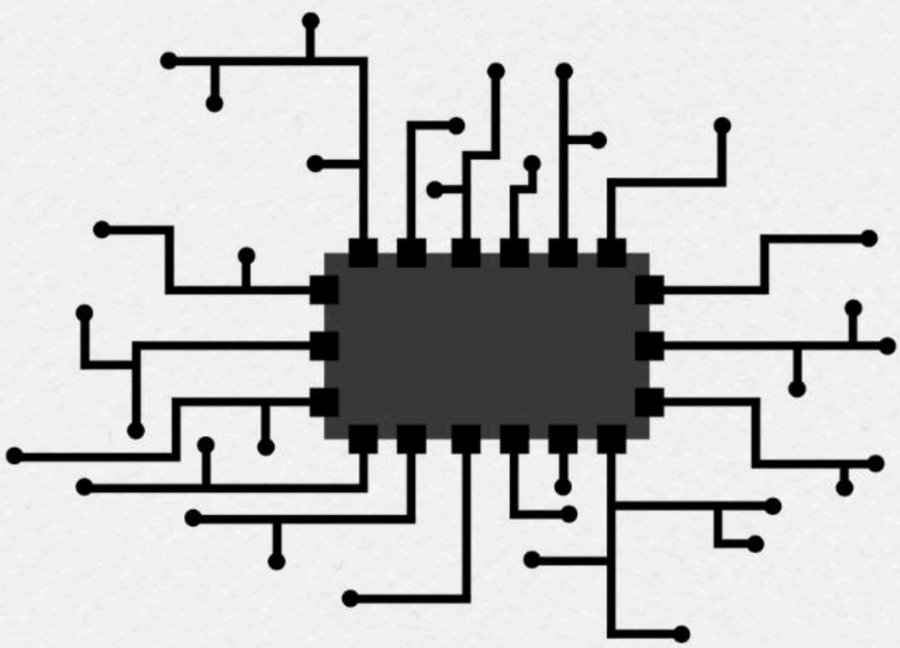
Docente: Dr. Carlos Iván Aldana López

Centro universitario de ciencias exactas e ingenierias

CONTROL I

Actividad 3



Equipo:
Gutiérrez Iñiguez Karla Goretti 219555062
Hernández López Daniel Alejandro 219567621
Martínez Valdez Isidro 222834487
Sánchez Mendoza Ricardo Yahir 222834347

NCR:159311
Clave:I9914

1.- Haga un resumen de máximo de dos cuartillas sobre el sintonizador "pidTuner" de Matlab y comente en que les puede servir esta herramienta de software.

El PID Tuner de Matlab es una herramienta que tiene el programa de Matlab para sintonizar un controlador distintos tipos de controladores, de los cuales el más famoso es el PID que por este lleva el nombre de dicha herramienta, este es un método de sintonización automática.

Funciona de la siguiente manera:

Para comenzar tienes que tener definida ya tu planta en forma de función de transferencia con la función tf(numerador, denominador).

Ahora procedemos a utilizar tal cual la App de PID Tuner que lo puedes hacer directamente de la pestaña superior de Apps y buscar el PID Tuner, como se muestra en la **Figura 1.1**, o también se puede hacer con código, que será como lo explicaremos.

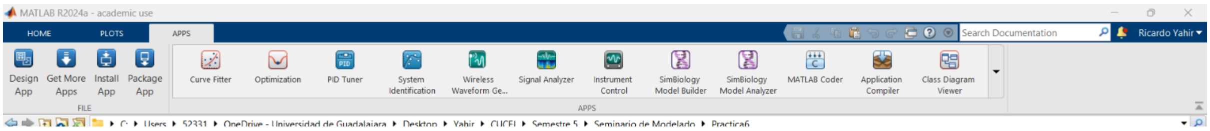


Figura 1.1 Cómo acceder al PID Tuner en las APPS de Matlab.

