Aplicación de *Matlab* para el análisis de sistemas LTI

Sonia Vadalá Barake*, Eduardo Zurek Varela**

Fecha de recepción: Septiembre 1 de 199

Resumen

Este artículo pretende ilustrar la utilización de Matlab *para analizar el efecto de estrategias de control proporcional, proporcional-derivativo, proporcional-integral y proporcional-integral-derivativo en la respuesta temporal de Sistemas Lineales e Invariantes en el Tiempo(LTI).*

Palabras claves: Control proporcional, proporcional-derivativo, proporcional-integral-derivativo, proporcional-integral, LTI, Matlab.

Abstract

This article tries to ilustrate the use of Matlab *to analize the effect of proportional, proportional-derivative, proportional-integral and proportional-integral-derivative control strategies in the time response of Linear Time Invariant Systems (LTI).*

Key words: Proportional, proportional-derivative, proportional-integral, proportional-integral-derivative control, LTI, Matlab.

1. INTRODUCCIÓN

Consideremos un sistema de control retroalimentado con un controlador en cascada que actúa sobre una planta cuya función de transferencia es de segundo orden:

- *Planta:* Sistema objeto de control
- *Controlador:* Dispositivo físico mediante el cual se lleva a cabo la acción de control o regulación de la energía suministrada al sistema.

La Función de transferencia de la planta es la siguiente:

* Ingeniera Eléctrica, Universidad de los Andes; profesora Departamento de Ingeniería Eléctrica y Electrónica, Universidad del Norte; miembro del Grupo de Investigación del Laboratorio de Automatización y Robótica y del Grupo de Investigación en Sistemas de Potencia.
(E-mail: svadala@guayacán.uninorte.edu.co).

** Ingeniero de Sistemas, Universidad del Norte; profesor Departamento de Ingeniería de Sistemas, Universidad del Norte; miembro del Grupo de Investigación del Laboratorio de Automatización y Robótica *(e-mail: ezurek@guayacan.uninorte.edu.co).*

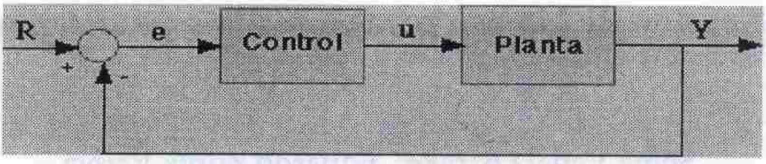


Figura 1. Sistema con realimentación unitaria

$$Planta(s) = \frac{(frec\_natural)^2}{s + (2*coef\_amortiguamoento*frec\_natural)*s + (frec\_natural)^2}$$

El controlador que se aplica a este sistema posee la siguiente función de transferencia:

$$K_p + \frac{K_i}{S} + K_d * S = \frac{K_d * S^2 + K_p * S + K_i}{S}$$

Este controlador posee tres parámetros: Ganancia proporcional (K_p), ganancia integral (K_i) y ganancia derivativa (K_d); y una variable: error (e), que representa el error del sistema.

El «error» se define como la diferencia entre la entrada de referencia (R) y la señal de retroalimentación (Y, en este caso, es la misma salida):

$$e(s) = R(s) - Y(s)$$

El controlador recibe esta señal y genera una señal de control *(u)* dependiendo de:

- *La cantidad del error:* La acción de control generada depende de la ganancia proporcional (K_p).
- *La integral del error:* La acción de control generada depende de la ganancia integral (K_i).
- *La derivada del error:* La acción de control generada depende de la ganancia derivativa (K_d).

$$u = K_p * e + K_i * \int e * dt + K_d * \frac{\partial e}{\partial t}$$

La señal *(u)*, a la salida del controlador, es igual a la magnitud del error por la ganancia proporcional (K_p), más la magnitud de la integral del error por la ganancia integral (K_i), más la magnitud de la derivada del error por la ganancia derivativa (K_d). Esta señal *(u)* se envía a la planta y se produce una señal de salida (Y). Esta nueva salida (Y) se envía de nuevo al sensor para determinar la nueva señal de error (e). El controlador utiliza esta nueva señal de error para calcular nuevamente su derivada y su integral. Este proceso se repite una y otra vez.

El objetivo de este artículo es mostrar la aplicacion de las herramientas de *software* Matlab para el análisis de los efectos de variar estos parámetros en la respuesta temporal de sistemas LTI.

