

Diseño analítico de un controlador PID analógico y digital de velocidad para un motor DC

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 motor DC.
- Diseñar un controlador PID para dicha planta.
- Analizar y validar los resultados en tiempo real.

II. PRESENTACIÓN DE RESULTADOS

A continuación se expondrán los resultados obtenidos de acuerdo al desarrollo progresivo de la experiencia. Para un mayor entendimiento del proyecto puede revisar la figura 2 para un mejor entendimiento del sistema.

- Análisis del sistema en lazo abierto.
 - Escalamiento de magnitudes.
 - Toma de datos para la identificación.
- Identificación del sistema.
- Diseño del controlador PID.
- Validación y comparación con un modelo teórico en tiempo real.

A. Análisis del sistema en lazo abierto

1) *Escalamiento de magnitudes*: La tabla I nos muestra datos obtenidos del sistema en lazo abierto, para ello se realizo el esquema labview de la figura ??.

De la data obtenida en la tabla I extraemos las siguientes gráficas y procedemos a linealizarlas.

- $V_o(volts)$ vs. $E_{je_1}(rpm)$ (Ver Fig. 3). La recta aproximada mediante minimos cuadrados queda descrita por la ecuación 1

$$RPM = 906.48V_o - 0.41 \quad (1)$$

$$R^2 = 0.9999$$

- $E_{je_1}(rpm)$ vs. $V_o(volts)$ (Ver Fig. 4). La recta aproximada mediante minimos cuadrados queda descrita por la ecuación 2

$$V_o = 0.0011RPM + 0.0006 \quad (2)$$

$$R^2 = 0.9999$$

- $Enc(Hz)$ vs. $V_o(volts)$ (Ver Fig. 5)
- $V_i(volts)$ vs. $E_{je_1}(rpm)$ (Ver Fig. 6)
- $V_i(volts)$ vs. $V_o(volts)$ (Ver Fig. 7)

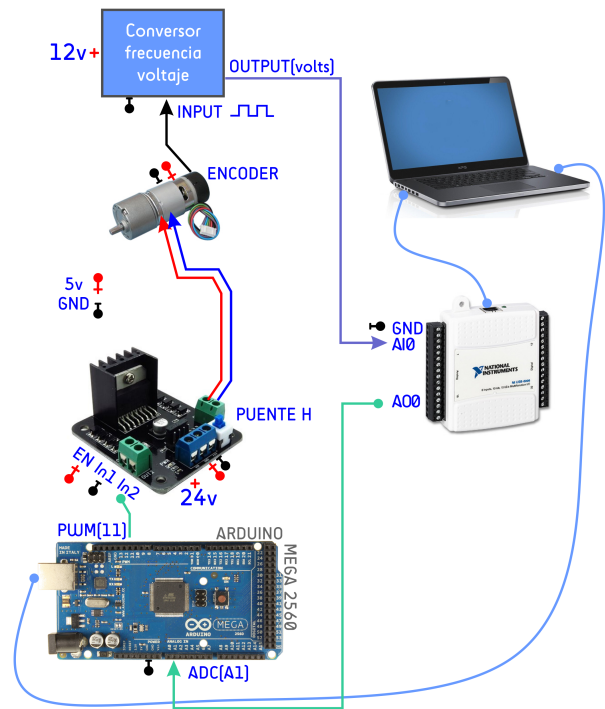


Fig. 1. Esquemático del sistema en lazo cerrado

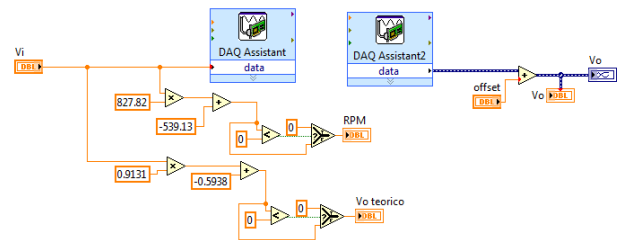
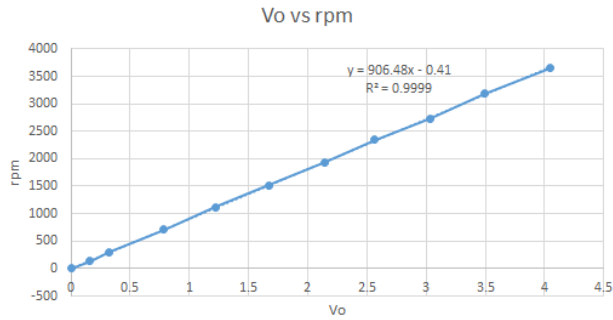
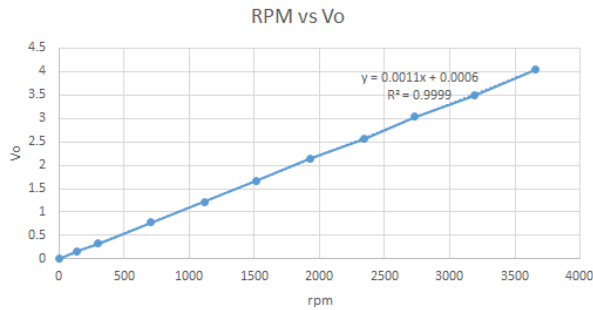


