

## Ejemplo de Diseño de Sistema de Control con Retardo

En este ejemplo se considerará un diseño que se efectuará utilizando métodos clásicos de control lineal en Laplace. En este caso el lugar geométrico de Evans o LGR.

El sistema de control es el siguiente:

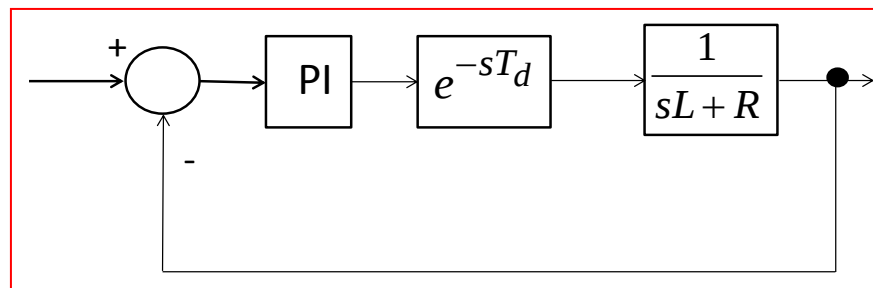


Fig. 1. Sistema de control a implementar.

Considerando los parámetros de la tarea se utiliza  $R=0.4\Omega$  y  $L=120\text{mH}$ . El retardo de transporte (promedio) es  $T_d=1.66\text{ms}$ . Utilizando Padé de primer orden el retardo de transporte queda representado por la función de transferencia:

$$e^{-sT_d} \approx \frac{1-sT_d/2}{1+sT_d/2} \approx \frac{-s+1200}{s+1200} \quad (1)$$

Y la función de transferencia de la planta es:

$$G(s) = \frac{1}{0.12s+0.4} \quad (2)$$

Las instrucciones utilizadas para obtener la planta de lazo abierto en Matlab son:

```
>> [Num,Den]=pade(1.6666e-3,1);
>> sys=tf(NuM,Den);
>> sys2=tf([1],[L R]);
>> sysT=sys*sys2;
>> sysT
```

Transfer function sysT:

$$-s + 1200$$

$$0.12 s^2 + 144.4 s + 480$$

El lugar de la raíz correspondiente a la planta sysT mas un controlador PI, se muestra en la Fig. 2. En el diseño se utilizó un coeficiente de amortiguamiento de 0.707 y una frecuencia natural de aproximadamente 70Hz. El lugar de la raíz en la zona de los polos

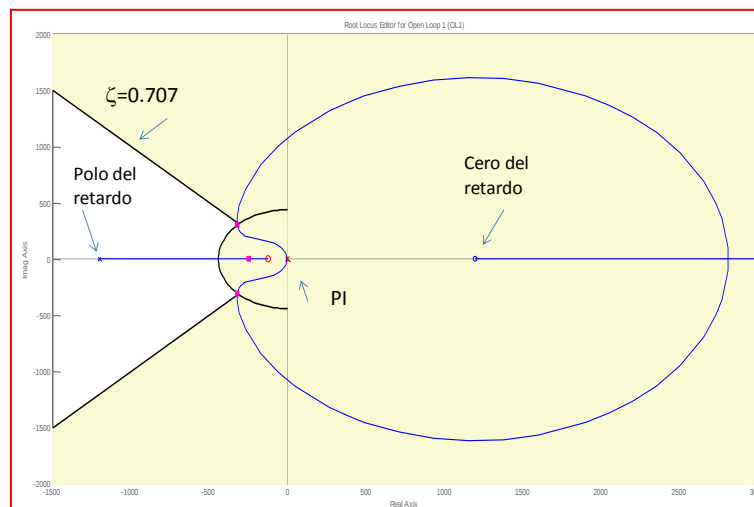


Fig. 2. Lugar de la raíz considerando planta, retardo y controlador.

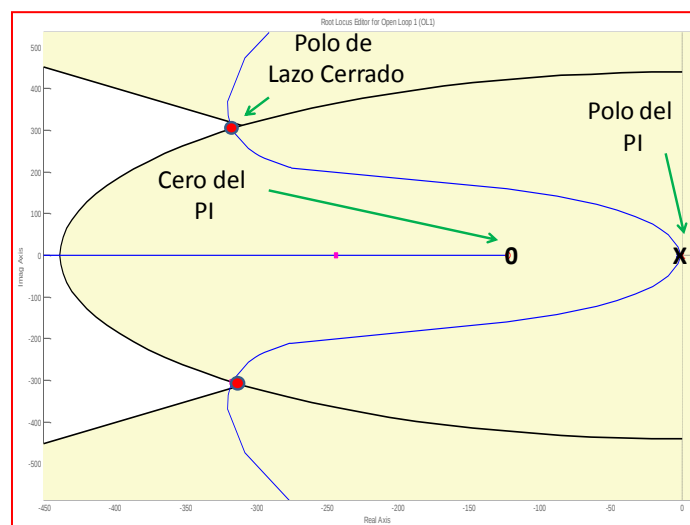


Fig. 3. Vista ampliada de la zona de polos de lazo cerrado dominante.

