DEPARTAMENTO DE ELECTRÓNICA Y AUTOMÁTICA
FACULTAD DE INGENIERÍA – UNIVERSIDAD NACIONAL DE SAN JUAN

Informe de Practica Nº7

CONTROL DE VELOCIDAD PARA MOTOR DE CORRIENTE CONTINUA

Asignatura: CONTROL 3
Ingeniería Electrónica

Autores (Grupo Nº 4):
Avila Juan Agustín - Registro 26076
Encina Leandro Nicolás - Registro 27044
Albornoz Rubén Fernando - Registro 9827

1º Semestre
Año 2020

1 Consigna

Realice un control de velocidad empleando un controlador PID modificado para:

Una referencia de ω_{des}=1.5rps

Un período de muestreo de T_0=5ms

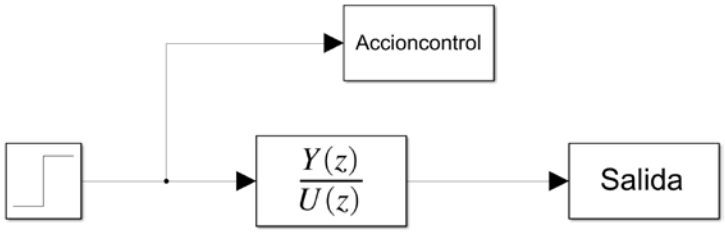
Teniendo en cuenta que el modelo del motor es el siguiente (rps/V):

$$H(z) = \frac{0.0008826\ z}{z^2\ - 1.864z + \ 0.8694} \tag{1}$$

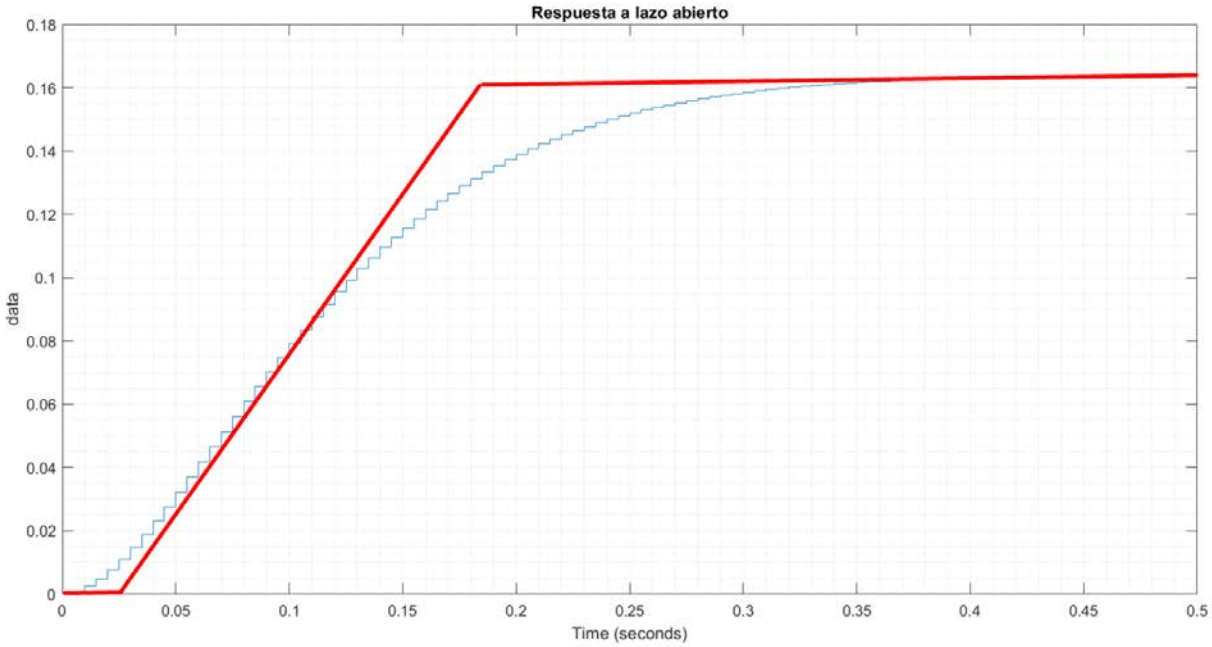
2 Resolución

2.1 Obtención de parámetros

Primero, se prueba la FT del motor a lazo abierto para ver su comportamiento, y según eso obtener los parámetros Ke, Tu y Tg utilizando el método de Ziegler-Nichols a lazo abierto.