Fig. 2. Captura de datos para el escalamiento

| V_i | V_o | Enc | Eje | Eje_1 | Eje_2 |
|-------|-------|-------|-------|---------|---------|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0.8 | 0.163 | 230 | 2.3 | 138 | 127 |
| 1 | 0.326 | 500 | 5 | 300 | 294 |
| 1.5 | 0.784 | 1180 | 11.8 | 708 | 683 |
| 2 | 1.222 | 1860 | 18.6 | 1116 | 1098 |
| 2.5 | 1.669 | 2530 | 25.3 | 1518 | 1504 |
| 3 | 2.148 | 3220 | 32.2 | 1932 | 1919 |
| 3.5 | 2.565 | 3900 | 39 | 2340 | 2405 |
| 4 | 3.034 | 4550 | 45.5 | 2730 | 2746 |
| 4.5 | 3.492 | 5310 | 53.1 | 3186 | 3189 |
| 5 | 4.042 | 6090 | 60.9 | 3654 | 3698 |

TABLE I

V_i (volts): VOLTAJE DE ENTRADA AL ARDUINO, V_o (volts): VOLTAJE DE SALIDA DEL CONVERSOR F/V, Enc (Hz): FRECUENCIA DEL ENCODER, Eje (Hz): FRECUENCIA DEL EJE EN HZ OBTENIDO INDIRECTAMENTE, Eje_1 (rpm): FRECUENCIA DEL EJE EN RPMs OBTENIDO INDIRECTAMENTE, Eje_2 (rpm): FRECUENCIA EN RPMs OBTENIDO DIRECTAMENTE POR UN TACÓMETRO

Fig. 3. V_o (volts) vs. Eje_1 (rpm)Fig. 4. Eje_1 (rpm) vs. V_o (volts)

2) *Toma de datos para la identificación:* Mediante un generador de señal pre-contruido en labview (Ver Fig. 8), introduciremos a nuestro sistema diversas señales de modo que capturaremos entradas y salidas para posteriormente identificarlas.

La data obtenida es la siguiente:

- Para una señal de entrada Gate:
 - Entrada - Salida (Ver Fig. 9)
 - Comparación de estructuras ARX (Ver Fig. 10)