2. INTERPRETACIÓN EN EL DOMINIO DEL TIEMPO DE LAS ESTRATEGIAS DE CONTROL PROPORCIONAL, INTEGRAL Y DERIVATIVO

Para ilustrar el efecto de los controladores debemos considerar, inicialmente, la respuesta de la planta ante una entrada escalón unitario, como se observa en la siguiente gráfica:

2.1. **Controlador proporcional (P)**

El controlador proporcional es el tipo más simple de controlador; la ecuación con que se describe su funcionamiento es la siguiente:

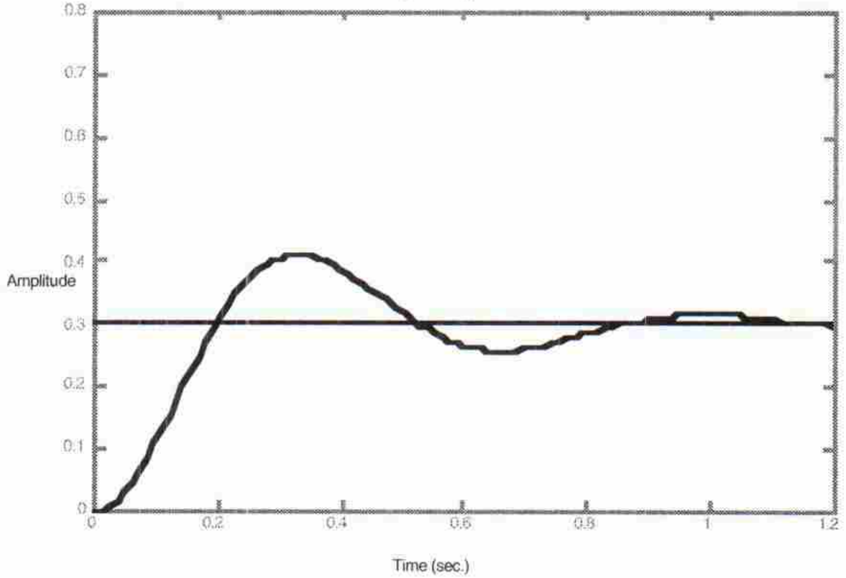
$$u = K_p * e + \overline{u}$$

donde:

- u: Salida del controlador
- K_p: Ganancia del controlador
- e: Señal de error
- $\overline{u}$ Valor base. El significado de este valor es la salida del controlador cuando el error es cero. Generalmente se fija durante la calibración del controlador.

En la ecuación del controlador proporcional se puede observar que la salida es proporcional al error entre el punto de control y la variable que se

Step Response



Gráfica 1. Sistema de segundo orden sin control

controla; la proporcionalidad la da la ganancia del controlador, K; con esta ganancia o sensibilidad del controlador se determina cuánto se modifica la salida del controlador con cierto cambio de error.

Los controladores que son únicamente proporcionales tienen la ventaja de que sólo cuentan con un parámetro de ajuste, K_p; sin embargo, tienen una gran desventaja, ya que operan con una desviación, o error estacionario en la variable que se controla. Cuanto mayor sea el valor de K_p, tanto menor es la desviación, pero la respuesta del proceso se hace más oscilatoria; sin embargo, para la mayoría de los procesos existe un valor máximo de K, más allá del cual el proceso se hace inestable, el cual se conoce como «ganancia última K».

Muchos fabricantes de controladores no utilizan el término «ganancia» para designar la cantidad de sensibilidad del controlador, sino el término «Banda Proporcional», PB. La relación entre la ganancia y la banda proporcional se expresa mediante:

$$PB = \frac{100}{K_p}$$

Se utiliza el término «100» porque la PB se conoce generalmente como «Porcentaje de Banda Proporcional». La banda proporcional se refiere al error (% de la variable que se controla) que se requiere para llevar la salida del controlador del valor más bajo al más alto.