La respuesta obtenida es la siguiente:




```
T0=0.005; fin=200;
```

```
t(1)=0;
nd=[0 .0008826 0];  %valores de la FT discreta
dd=[1 -1.864 .8694];%para simulink
ref=1.5;            %entrada de referencia

% Primer prueba midiendo valores en lazo abierto:
sim('motorLA.slx');
plot(Salida),grid minor,xlim([0 .5]);title('Respuesta a lazo
abierto');
%A partir de la grafica anterior se estiman los parametros
Tu=.03; %Tiempo de retardo puro estimado
Tg=.14; %Tiempo de subida estimado
Ke=.165;%Valor final de ganancia
K=((1.2*Tg)/(Ke*(Tu+(T0/2))))-((.3*Tg*T0)/(Ke*(Tu+(T0/2))^2));
Ki=(.6*Tg)/(K*Ke*(Tu+(T0/2))^2);
Td=.5*Tg/(K*Ke);
kp=K
ki=K*Ki
kd=K*Td
```

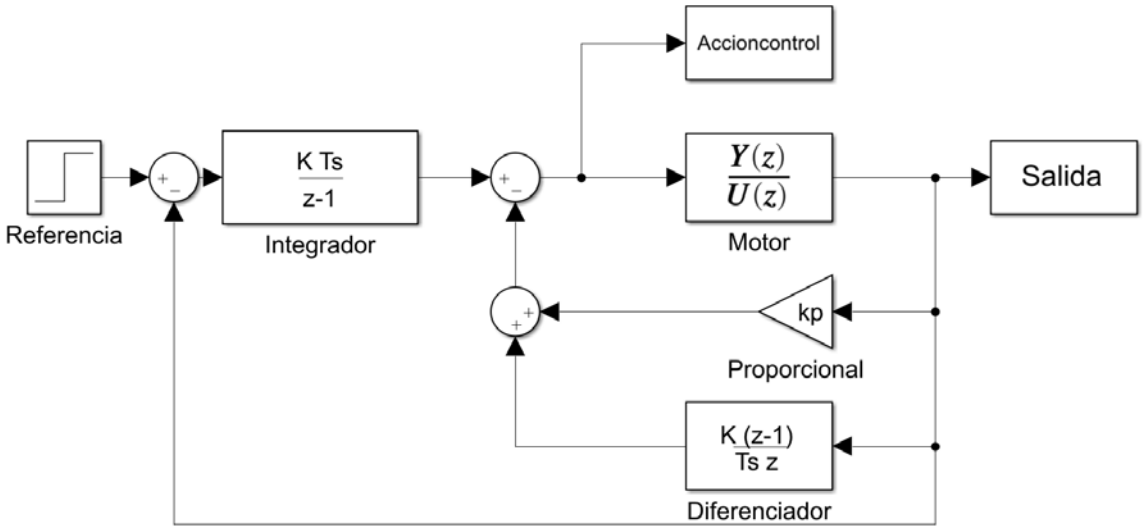
Los parámetros obtenidos tienen los siguientes valores:

kp = 30.1237

ki = 481.9796

kd = 0.4242

Y se arma en simulink el sistema con el controlador PID modificado:



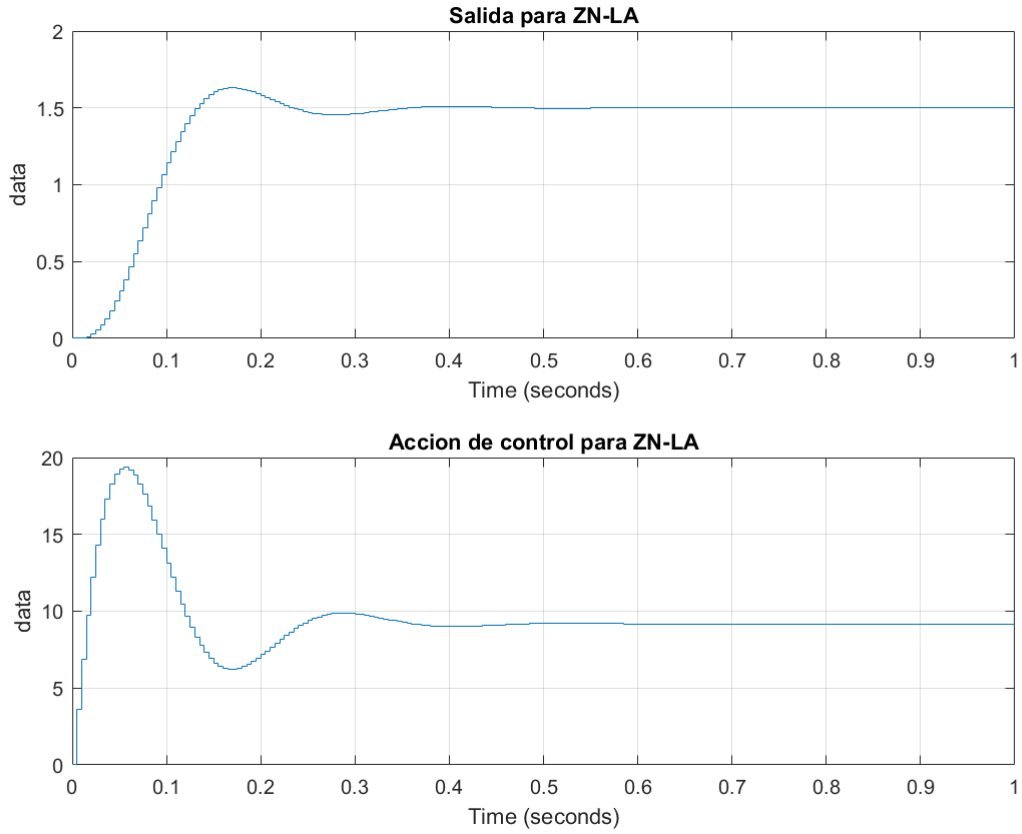
Utilizando los parámetros obtenidos por ZN-LA.

A continuación, se simula el sistema y se observa si cumple con las especificaciones (acción de control menor a 12):

```
sim('motor.slx');
subplot(211),plot(Salida),grid;title('Salida para ZN-LA')
subplot(212),plot(Accioncontrol),grid;
title('Accion de control para ZN-LA');
Max=max(Accioncontrol.Data) %Valor maximo de la accion de control
```

Y se obtienen los siguientes resultados:

Max= 19.3619



2.2 Ajuste de parametros

Se observa que el controlador no cumple con los requerimientos del sistema, por lo tanto se ajustan los parametros por prueba y error hasta encontrar valores que sean adecuados. Finalmente, se llega a los siguientes valores:

```
kp=30;
ki=280;
kd=.5;
```

2.3 Ecuación recursiva de la acción de control

Con los valores de Kp, Ki y Kd se determinan los valores A, B, C y D para la ecuación diferencial de la acción de control u(k):

$$u(k) = u(k-1) + A * y(k) + B * y(k-1) + C * y(k-2) + D * e(k-1) \quad (2)$$

```
A=-(kp+(kd/T0))
B=(kp+(2*kd/T0))
C=-kd/T0
D=ki*T0
```

A = -130

B = 230

C = -100

D = 1.4000

Con lo cual la ecuación recursiva del controlador PID modificado es la siguiente:

$$u(k) = u(k-1) - 130 * y(k) + 230 * y(k-1) - 100 * y(k-2) + 1.4 * e(k-1) \quad (3)$$