Con código es lo siguiente:

Para que funcione el PID Tuner ya tienes que tener la función de transferencia de tu sistema a lo que se llama planta, por lo que comencemos definiendo una función de transferencia con la función de matlab TranferFunction tomaremos como ejemplo la de un sistema masa,resorte, amortiguador.

$G(s) = \frac{1}{ms^2+bs+k}$ donde el valor de m= 6, el del b= 0.7 y el del k= 18

La **Figura 1.2** nos muestra el resultado de la función de transferencia en Matlab:

```
>>  G = tf([1],[6 0.7 18])

G =

          1
  ------------------
  6 s^2 + 0.7 s + 18

Continuous-time transfer function.
Model Properties
```

Figura 1.2 Definición de función de transferencia de la planta en Matlab.

ahora con la función PID Tuner escribimos y damos enter y con eso nos abrirá el PID Tuner de matlab:

```
>> pidTuner
```

la cual nos abre la ventana, como la que se observa en la **Figura 1.3.** Ya con el PID Tuner, importamos la planta abriendo la siguiente:

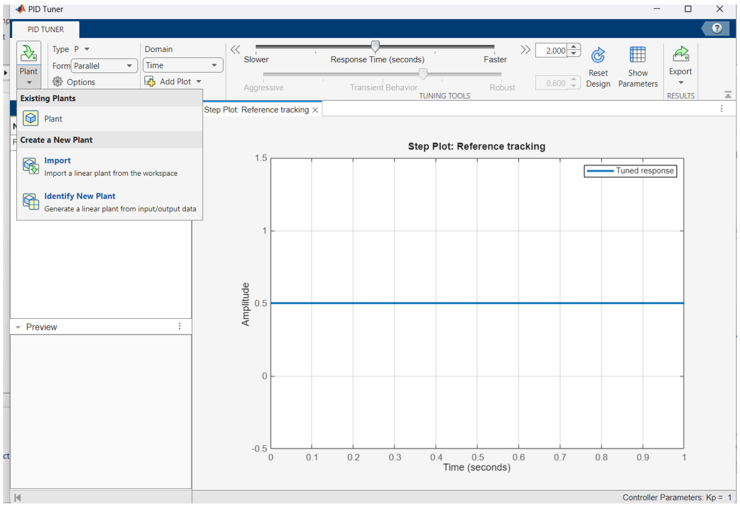
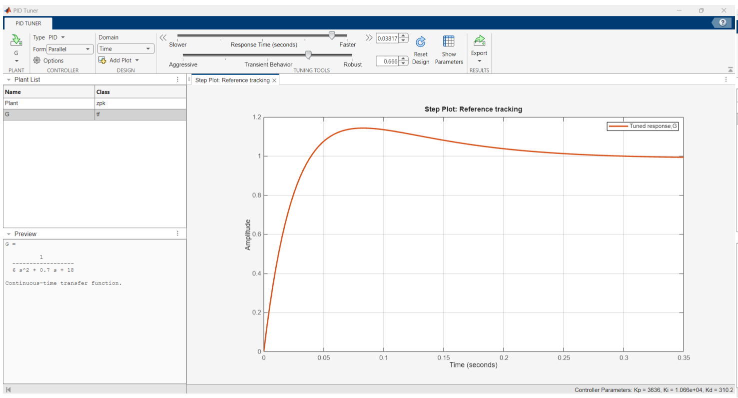
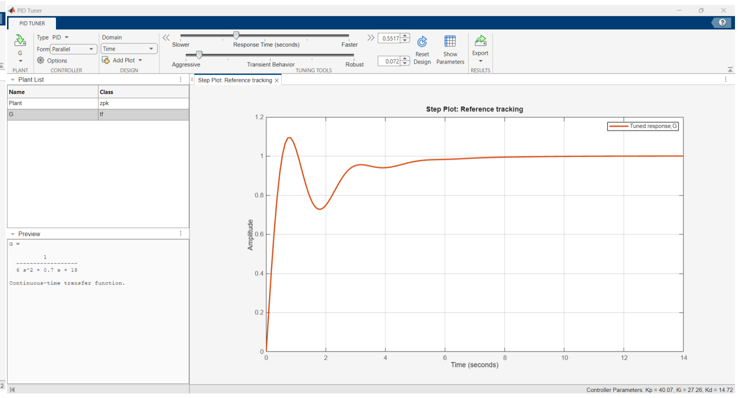


Figura 1.3 Imagen de la ventana de PID Tuner con la pestaña de importar planta.