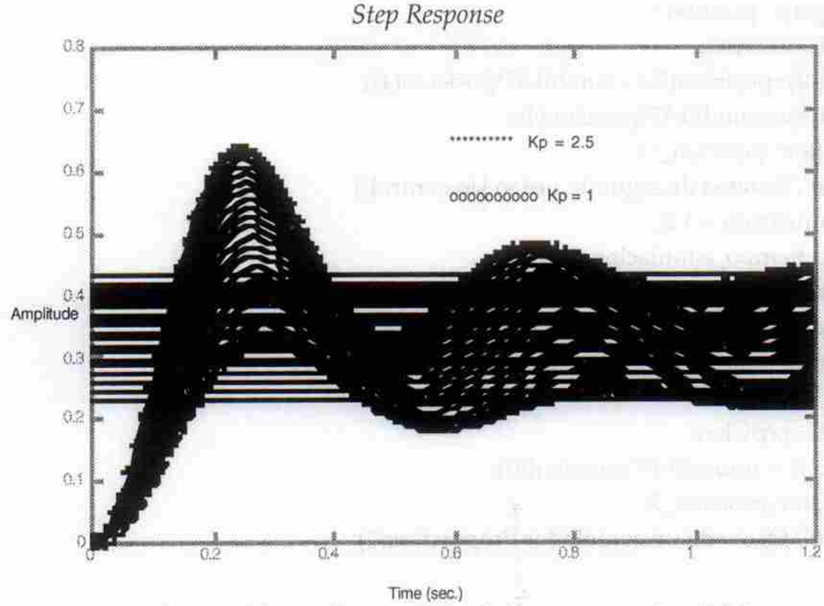
Un controlador proporcional ($K_p > 0$, $K_i = 0$, $K_d = 0$) reduce el tiempo de subida pero no elimina nunca el error de estado estable. Para ilustrar este efecto, veamos el siguiente algoritmo en Matlab:

```
%Programa para observar los efectos de un Contolador Proporcional
clear all
close all
%
%Definición de la Funcion de transfrencia
frec_natural=10;
coef_amortiguacion = 0.3;
ganancia = 0.3;
ceros_planta = [ganancia*frec_natural^2];
polos_planta = [1,2*coef_amortiguacion*frec_natural,frec_
natural^2];
planta = tf(ceros_planta,polos_planta);
Kp = 1;
% El controlador proporcional se asume inicialmente en 1
% y dentro de un ciclo repetitivo se incrementa hasta llegar a 2.5
func_transfer_total = Kp * planta / (1 + Kp * planta);
```

```
figure(1)
posicion=get(1,'position');
posicion_1=posicion;
posicion_1(2)=posicion(2)+ round(0.55*posicion(4));
posicion_1(4)=round(0.45*posicion(4));
set(1,'position',posicion_1)
set(1,'name','Sistema de segundo orden sin control')
tiempo_simulacion = 1.2;
step(planta,tiempo_simulacion,'k')
title('Sistema de segundo orden sin control')
%Efecto de un contolador Proporcional asumiendo como entrada
%un escalón unitario
figure(2)
posicion_2 = posicion;
posicion_2(4) = round(0.45*posicion(4));
set(2,'position',posicion_2)
set(2,'name','Efecto de un contolador Proporcional')
hold on
Kp = 1;
func_transfer_total = Kp * planta / (1 + Kp * planta);
step(func_transfer_total,tiempo_simulacion,'ko')
pause(0.01)
for Kp=1:1.5/19:2.5,
  func_transfer_total = Kp * planta / (1 + Kp * planta);
  step(func_transfer_total,tiempo_simulacion,'k')
  pause(0.01)
end
Kp = 2.5;
func_transfer_total = Kp * planta / (1 + Kp * planta);
step(func_transfer_total,tiempo_simulacion,'k*')
pause(0.01)
ejes = axis;
text(0.5*ejes(2),0.7*ejes(4),'ooooooooo  Kp = 1')
text(0.5*ejes(2),0.8*ejes(4),'*********  Kp = 2.5')
title('Efecto de un Controlador Proporcional
```

Los resultados arrojados por la simulación en Matlab se ilustran en la gráfica 2.

En dicha 2 se muestra cómo el controlador proporcional reduce tanto el tiempo de subida como el error de régimen permanente; se incrementa el sobreimpulso y se disminuye, en una pequeña cantidad, el tiempo de establecimiento.



Gráfica 2. Efecto de un controlador proporcional (P)

En los casos en el que el proceso se controla dentro de una banda del punto de control, los controladores proporcionales son suficientes; sin embargo, en los procesos en el que el control debe estar en el punto de control, los controladores proporcionales no proporcionan un control satisfactorio.

2.2. Controlador proporcional-derivativo (PD)

Este controlador se utiliza en los procesos en los que es posible utilizar un controlador proporcional, pero se desea cierta cantidad de anticipación. La ecuación con la que se describe su funcionamiento es la siguiente:

$$u = \overline{u} + K_p * e + K_d * \frac{\partial e}{\partial t}$$

Una desventaja del controlador proporcional-derivativo es que opera con una desviación en la variable que se controla; sin embargo, un controlador PD puede soportar mayor ganancia de lo que resulta una menor desviación que cuando se utiliza un controlador únicamente proporcional en el mismo circuito.

Se dice que el control derivativo es «anticipativo» porque al conocer la pendiente el controlador puede anticipar la dirección del error y emplearla para controlar mejor el proceso. Normalmente, en los sistemas lineales, si la pendiente debida a la entrada escalón es muy grande, subsecuentemente ocurrirá un sobrepaso alto. El control derivativo mide la pendiente instantánea de *e(t)*, predice el sobrepaso grande adelante en el tiempo, y hace un esfuerzo correctivo antes de que el sobrepaso excesivo ocurra.