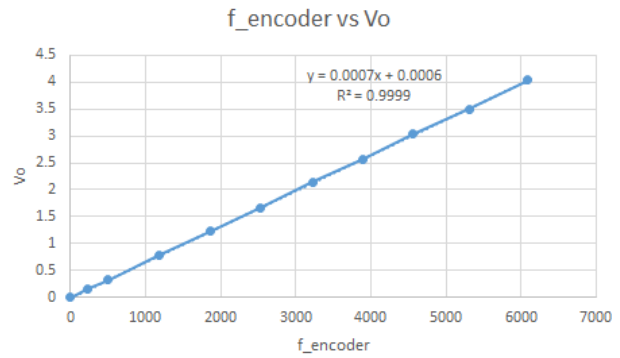
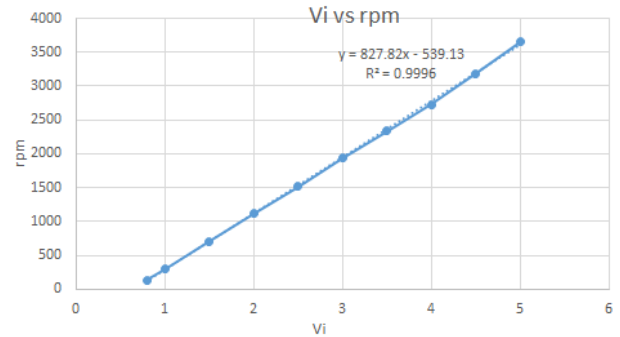
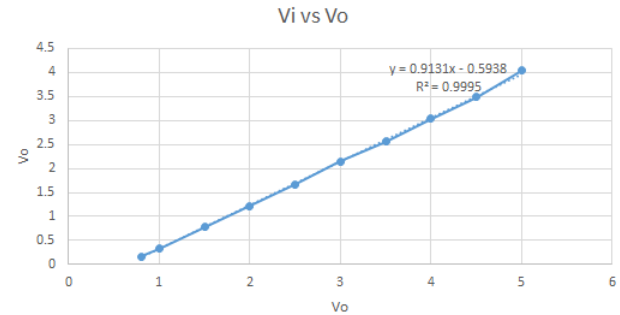
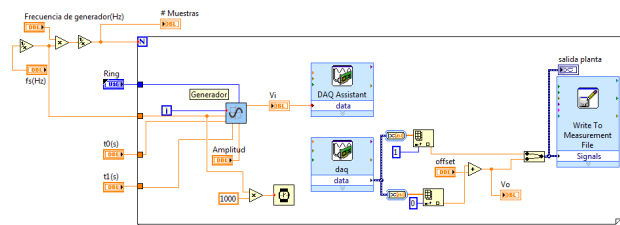
Fig. 5. Enc (Hz) vs. V_o (volts)Fig. 6. V_i (volts) vs. Eje_1 (rpm)Fig. 7. V_i (volts) vs. V_o (volts)

Fig. 8. Toma de datos en labview

- Para una señal de entrada Step:
 - Entrada - Salida (Ver Fig. 11)
 - Comparación de estructuras ARX (Ver Fig. 12)
- Para una señal de entrada Rampa:
 - Entrada - Salida (Ver Fig. 13)

