

Identificación de una planta de 3 Opams (sistema de tercer orden) utilizando la estructura paramétrica ARMAX.

Autores

Estrada Vidal, Jorge jor1550g@gmail.com
 Florian Chacon, Erick erick.florian.uni@gmail.com
 Giraldo Castillo, Oscar oscar.gi.cast@gmail.com

Asesores:

Ing. Rodriguez Bustinza, Ricardo robust@uni.edu.pe

Universidad Nacional de Ingeniería

I. OBJETIVOS

- Identificar el modelo de un planta de 3 Op-Amp's a través de la adquisición de datos y de la estructura paramétrica ARMAX

II. MARCO TEÓRICO

A. Estructuras de modelos parametricos

Las estructuras de modelos tambien conocidas como “cajas negras” quedan representadas por ejemplo, mediante una ecuación lineal en diferencias dado por:

$$y(t) + a_1 y(t-1) + \dots + a_{n_a} y(t-n_a) = b_1 u_{t-1} + \dots + b_{n_b} y(t-n_b) + e(t) \quad (1)$$

El termino de ruido blanco, $e(t)$ aca ingresa como un error directo en la ecuacion en diferencias dada en 1, a menudo es llamada ecuacion del modelo de error (estructura).

El vector que es llamado vector de parametros es el objetivo del estudio, es decir, encontrando dicho vector podemos conocer el modelo discreto y por ende el modelo continuo.

$$\theta = (a_1 \ a_2 \ \dots \ a_{n_a} \ b_1 \ \dots \ b_{n_b})^T \quad (2)$$

Dentro de estas estructuras parametricas principales que nos proporcionan las herramientas del software de simulacion de Matlab y LabVIEW está:

- Estructura Parametrica ARX:*
- Estructura Parametrica ARMAX:*
- Estructura Parametrica OE (Output Error):*
- Estructura Parametrica BJ (Box Jenkins):* En este laboratorio, estudiaremos la segunda estructura que describiremos a continuación:

B. Estructura ARMAX

El modelo ARMAX (AutoRegresive eXogenous Moving Average) es mas flexible describiendo la ecuacion de error como un “promedio movil”, el ruido blanco puede ser representado por $e(t)$, el modelo presenta la relacion de entrada y salida que puede ser descrita por una ecuacion en diferencia lineal en la forma.

$$y(t) = \frac{B(q)}{A(q)} U(t) + \frac{C(q)}{A(q)} e(t) \quad (3)$$

Siendo su función de transferencia:

$$Y(z) = \frac{B(z)}{A(z)} z^{-d} U(z) + \frac{C(z)}{A(z)} E(z) \quad (4)$$

A continuación presentamos un esquema básico del sistema analizado (ver Fig. ??).

En donde el bloque etiquetado como planta contiene lo siguiente: (ver Fig. ??)

Cuya simulación obtenida en Proteus es la siguiente: (ver Fig. ??)

III. PRESENTACIÓN DE RESULTADOS

En este capitulo mostraremos los resultados del análisis anterior. Llegando a obtener:

- Diseño analógico de un controlador PI.
- Rediseño digital de un controlador PI por Tustin.
- Comparación entre control PI y PID.

A. MATLAB

La data obtenida de la adquisición se puede apreciar en la figura 1

Se identifica el sistema usando la estructura paramétrica ARX (ver Fig. 2) usando una frecuencia de muestreo f_s