El control derivativo afecta el error estacionario de un sistema sólo si el error en estado estacionario varía con el tiempo. Si el error en estado estacionario de un sistema es constante con respecto al tiempo, la pendiente de esta señal es igual a cero, y la porción derivativa del controlador no provee ninguna entrada al proceso; pero si el error en estado estacionario se incrementa con el tiempo, se genera otra vez un par en proporción a $\partial e/\partial t$, lo cual reduce la magnitud del error.

Un controlador proporcional-derivativo ($K_p > 0$, $K_i = 0$, $K_d > 0$) reduce tanto el sobreimpulso como el tiempo de establecimiento y tiene poco efecto sobre el tiempo de subida y el error en régimen permanente. Para ilustrar este efecto, veamos el siguiente algoritmo en Matlab:

```
%Programa para observar los efectos de un
%Contolador Proporcional-Derivativo
clear all
close all
%
%Definición de la Funcion de transfrencia
frec_natural=10;
coef_amortiguacion = 0.3;
ganancia = 0.3;
ceros_planta = [ganancia*frec_natural^2];
polos_planta = [1,2*coef_amortiguacion*frec_natural,
frec_natural^2];
planta = tf(ceros_planta,polos_planta);
Kp = 2.5;
Kd = 0;
% Para el ejemplo del controlador proporcional-derivativo
% se asume inicialmente en Kp = 2.5 y Kd = 0
% y dentro de un ciclo repetitivo se incrementa Kd hasta llegar a 0.7
controlador= tf([Kd,Kp],[1]);
func_transfer_total = controlador * planta / (1 + controlador * planta);
figure(1)
posicion=get(1,'position');
posicion_1=posicion;
posicion_1(2)=posicion(2)+ round(0.70*posicion(4));
posicion_1(4)=round(0.35*posicion(4));
set(1,'position',posicion_1)
set(1,'name','Sistema de segundo orden sin control')
```

```
tiempo_simulacion = 1;
step(planta,tiempo_simulacion,'k')
title('Sistema de segundo orden sin control')
% Efecto de un contolador Proporcional-Derivativo asumiendo como entrada
% un escalón unitario
figure(2)
posicion_2 = posicion;
posicion_2(2)=posicion(2)+ round(0.25*posicion(4));
posicion_2(4) = round(0.35*posicion(4));
set(2,'position',posicion_2)
set(2,'name','Efecto de un controlador Proporcional-Derivativo PD')
hold on
valores1 = [0:0.2/14:0.2];
Kp = 2.5;
Kd = 0;
controlador= tf([Kd,Kp],[1]);
func_transfer_total = controlador * planta / (1 + controlador * planta);
step(func_transfer_total,tiempo_simulacion,'ko')
pause(0.01)
for Kd=valores1,
  controlador= tf([Kd,Kp],[1]);
  func_transfer_total = controlador * planta / (1 + controlador * planta);
  step(func_transfer_total,tiempo_simulacion,'k')
  pause(0.01)
end
Kp = 2.5;
Kd = 0.2;
controlador= tf([Kd,Kp],[1]);
func_transfer_total = controlador * planta / (1 + controlador * planta);
step(func_transfer_total,tiempo_simulacion,'k*')
pause(0.01)
ejes = axis;
text(0.5*ejes(2),0.7*ejes(4),'ooooooooo  Kp = 2.5, Kd = 0')
text(0.5*ejes(2),0.85*ejes(4),'********** Kp = 2.5, Kd = 0.2')
title('Efecto de un controlador proporcional-derivativo')
figure(3)
posicion_3 = posicion;
posicion_3(2)=posicion(2)- round(0.20*posicion(4));
posicion_3(4) = round(0.35*posicion(4));
set(3,'position',posicion_3)
set(3,'name','Efecto de un controlador Proporcional-Derivativo PD')
```