2.4 Modelado en matlab

Con estos valores, se realiza el código recursivo en matlab:

```
for k=1:1:fin
    if (k>1)
        t(k)=t(k-1)+T0;
    end
    % La accion de control esta incluida en el mismo if
    % que el modelo del motor para simplificar el codigo
    if (k>2)
        y(k)=1.864*y(k-1)-.8694*y(k-2)+.0008826*u(k-1);     %Salida
        u(k)=u(k-1)+A*y(k)+B*y(k-1)+C*y(k-2)+D*(ref-y(k-1));
                                                            %Accioncontrol
    else
        if (k>1)
            y(k)=1.864*y(k-1)+.0008826*u(k-1);              %Salida
            u(k)=u(k-1)+A*y(k)+B*y(k-1)+D*(ref-y(k-1));
                                                            %Accion control
        else
            y(k)=0;
            u(k)=0;
        end
    end
    % establecimiento del período de muestro
    pause(.005);
end
```

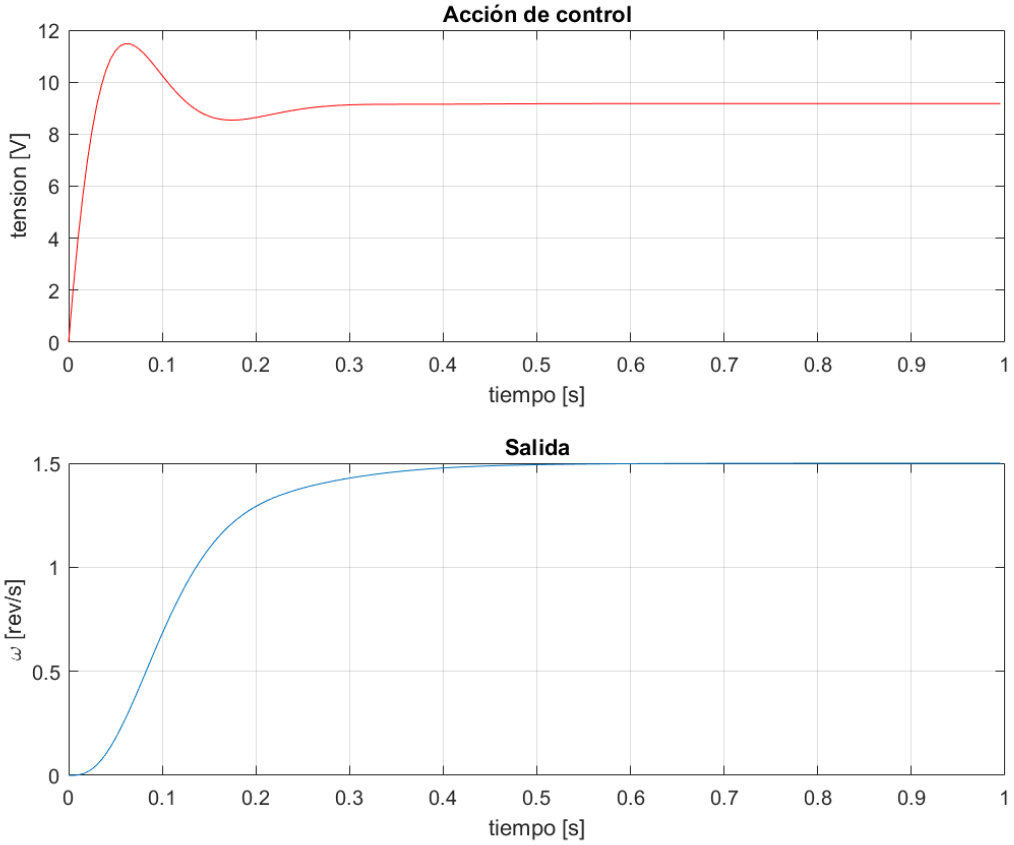
Luego, se comprueba que la acción de control no exceda los 12V y se grafican los resultados:

```
Max=max(u) %Para comprobar que no exceda los 12V
figure();subplot(2,1,1);
plot(t(1:fin),u(1:fin),'r');title("Acción de control");grid;
xlabel("tiempo [s]");ylabel("tension [V]");ylim([0 12]);

subplot(2,1,2); plot(t(1:fin),y(1:fin));grid;
title("Salida");xlabel("tiempo [s]");ylabel("\omega [rev/s]");
```

Max = 11.4750

Por lo tanto la acción de control no supera los 12V.