dominantes, se muestra en la Fig. 3. Utilizando rltol un controlador propuesto es:

$$G_c(s) = 38.6 \frac{(s+123)}{s} \quad (3)$$

Para discretizar utilizando la transformada bilineal se utiliza  $\omega_s = 25\omega_n = 25(2\pi 70) = 10996$  (recetas válidas para la mayor parte de los sistemas). El tiempo de muestreo  $T_s$  se obtiene como  $T_s = \frac{2\pi}{\omega_s} \approx 570\mu s$ . La transformada bilineal se obtiene utilizando el comando c2dm (continuous to discrete using method) de Matlab. Las instrucciones necesarias para obtener el controlador digital a partir de (3) se muestran en la Fig. 4.

En el plano z el controlador es:

```
>> num=38.6*[1 123];      Numerado PI analógico
>> den=[1 0];              Denominador PI analógico

>> [numd,dend]=c2dm(num,den,570e-6,'tustin');  Conversión digital

numd =    [39.953    -37.247]  Numerador discreto

dend =    [1    -1]            Denominador discreto
```

Fig. 4. Instrucciones requeridas en Matlab para obtener el controlador en el plano z.

$$G_c(z) = 39.95 \frac{(z-0.93226)}{z-1} \quad (4)$$

Como ya se ha explicado anteriormente, un controlador PI discreto se debe separar en su parte proporcional e integral utilizando algunas operaciones algebraicas simples:

$$K_z \frac{(z-a_z)}{(z-1)} = K_{pz} + K_{iz} \frac{z}{z-1}$$

PI Digital
Parte proporcional
Integrador Digital

Fig. 5. Separación del PI en parte proporcional e integral.

Finalmente el controlador PI, considerando antiwinding-up se implementa utilizando instrucciones de Matlab, tal como se muestra en la Fig. 6.

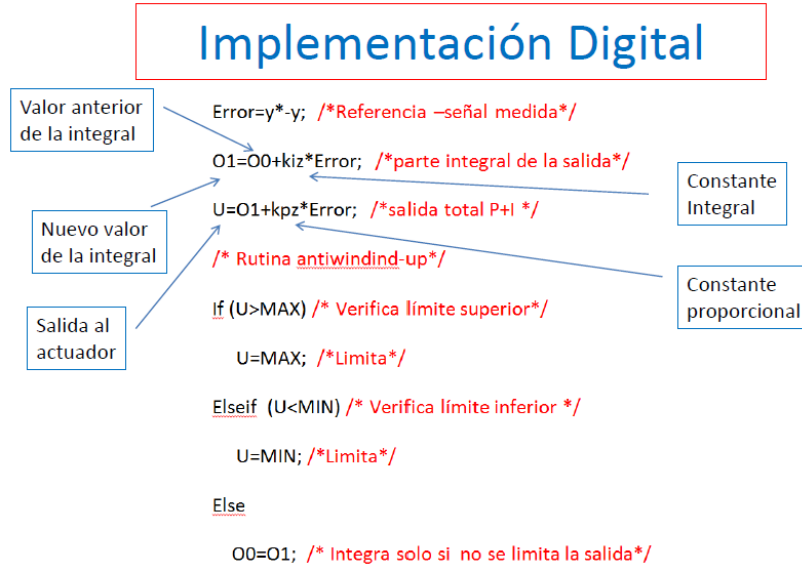


Fig. 6. Código Matlab requerido para implementar un controlador PI discreto.

El sistema de control digital se implementa en Simulink, tal como se muestra en la Fig. 7.

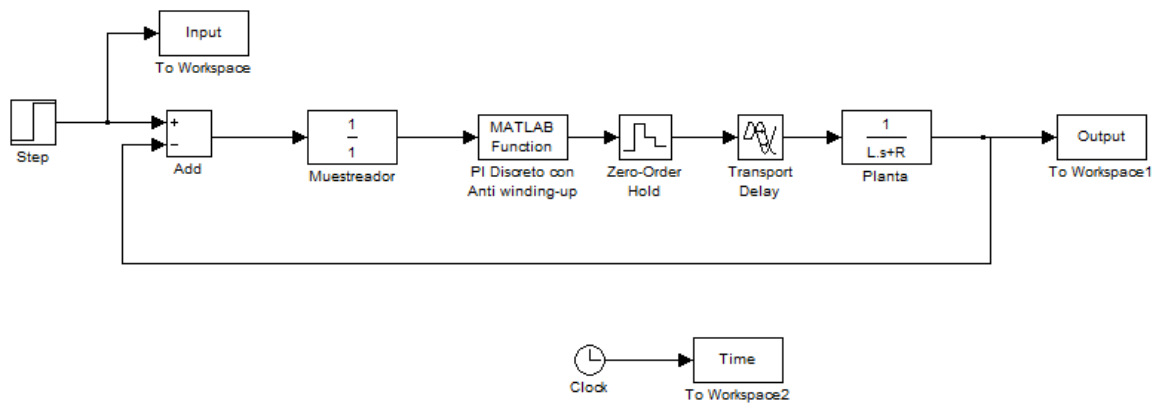


Fig. 7. Sistema de control digital.