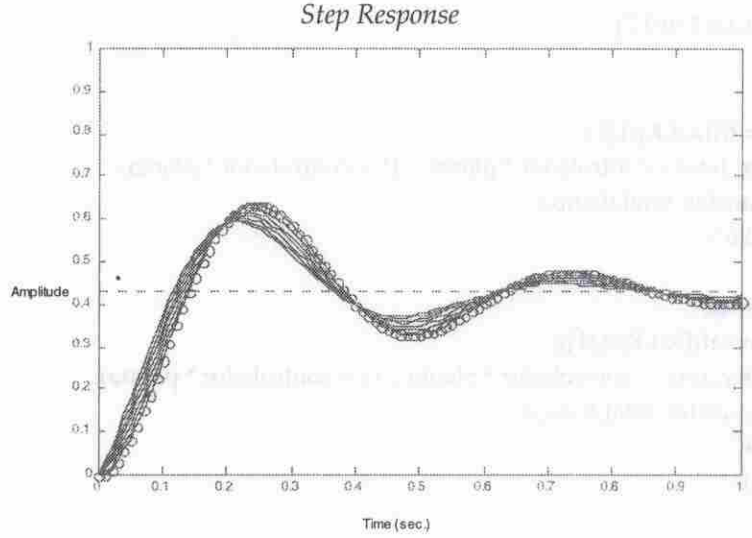
```
hold on
valores2 = [0.2:0.5/9:0.7];
Kp = 2.5;
Kd = 0.2;
controlador= tf([Kd,Kp],[1]);
func_transfer_total = controlador * planta / (1 + controlador * planta);
step(func_transfer_total,tiempo_
simulacion,'ko')
pause(0.01)
for Kd=valores2,
  controlador= tf([Kd,Kp],[1]);
  func_transfer_total = controlador * planta / (1 + controlador * planta);
  step(func_transfer_total,tiempo_
simulacion,'k')
  pause(0.01)
end
Kp = 2.5;
Kd = 0.7;
controlador= tf([Kd,Kp],[1]);
func_transfer_total = controlador * planta / (1 + controlador * planta);
step(func_transfer_total,tiempo_
simulacion,'k*')
pause(0.01)
ejes = axis;
text(0.5*ejes(2),0.7*ejes(4),'ooooooooo  Kp = 2.5, Kd = 0.2')
text(0.5*ejes(2),0.85*ejes(4),'*********  Kp = 2.5, Kd = 0.7')
title('Efecto de un controlador proporcional-derivativo')
```

Los resultados arrojados por la simulación en Matlab se ilustran en la gráfica 3.

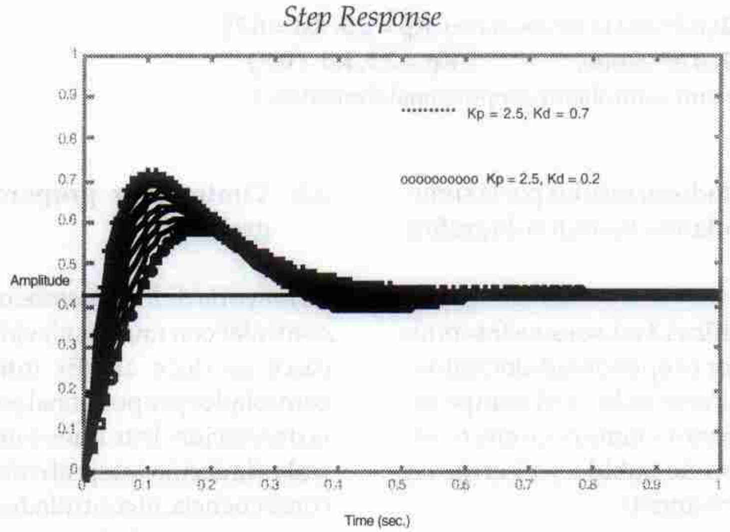
En las gráficas 3 y 4 se muestra cómo el controlador proporcional-derivativo reduce el sobreimpulso y el tiempo de establecimeiento y tiene poco efecto sobre el tiempo de subida y el error en régimen permanente.

2.3. Controlador proporcional-integral (PI)

La mayoría de los procesos no se pueden controlar con una desviación, y en estos casos se debe añadir inteligencia al controlador proporcional para eliminar la desviación. Este nuevo modo de control es la acción integral o de ajuste, y en consecuencia, el controlador se convierte en un controlador proporcional-integral (PI).



Gráfica 3. Efecto de un controlador proporcional-derivativo (PD)



Gráfica 4. Efecto de un controlador proporcional-derivativo (PD)

La siguiente es la ecuación descriptiva:

$$u = \overline{u} + K_p * e + K_i * \int e(t)dt$$

En un controlador PI, mientras el error está presente, el controlador se mantiene integrándolo y, por lo tanto, añadiéndolo a la salida hasta que el error desaparece; cuando éste es el caso, la salida del controlador se expresa mediante:

$$\overline{u} = u + K_i * \int[0] * dt$$

El hecho de que el error sea cero no significa que el término con la integral sea cero. Esto significa que el controlador integra una función de valor cero; o mejor aún, añade cero a su salida, con lo cual ésta se mantiene constante.