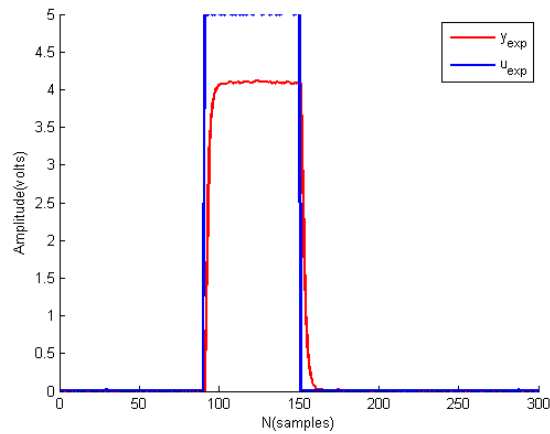


Fig. 9. Respuesta del sistema ante una entrada tipo gate

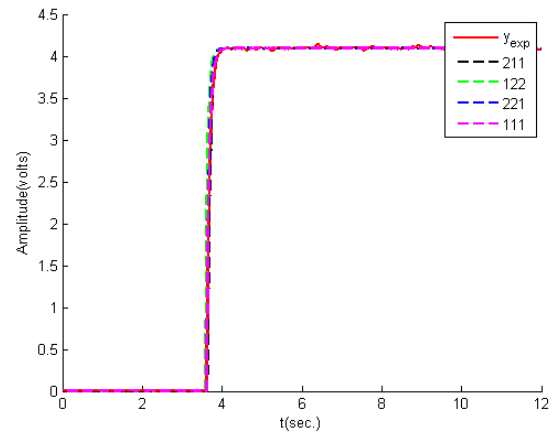


Fig. 12. Comparación de estructuras paramétricas ARX

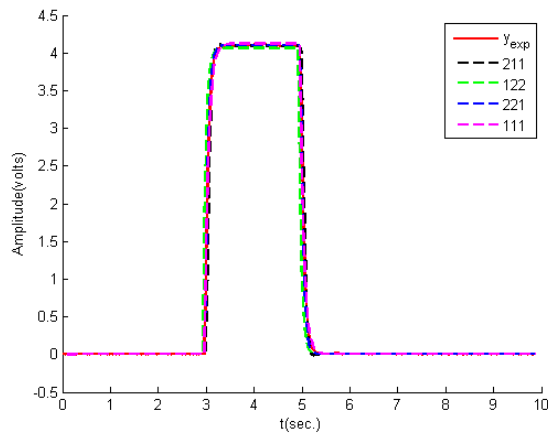


Fig. 10. Comparación de estructuras paramétricas ARX

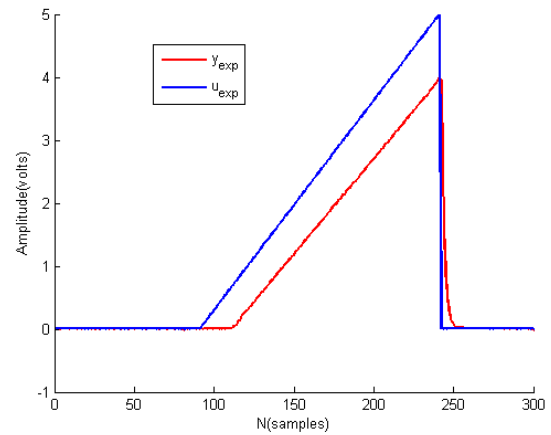


Fig. 13. Respuesta del sistema ante una entrada tipo rampa

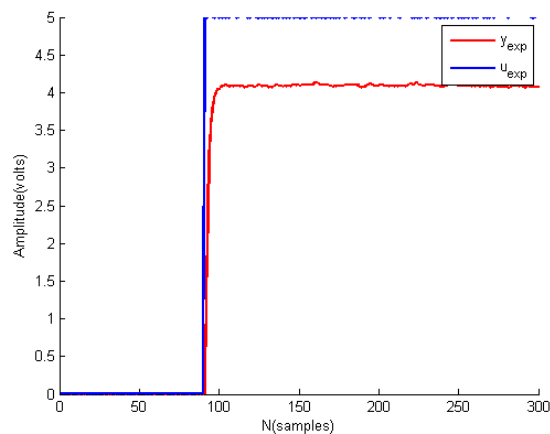


Fig. 11. Respuesta del sistema ante una entrada tipo step

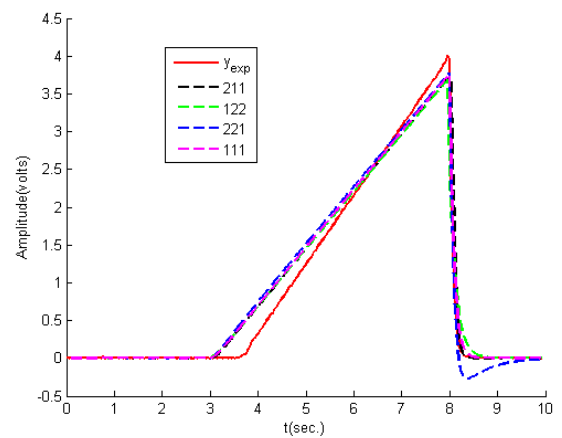


Fig. 14. Comparación de estructuras paramétricas ARX

- Comparación de estructuras ARX (Ver Fig. 14)
- Para una señal de entrada Seno:
 - Entrada - Salida (Ver Fig. 15)
 - Comparación de estructuras ARX (Ver Fig. 16)

B. Identificación del sistema.

Debido a que solo controlaremos la velocidad de nuestro motor y considerando la inductancia de este muy cercana a cero ($L \approx 0$), el sistema puede ser aproximado a uno de primer orden. Por lo tanto, procederemos a identificar nuestro sistema


```

77 % armax
78 plot(t, yc, strcat(colors(i), '—'), 'LineWidth',
      line_widht);
79 % legend(colors(i))
80 end
81
82 ylabel('Amplitude(volts)'), xlabel('t(sec.)');
83 legend(...
84 'y_{exp}', ...
85 arrayfun(@num2str, nn(1,:), 'UniformOutput', true),
86 ...
87 arrayfun(@num2str, nn(2,:), 'UniformOutput', true),
88 ...
89 arrayfun(@num2str, nn(3,:), 'UniformOutput', true),
90 ...
91 arrayfun(@num2str, nn(4,:), 'UniformOutput', true));