En la Fig. 7 no se utiliza la aproximación de Padé para implementar el retardo. Esto se debe a que existen algunas simplificaciones que son adecuadas para diseñar controladores, pero que no deben ser utilizadas para fines de simulación. Matlab tiene un bloque diseñado para simular retardos que es bastante más exacto que la aproximación de Padé de primer orden.

La respuesta del sistema de control se muestra en la siguiente figura, para una entrada tipo escalón de corriente de 50A. La señal  $i^*$  es la referencia de corriente y  $V$  es la salida del controlador. Nótese que desde  $t=0.005$ s a  $t=0.02$ s el componente anti-winding up del controlador se encuentra activo.

Al final de este ejemplo se encuentran las dos rutinas en formato \*.m utilizadas en este trabajo. La rutina Initial\_tarea.m debe ejecutarse primero para obtener los valores iniciales de todas las variables.

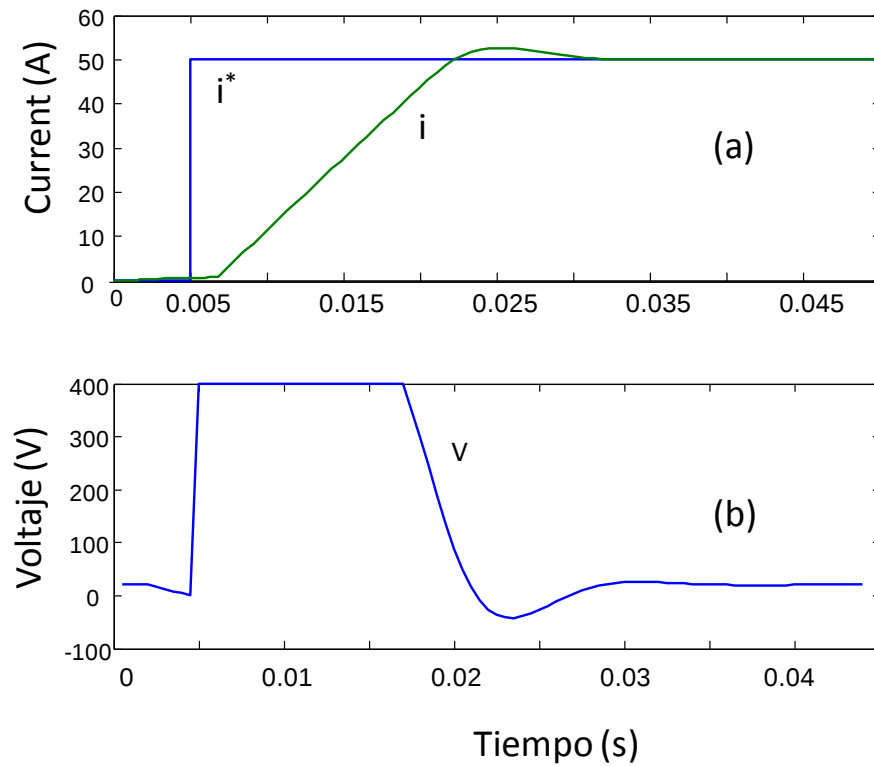


Fig. 8. Respuesta tipo escalón de  $i^* = 50$ A.

## Rutina Initial\_Tarea.m

```
%  
% Valores iniciales del sistema de control  
%  
global Oo Ol;    % Salida actual y salida anterior del componente I  
global Kpz Kiz;  % Ganancias proporcional e integral. ver Fig. 5  
global Emax Emin; % Limites superiores e inferiores del antiwinding-up  
Ts=570e-6;      %Tiempo de muestreo  
Td=1.6666666e-3;% Retardo de transporte  
R=0.4;          % Resistencia de la planta  
L=120e-3;       % Inductancia de la planta  
Emax=400;  
Emin=-400;  
Oo=0;  
Kz=39.5;  
az=0.93226;  
Kpz=Kz*az;      % simple pirotecnia matemática a partir de Fig. 5 Kpz=Kz*az  
Kiz=Kz*(1-az);  % simple pirotecnia matemática a partir de Fig. 5 Kiz=Kpz(1-az)
```

## Rutina PI\_Tarea

```
% Esta función implementa un controlador
% PI incluyendo antiwinding up
%
function [Vout] = PI(error)
global Oo Ol Kpz Kiz;
global Emax Emin;
% controlador PI
Ol=Oo+Kiz*error;    %Integrador
Vout=Ol+Kpz*error; % I+P del controlador
%
% Rutina antiwindind-up.
%
if (Vout>Emax)
    Vout=Emax;
elseif (Vout<Emin)
    Vout=Emin;
else
    Oo=Ol; %Integra solo si la salida está entre Emax y Emin.
end;
return
```



Electrónica de potencias y accionamientos EL7032-1