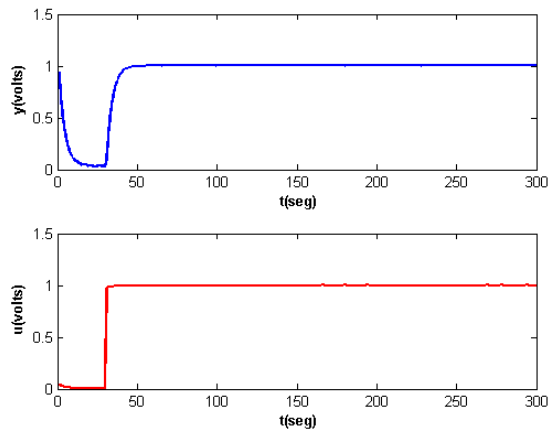


Fig. 1. Data obtenida de la adquisición

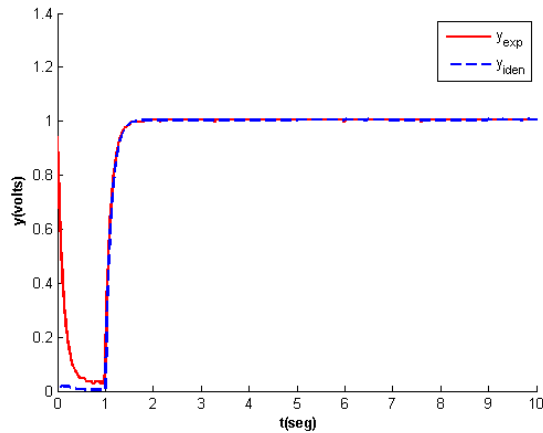


Fig. 2. Identificación ARX

de = 30Hz. Posteriormente, hallamos los controladores PI y exportamos los coeficientes obtenidos del controlador digital al directorio /data para su procesamiento en labview. El código generado es el siguiente:

```

1 clear all; close all; clc
2 %% Programa para calcular el controlador
3 %
4 %% Cargando la DATA
5 %
6 dataLeida = load('..\data\data_rc.lvm');
7 T=1/30; % Tiempo de Muestreo
8 y1=dataLeida(:,4);
9 u1=dataLeida(:,6);
10
11 figure
12 subplot(211)
13 plot(y1,'b','LineWidth',2);
14 xlabel('\bf t(seg)'); ylabel('\bf y(volts)');
15 subplot(212)
16 plot(u1,'r','LineWidth',2);
17 xlabel('\bf t(seg)'); ylabel('\bf u(volts)');
18
19 %
20 %% Identificación ARX
21 %
22 data=iddata(y1,u1,T);
23 th=arx(data,[1 1 1]);
24 present(th)

```

```

25 thc=d2c(th);
26 [num,den]=tfdata(thc);
27 Gp=tf(num,den)
28 gain = num{1}(2);
29 tau = 1/num{1}(2);
30
31 %
32 %% Polos de la planta
33 %
34 sp=pole(Gp);
35 ip=abs(imag(sp));
36 rp=abs(real(sp));
37
38 %
39 %% Especificaciones de diseño polos deseados
40 %
41 ts=1;
42 Mp=0.1;
43 zeta=-log(Mp)/sqrt((log(Mp))^2+pi^2);
44 wn=4.6/(zeta*ts);
45 s1=-zeta*wn+1j*wn*sqrt(1-zeta^2);
46 % Polo deseado
47 sd=s1;
48 id=abs(imag(sd));
49 rd=abs(real(sd));
50
51 %
52 %% Diseño del control PI continuo
53 %
54 theta1=pi-atan(id/rd);
55 theta2=atan((id-ip)/(rp-rd));
56 theta3=theta1+theta2+pi; % condición de fase
57 theta3 = pi_to_pi(theta3);
58
59 if abs(theta3)<pi/2
60     zc = rd+id/tan(theta3);
61 else
62     zc = rd-id/tan(theta3);
63 end
64 a = zc; % zero del controlador
65 % polo del controlador
66 sc = 0;
67
68 r1 = sd-sc; %polo
69 r2 = sd-sp; %polo
70 r3 = sd-zc; %zero
71
72 K = abs(r1)*abs(r2)/(gain*abs(r3));
73
74 %
75 %% Simulación del Controlador PI continuo
76 %
77 Gc=tf(K*[1 a],[1 0])
78 % Funcion de transferencia en lazo cerrado H
79 L=series(Gc,Gp);
80 H=L/(L+1)
81
82 figure; hold on;
83 t = 0:0.001:5;
84
85 u=ones(size(t));
86 yp=lsim(H,u,t);
87
88 plot(t,u,'r')
89 plot(t,yp,'b','LineWidth',2)
90
91 xlabel('\bf t(seg)'); ylabel('\bf y(volts)');
92 legend('set point','y_{lazo cerrado}');
93
94 %
95 %% Re-diseño por tustin del Control
96 %% en Tiempo Discreto
97 %
98 T=tau/5;
99 [Nt,Dt] = tfdata(Gc,'v');
100 Nt = poly2sym(Nt,'s');
101 Dt = poly2sym(Dt,'s');