```

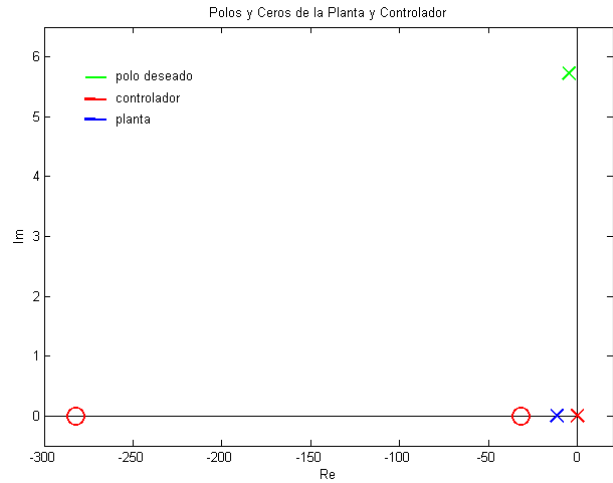


Fig. 17. Lugar de las raíces

C. Diseño de un controlador PID

Observamos que nuestra planta presenta un polo real

$$s_p = -12.85 \quad (6)$$

Sabemos que la Función de Transferencia de nuestro controlador PID es de la siguiente forma

$$G_c(s) = \frac{K(s+a)(s+b)}{s} \quad (7)$$

Assumiendo las siguientes variables de diseño:

$$t_s = 1; \quad (8)$$

$$M_p = 0.08; \quad (9)$$

Con esto procederemos a hallar nuestros polos deseados (s_d) de la siguiente manera:

$$\zeta = -\frac{\log(M_p)}{\sqrt{\log(M_p^2 + \pi^2)}} = 0.6266 \quad (10)$$

$$w_n = \frac{4.6}{\zeta t_s} = 7.3415 \quad (11)$$

$$\begin{aligned} s_d &= -\zeta w_n \pm j w_n \sqrt{1 - \zeta^2} \\ &= -4.6 \pm j 5.7216 \end{aligned} \quad (12)$$

Graficando el lugar de las raíces (Ver Fig. 17), en ella se observa el polo deseado, así como el polo de la planta y el cero del controlador PI cuyos parámetros están hallados aplicando el criterio de fase para el caso del cero y el criterio de la magnitud para la ganancia K.

Para el cálculo de la ganancia K, hacemos uso de la ganancia unitaria en el lugar de las raíces:

$$K = \frac{|r1| |r2|}{Gain |r3| |r4|} = 0.00089201 \quad (13)$$

Obteniendo finalmente las ganancias de nuestro controlador $K_p = 0.2804$ $K_i = 8.0547$ $K_d = 0.0009$, nuestro sistema responde de la siguiente forma (Ver Fig. 18)

El código ?? escrito en matlab realiza el proceso de calculo de las ganancias de nuestro controlador PID por el método de ubicación de polo.

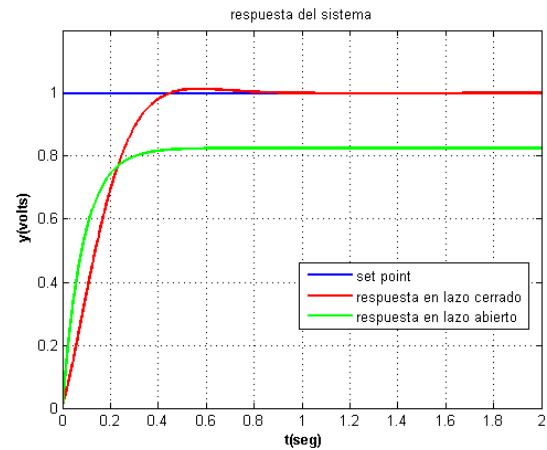


Fig. 18. Respuesta del sistema

```

1 \label{code:pid}
2 clear all; close all; clc
3 %
4 %
5 % Cargando la DATA
6 %
7 data_load = load('..data/motor_gate.lvm');
8 T = 1/30; % Tiempo de Muestreo
9 yl=data_load(:,4);
10 ul=data_load(:,2);
11 %
12 % Identificación ARX
13 %

```

```

14 data=iddata(yl,ul,T);
15 th=arx(data,[1 1 1]);
16 present(th)
17 thc=d2c(th);
18 [num,den]=tfdata(thc);
19 Gp=tf(num,den);