Los controladores proporcionales integracionales tienen dos parámetros de ajuste: La ganancia o banda proporcional y el tiempo de reajuste o rapidez de reajuste. La ventaja de este controlador consiste en que la acción de integración o de reajuste elimina la desviación.

Los efectos del controlador PI son:

- Se añade un cero en s = $-K_i / K_p$ a la función de transferencia de la trayectoria directa.
- Se añade un polo en s =0 a la función de trasferencia de la trayectoria directa; por lo tanto, el error en estado estacionario del sistema original se mejora en un orden, es decir, si el error en estado estacionario de una entrada dada es constante, el controlador PI lo reduce a cero (considerando que el sistema compensado permanece estable). Se debe tener mucho cuidado con la escogencia de los parámetros K_i y K_p para que un sistema no se vuelva inestable.

En un controlador proporcional - integral ($K_p > 0$, $K_i > 0$, $K_d = 0$) se reduce el valor de la ganancia proporcional K_p porque el controladdor integral también reduce el tiempo de subida e incrementa el sobreimpulso, tal y como hace el controlador proporcional (efecto doble). El controlador integral elimina el error en régimen permanente. Para ilustrar este efecto, veamos el siguiente algoritmo en Matlab:

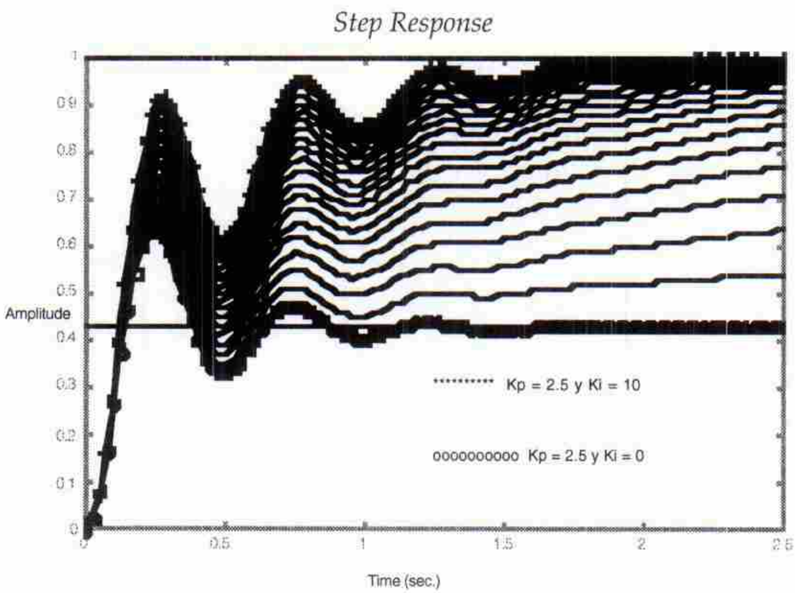
```
%Programa para observar los efectos de un
%Contolador Proporcional-Integral
%
clear all
close all

%Definición de la Funcion de transfrencia
frec_natural=10;
coef_amortiguacion = 0.3;
```

```
ganancia=0.3;
ceros_planta = [ganancia*frec_natural^2];
polos_planta = [1,2*coef_amortiguacion*frec_natural,frec_natural^2];
planta = tf(ceros_planta,polos_planta);
Kp = 2.5;
Ki = 0;
% Para el ejemplo del controlador proporcional-integral
% se asume inicialmente en Kp = 2.5 y Ki = 0
% y dentro de un ciclo repetitivo se incrementa Ki hasta llegar a 4
controlador= tf([Kp,Ki],[1,0]);
func_transfer_total = controlador * planta / (1 + controlador * planta);
figure(1)
posicion=get(1,'position');
posicion_1=posicion;
posicion_1(2)=posicion(2)+ round(0.55*posicion(4));
posicion_1(4)=round(0.45*posicion(4));
set(1,'position',posicion_1)
set(1,'name','Sistema de segundo orden sin control')
tiempo_simulacion = 2.5;
step(planta,tiempo_simulacion,'k')
% Efecto de un contolador Proporcional-Integral asumiendo como entrada
% un escalón unitario
figure(2)
posicion_2 = posicion;
posicion_2(4) = round(0.45*posicion(4));
set(2,'position',posicion_2)
set(2,'name','Efecto de un controlador Proporcional-Integral PI')
hold on
Kp = 2.5;
Ki = 0;
controlador= tf([Kp,Ki],[1,0]);
func_transfer_total = controlador * planta / (1 + controlador * planta);
step(func_transfer_total,tiempo_
simulacion,'ko')
pause(0.01)
for Ki=0:10/19:10,
  controlador= tf([Kp,Ki],[1,0]);
  func_transfer_total = controlador * planta / (1 + controlador * planta);
  step(func_transfer_total,tiempo_
simulacion,'k')
  pause(0.01)
```

```
end
Kp = 2.5;
Ki = 10;
controlador= tf([Kp,Ki],[1,0]);
func_transfer_total = controlador * planta / (1 + controlador * planta);
step(func_transfer_total,tiempo_
simulacion,'k*')
pause(0.01)
ejes = axis;
text(0.5*ejes(2),0.15*ejes(4),'oooooooooo Kp = 2.5 y Ki = 0')
text(0.5*ejes(2),0.3*ejes(4),'********** Kp = 2.5 y Ki = 10')
title('Efecto de un controlador proporcional-integral')
```