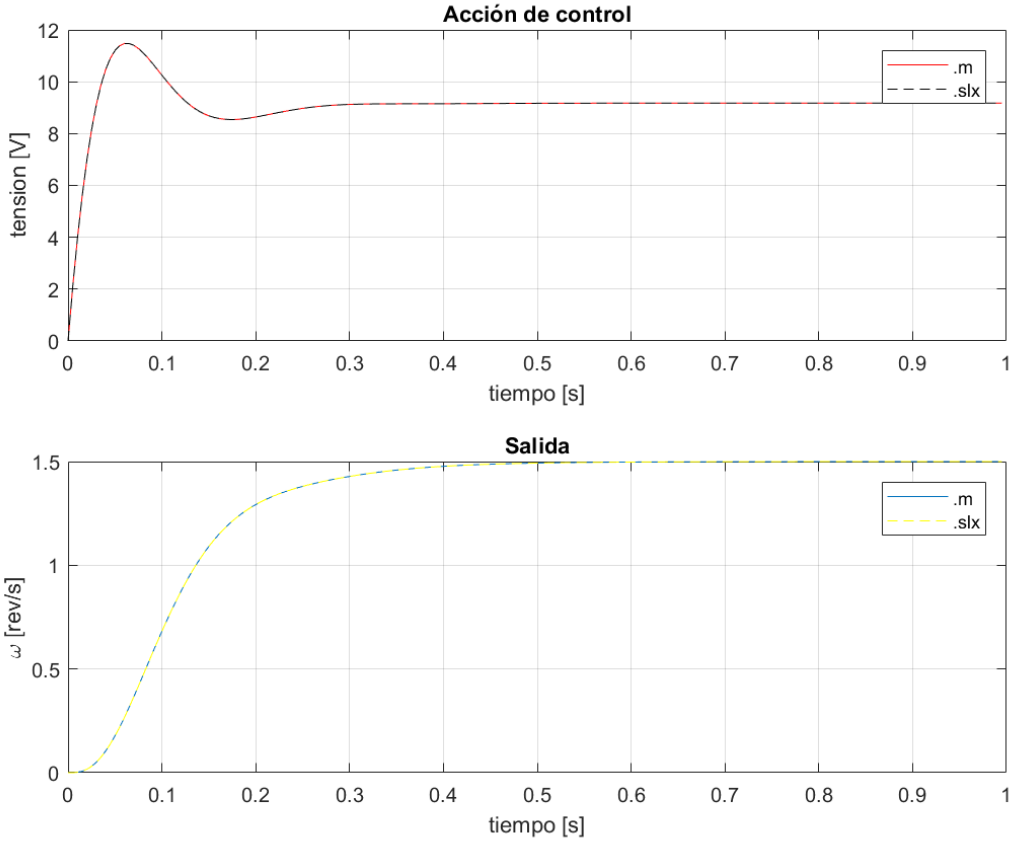


2.5 Comparacion entre modelo de MatLab y modelo de Simulink

Finalmente, se simula el modelo de simulink con los nuevos parámetros y se compara con los resultados obtenidos en matlab:

```
%% Comparacion entre los dos metodos
sim("motor.slx");
figure()
subplot(2,1,1);
plot(t(1:fin),u(1:fin),'r'); hold on;
plot(t(1:fin),Accioncontrol.Data(1:fin),'k--');
legend(".m",".slx");
title("Acción de control");grid;
xlabel("tiempo [s]");
ylabel("tension [V]");ylim([0 12]);

subplot(2,1,2);
plot(t(1:fin),y(1:fin));grid; hold on;
plot(tout(1:fin),Salida.Data(1:fin),'y--');
legend(".m",".slx");
title("Salida");
xlabel("tiempo [s]");
ylabel("\omega [rev/s]");
```



Se observa asi que ambos modelos coinciden, siendo iguales las respuestas y las acciones de control.