```

```

syms z
103 Gdt = Nt/Dt;
Gdt = subs(Gdt,{'s'},(2*(z-1))/(T*(z+1)));
105 Gdt = simplify(Gdt);
Gdt = vpa(Gdt,4);
107 [NDt, DDt] = numden(Gdt);
NDt = sym2poly(NDt);
109 DDt = sym2poly(DDt);

111 % -----
112 %% FT del Controlador digital D(z)
113 % -----
GDt = tf(NDt,DDt,T);

115 % -----
116 %% Coeficientes para lectura de LabVIEW
117 % -----
119 [Np,Dp]=tfdata(Gp,'v');
planta = [Np Dp];
121 save '../data/coef_planta.lvm' planta -ascii -tabs
122 save '../data/num_controller.lvm' NDt -ascii -tabs
123 save '../data/den_controller.lvm' DDt -ascii -tabs

```

Finalmente podemos validar que las respuestas de los sistemas controlados por un controlador PI (ver Fig. 3) y PID (ver Fig. 4) en lazo cerrado cumplen con las condiciones de diseño propuestas.

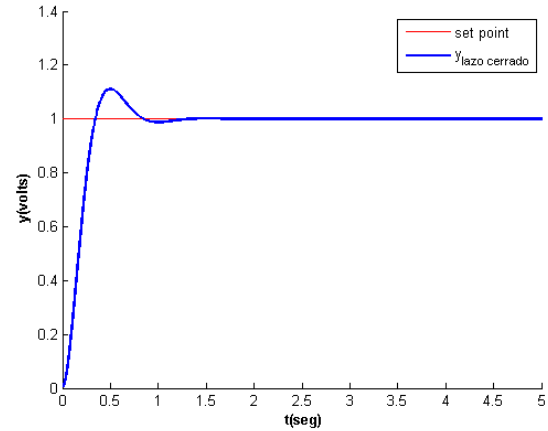


Fig. 4. Planta controlada usando un controlador PID

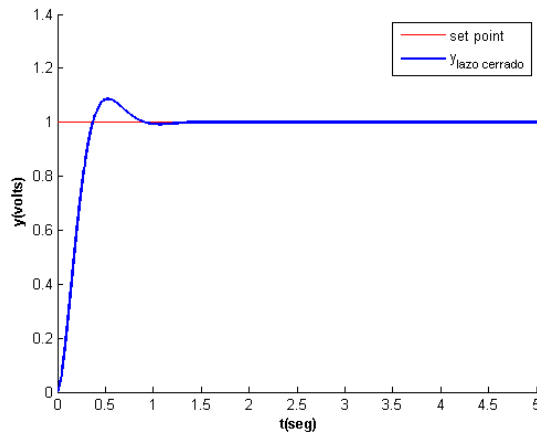


Fig. 3. Planta controlada usando un controlador PI

IV. CONCLUSIONES

- Notamos que comparando ambos controladores empleados (PI y PID), el sistema responde mejor frente al controlador PI ya que posee un menor sobre impulso ante el PID. (ver Fig. 5)

REFERENCES

- [1] Repositorios <https://github.com/oskargicast/ControladorPI>
- [2] Ing. Rodríguez Bustinza, Ricardo *Diseño del controlador discreto usando aproximador digital*.
- [3] Leonardo J. Marín, Víctor M. Alfaro *Sintonización de controladores por ubicación de polos y ceros* Departamento de Automática, Escuela de Ingeniería Eléctrica, Universidad de Costa Rica

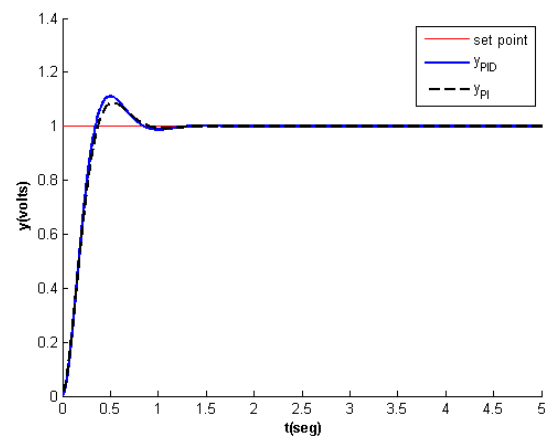


Fig. 5. Comparación entre PI y PID