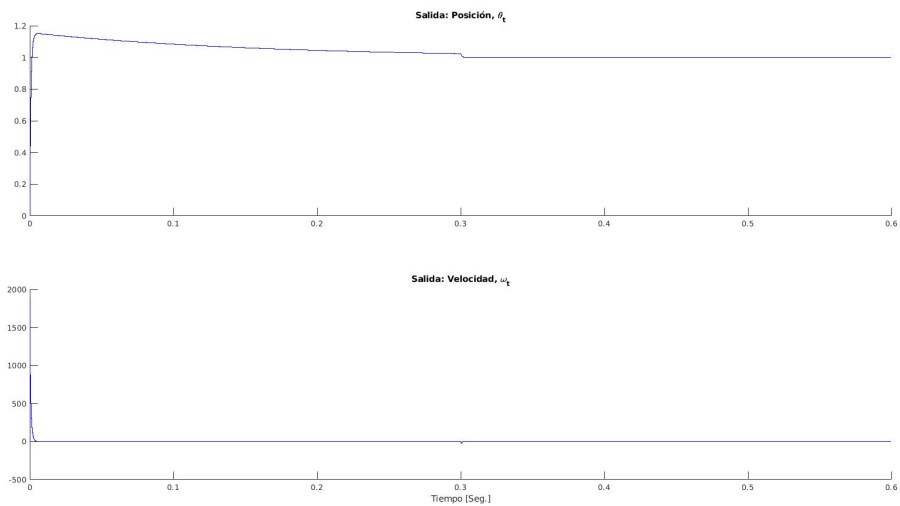


Figura 15: Sistema compensado con PID discreto

### 3. Conclusiones

Mediante la realización del presente trabajo, se llegaron a las siguientes conclusiones:

- Es posible y relativamente sencillo obtener las constantes que modelan de forma matemática a un sistema siempre y cuando sea posible medirlo.
- El uso de software de computo como MATLAB facilita la automatización, simulación y cálculo de procesos.
- El modelado en espacio de estados es una herramienta matemática sumamente útil en el análisis de sistemas con varias entradas y salidas, resolviendo limitaciones que posee trabajar solamente con funciones de transferencia.

- El diseño de controladores en dominio discreto es simple y a la vez aplicable en un microcontrolador.
- A la hora de implementar un controlador es posible y necesario observar los parámetros eléctricos con los cuales el sistema interactúa. Esto ayuda a conocer los valores máximos absolutos con los cuales podemos trabajar para evitar la ruptura de la planta que queremos controlar. Estos valores máximos no se tuvieron en cuenta para la implementación del PID discreto, lo cual podría ser un problema si se implementa sin limitar la acción de control.
- Se pueden relacionar de forma directa los parámetros y valores de una función de transferencia con una magnitud física.

## 4. Lecciones aprendidas

En el desarrollo del presente trabajo, se logró:

- Diseñar controladores con realimentación de estados para obtener una dinámica estable considerando la dinámica y la magnitud de las acciones de control.
- Seleccionar las variables de estado de un proceso lineal para generar una expresión matricial lineal.
- Inferir la evolución temporal de procesos reales representados en variables de estado.
- Calcular el modelo lineal de un proceso estable multivariable a partir de su respuesta al escalón.

A su vez, es recomendable usar, dentro de lo posible MATLAB por sobre Octave. Los programas que se usaron corrieron más rápido sobre MATLAB que sobre su alternativa open source, lo que ahorro mucho tiempo de trabajo.

También es importante destacar lo mucho que GitHub agiliza el proceso de compartir y corregir código mediante sus funciones colaborativas.

## 5. Recursos

El código utilizado para el desarrollo del presente trabajo se encuentra en el siguiente repositorio: [GitHub de Facundo Galvagno](#)