```

```

20 [Pp,Zp]=pzmap(Gp);
21 %
22 % Especificaciones de Diseño
23 %
24 ts=1; % Tiempo de Establecimiento
25 % ts=0.5;
26 Mp=0.08; % Máximo Sobrepasso
27 %
28 % Polos deseados
29 %
30 zeta=-log(Mp)/sqrt((log(Mp))^2+pi^2);
31 wn=4.6/(zeta*ts);
32 sl=-zeta*wn+1j*wn*sqrt(1-zeta^2);
33 sd=sl;
34 pdi=imag(sd);
35 pdr=real(sd);
36 %
37 %
38 % Ploteo de polos y ceros Planta + Controlador
39 %
40 plot(real(Pp),imag(Pp),'xb','MarkerSize',14,'
    LineWidth',2); % polos planta
41 hold on;
42 plot(real(Zp),imag(Zp),'ob','MarkerSize',14,'
    LineWidth',2); % ceros planta
43 plot(pdr,pdi,'xg','MarkerSize',14,'LineWidth',2); %
    polos deseados
44
45 plot([-300 1],[0 0],'k') % ejex
46 plot([0 0],[-2 8],'k') % ejey
47 %
48 % Angulos hacia los polos deseados
49 %
50 theta1=(pi-atan(abs(pdi/pdr)))*180/pi; % polo
    del PID (s=0)
51 theta2=atan(abs(pdi)/abs(Pp-pdr))*180/pi; % polo de
    la planta
52 %
53 % Fase positiva que deben aportar los ceros del PID
54 theta_c=-180+(theta1+theta2);
55 %
56 % Calculo de a y b (Condicion de Fase)
57 theta_a=0.1*theta_c;
58 a=pdr-pdi/tan((180-theta_a)*pi/180);
59 theta_b=theta_c;
60 b=pdr-pdi/tan((180-theta_b)*pi/180);
61 plot(a,0,'or','MarkerSize',14,'LineWidth',2) % ceros
    PID (s=-a)
62 plot(b,0,'or','MarkerSize',14,'LineWidth',2) % ceros
    PID (s=-b)
63 plot(0,0,'xr','MarkerSize',14,'LineWidth',2); %
    polos PID
64
65 title('Polos y Ceros de la Planta y Controlador');
66 axis([-300 20 -0.5 6.5]);
67 ylabel('Im'), xlabel('Re');
68
69
70 %
71 % Calculo de la Ganancia K (Condicion de Magnitud)
72 %
73 Gc=tf(conv([1 abs(a)],[1 abs(b)]],[1 0]);
74 FLA=series(Gc,Gp);
75 K=rlocfind(FLA,sd);
76 % Ganancias Kp, Ki, Kd del controlador continuo
77 Kp=K*abs(a+b);
78 Ki=K*abs(a*b);
79 Kd=K;
80 KPID=[Kp Ki Kd];
81
82 %
83 % PID No Interactivo GPID(s)=Kp*(1+1/(Ti*s))+Td*s)
84 %
85 Ti=Kp/Ki;
86 Td=Kd/Kp;
87 Gc=K*Gc;
88 L=series(Gc,Gp);
89 H=L/(L+1);
90 % Ploteos
91 figure
92 t=0:0.001:2;
93 u=ones(size(t));
94 yp = lsim(H,u,t);
95 yla = lsim(Gp,u,t);
96 plot(t,u,'b','LineWidth',2)
97 hold
98 plot(t,yp,'r','LineWidth',2)
99 plot(t,yla,'g','LineWidth',2)
100 axis([0 2 0 1.2])
101 xlabel('\bf t(seg)')
102 ylabel('\bf y(volts)');
103 title('Respuesta del Sistema');
104 legend('set point','respuesta en lazo cerrado','
    respuesta en lazo abierto')
105 grid
106
107 %
108 % Re-diseño por tustin del Control en Tiempo
    Discreto
109 %
110 [num,den]=tfdata(Gp);
111 tau=(1/den{1}(2));
112 T=tau/5;
113 [Nt,Dt] = tfdata(Gc,'v');
114 Nt = poly2sym(Nt,'s');
115 Dt = poly2sym(Dt,'s');
116 syms z
117 Gdt = Nt/Dt;
118 Gdt = subs(Gdt,{'s'},(2*(z-1))/(T*(z+1)));
119 Gdt = simplify(Gdt);
120 Gdt = vpa(Gdt,4);
121 [NDt, DDt] = numden(Gdt);
122 NDt = sym2poly(NDt);
123 DDt = sym2poly(DDt);
124 %
125 % FT del Controlador digital D(z)
126 %
127 GDt = tf(NDt,DDt,T);

```

```

128 printsys (NDt,DDt, 'z')
129 [Np,Dp]=tfdata (Gp, 'v');
130 datos=[Np Dp NDt DDt];
131
132 disp ('Kp Ki Kd: ');
133 disp (KPID);

```

D. Validación y comparación con un modelo teórico en tiempo real

Finalmente podemos validar mediante el esquema mostrado en la figura 19 que las respuesta del sistemas controlados por un controlador PID (ver Fig. 20) en lazo cerrado cumplen con las condiciones de diseño propuestas.