Los resultados arrojados por la simulación en Matlab se ilustran en la gráfica 5:

En esta gráfica se puede observar cómo un controlador PI reduce el tiempo de subida e incrementa el sobreimpulso, además de que elimina el error en régimen permanente.



Gráfica 5. Efecto de un controlador proporcional-integral (PI)

2.4. **Controlador proporcional-integral-derivativo (PID)**

Algunas veces se añade otro modo de control al controlador PI. Este nuevo modo de control es la «acción derivativa», que también se conoce como «rapidez de derivación o preactuación» y tiene como propósito anticipar hacia dónde va el proceso, mediante la observación de la rapidez para el cambio del error, su derivada. La ecuación descriptiva es:

$$u = K_p * e + K_i * \int e * dt + K_d * \frac{\partial e}{\partial t}$$

Con la acción derivativa se da al controlador la capacidad de anticipar hacia dónde se dirige el proceso, es decir, «ver hacia delante», mediante el cálculo de la derivada del error.

Los controladores PID se utilizan en procesos en los que las constantes de tiempo son largas. Ejemplos típicos de ello son circuitos de temperatura y los de concentración. Los procesos en que las constantes de tiempo son cortas (capacitancia pequeña) son rápidos y susceptibles al ruido del proceso. Son característicos de este tipo de proceso, los circuitos de control de flujos y los circuitos para controlar la presión en corrientes de líquidos. Los procesos en los que las constantes de tiempo son largas (capacitancia grande) son generalmente amortiguados y, en consecuencia, menos susceptibles al ruido; sin embargo, se debe estar alerta, ya que se puede tener un proceso con constante de tiempo larga; por ejemplo, un circuito de temperatura, en el que el transmisor sea ruidoso, en cuyo caso se debe reparar el transmisor antes de utilizar el controlador PID.

Los controladores PID tienen tres parámetros de ajuste: La ganancia o banda proporcional, el tiempo de reajuste o rapidez de reajuste y la rapidez derivativa.

Los controladores PID se recomiendan para circuitos con constantes de tiempo larga en los que no hay ruido. La ventaja del modo derivativo consiste en que proporciona la capacidad de ver hacia dónde se dirige el proceso.

Un controlador proporcional-integral-derivativo ($K_p > 0$, $K_i > 0$, $K_d > 0$) elimina el error en régimen permanente, no hay sobreimpulso y el tiempo de subida es rápido. Para ilustrar este efecto, veamos el siguiente algoritmo en Matlab:

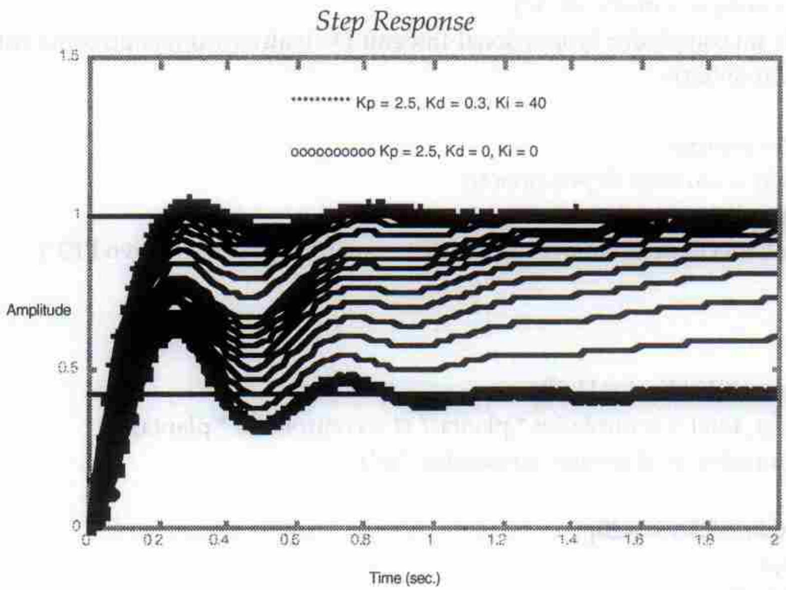
```
%Programa para observar los efectos de un
%Contolador Proporcional-Integral-Derivativo
%
clear all
close all
%Definición de la Funcion de transfrencia
```

```
% y dentro de un ciclo repetitivo se incrementa Ki hasta llegar a 20
% y Kd hasta llegar a 0.3
controlador= tf([Kd,Kp,Ki],[1,0]);
func_transfer_total = controlador * planta / (1 + controlador * planta);
figure(1)
posicion=get(1,'position');
posicion_1=posicion;
posicion_1(2)=posicion(2)+ round(0.55*posicion(4));
posicion_1(4)=round(0.45*posicion(4));
set(1,'position',posicion_1)
set(1,'name','Sistema de segundo orden sin control')
tiempo_simulacion = 2;
step(planta,tiempo_simulacion,'k')
% Efecto de un contolador Proporcional-Integral-Derivativo asumiendo como entrada
% un escalón unitario
figure(2)
posicion_2 = posicion;
posicion_2(4) = round(0.45*posicion(4));
set(2,'position',posicion_2)
set(2,'name','Efecto de un controlador Proporcional-Integral-Derivativo PID')
hold on
Kd = 0;
Ki = 0;
controlador= tf([Kd,Kp,Ki],[1,0]);
func_transfer_total = controlador * planta / (1 + controlador * planta);
step(func_transfer_total,tiempo_simulacion,'ko')
pause(0.01)
valores = [0:5/9:5,5:15/9:20];
for i=valores,
  Kd = i*(0.3/20);
  Ki = i*(40/20);
  controlador= tf([Kd,Kp,Ki],[1,0]);
  func_transfer_total = controlador * planta / (1 + controlador * planta);
  step(func_transfer_total,tiempo_simulacion,'k')
  pause(0.01)
end
Kd = 0.3;
Ki = 40;
controlador= tf([Kd,Kp,Ki],[1,0]);
func_transfer_total = controlador * planta / (1 + controlador * planta);
step(func_transfer_total,tiempo_simulacion,'k*')
```

```
pause(0.01)
ejes = axis;
text(0.3*ejes(2),0.8*ejes(4),'ooooooooo Kp = 2.5, Kd = 0, Ki = 0')
text(0.3*ejes(2),0.9*ejes(4),'********** Kp = 2.5, Kd = 0.3, Ki = 40')
title('Efecto de un controlador proporcional-integral-derivativo')
```

Los resultados arrojados por la simulación en Matlab se ilustran en la gráfica 6.

En esta gráfica se puede observar cómo un controlador PID elimina el error en régimen permanente, no hay sobreimpulso y el tiempo de subida es rápido.



Gráfica 6. Efecto de un controlador proporcional-integral-derivativo (PID)

Tabla 1. Efecto de los controladores proporcional, derivativo e integral

Respuesta lazo cerrado	**Tiempo de subida**	**Tiempo de establecimiento**	**Sobreimpulso**	**Error en régimen permanente**
K_p	Disminuye	Poca variación	Aumenta	Disminuye
K_i	Disminuye	Aumenta	Aumenta	Elimina
K_d	Poca variación	Disminuye	Disminuye	Poca variación

CONCLUSIONES

Un controlador proporcional (K_p) reduce el tiempo de subida pero no elimina nunca el error en régimen permanente. El control integral (K_i) elimina el error en régimen permanente pero empeora la respuesta transitoria. Un control derivatiavo (K_d) incrementa la estabilidad del sistema, reduce el sobreimpulso y mejora la respuesta transitoria. En la siguiente tabla se resumen los esfectos para cada controlador, K_p, K_d y K_i, sobre un sistema en lazo cerrado.

Referencias

[1] OGATA, Katsuhiko. *Ingeniería de Control.* 2ª ed. México, Prentice-Hall Hispanoamericana, 1993.

[2] ROHRS, Charles E, MELSA, Jaimes L., SHULTZ, Donald G. *Sistemas de Control Lineal.* México, McGraw-Hill, 1994.

[3] KUO, Benjamín. *Sistemas de Control Automático.* 7ª ed. México, Prentice-Hall Hispanoamericana, 1996.

DIRECCIONES DE INTERNET RELACIONADAS CON EL TEMA

http://harpo.isa.uma.es/eugenio/etm/es/pid
http://www.euiti.upv.es/innova/docum/unipie.html
http://www.manufacturing.net/magazine/cc/archives/1996/02/issues/na/02c152