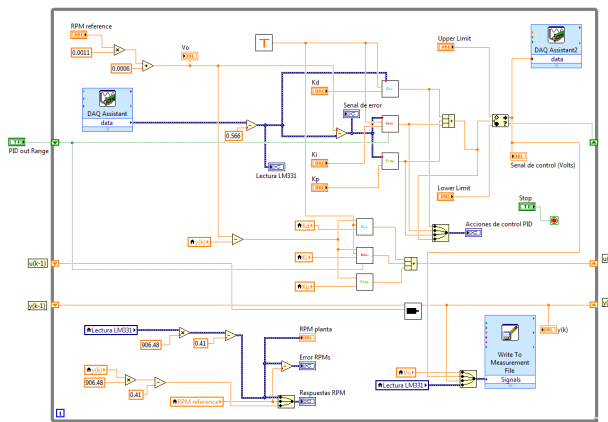


Fig. 19. Esquema de un Controlador PID digital en labview

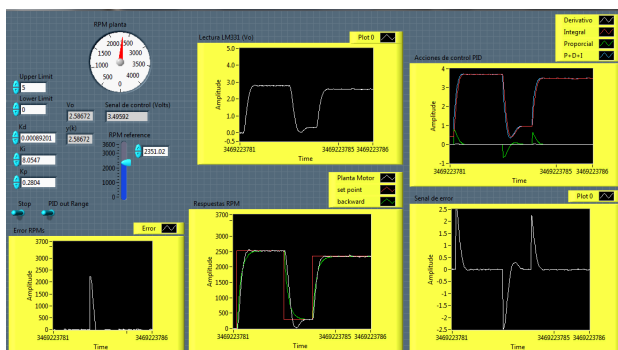


Fig. 20. Panel labview mostrando respuesta simulada(backward) y experimental(planta real)

Podemos del mismo modo visualizar la respuesta del controlador PID con la misma data (ver Fig. 21).

III. CONCLUSIONES

- Es importante quitar el offset (hacer que nuestra señal comience en cero) mediante software, para poder realizar la identificación.
- Basta con modelar a nuestro sistema como uno de primer orden, disminuyendo la complejidad de diseño de las ganancias de nuestro controlador.

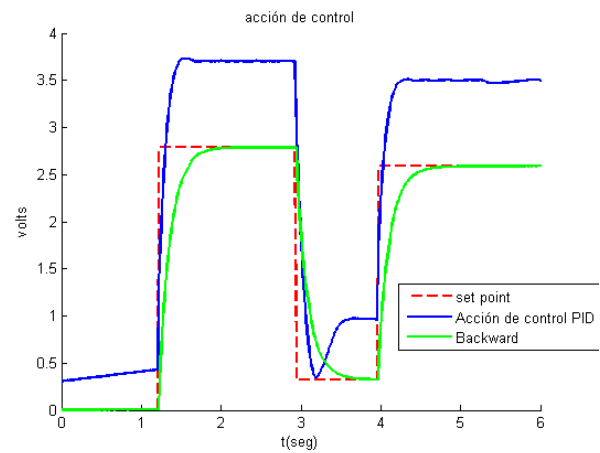


Fig. 21. Acción del controlador

- Es necesario configurar la frecuencia de PWM a una frecuencia mayor de 15kHz (nuestro PWM esta a 20KHz). De este modo se evita un chillido de parte del motor.
- Nuestro sistema responde de forma muy eficiente y rápida, a pesar de emplear un motor con desgaste mecánico.
- Todo parte del escalamiento, si este esta mal realizado, nuestro sistema en lazo cerrado responderá de acuerdo a las ecuaciones derivadas de este.

REFERENCES

- [1] Repositorios <https://github.com/oskargicast/Control-Velocidad-Motor-DC>
- [2] Ing. Rodriguez 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