Ya al importar la planta podemos comenzar a sintonizar de manera automática las ganancias de muchos tipos de controladores por ejemplo un PID, esto elige desplayando la pestaña que dice “type” eligiendo el tipo de controlador. Teniendo esto se puede sintonizar con distintos resultados jugando con las barras de arriba, las de “Slower” “Faster” donde te modifica el tiempo de respuesta es decir que tan rápido quieres que se acerque al valor deseado la señal de salida; modificando este valor en segundos y “Aggressive” “Robust” en donde te modifica qué tan agresivo o suave quieres que llegue a el estado estacionario la señal de salida tu sistema.

Figuras 1.4 y 1.5 Ejemplos de cómo se puede jugar con las ganancias en la herramienta PID Tuner.

Podrás ver que la herramienta te arroja ya directamente en la esquina inferior derecha los valores kp, ki y kd para tu control PID o el respectivo que hayas elegido. Podemos observar que lo puedes hacer más suave, moviendo más el sobreimpulso, y rápido como el ejemplo de la **Figura 1.4**; así como más agresivo pero lento como el de la **Figura 1.5**, depende de lo que busques, ahí puedes jugar hasta encontrar la respuesta deseada.

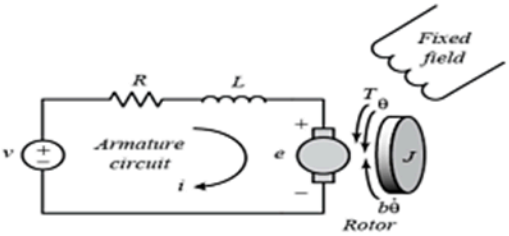
2.- Para el motor de la Figura 1, diseñe:

a) Un controlador PI

b) Un controlador PD

c) Un controlador PID

en lazo cerrado tal que su salida (posición θ) cumpla con las siguientes especificaciones: Error en Estado Estacionario menor a 2%, Tiempo de Asentamiento menor de 15s y un Sobreimpulso de menos del 20%. La referencia es un escalón unitario (1 rad). Hacer esto de manera detallada usando el método de sintonización de Ziegler-Nichols y puliendo las ganancias de manera manual con base en la tabla vista en clase. Graficar y comparar la acción de control de cada uno de los controladores diseñados y comentar los resultados.



$$J\ddot{\theta} + b\dot{\theta} = Ki$$

$$L\frac{di}{dt} + Ri = V - K\dot{\theta}$$

Figura 1. Motor y su modelo (***J=0.05 kg·m², b=0.07 N·m·s, K=0.04 N·m/A, R=4 ohm, L=1.7 H***).

Lo primero que tenemos que hacer para poder realizar los cálculos es obtener la función de transferencia del sistema. Aplicamos transformada de Laplace a las ecuaciones dadas, de la siguiente manera:

$$\mathcal{L}\left[J\frac{d^2\theta}{dt^2} + b\frac{d\theta}{dt} = Ki\right] \quad \textbf{Ec. 1.1}$$

$$\mathcal{L}\left[L\frac{di}{dt} + Ri = V - K * \frac{d\theta}{dt}\right] \quad \textbf{Ec. 2.1}$$

Ahora aplicamos la fórmula de la transformada de laplace de las derivadas sin olvidar que las condiciones iniciales son iguales a cero a **Ec. 1.1** y **Ec. 2.1**.

$$Js^2\theta(s) - J\cancel{s\theta}(0) - \cancel{J\theta}(0) + bs\theta(s) - b\theta\cancel{(0)} = KI(s)$$

$$Js^2\theta(s) + bs\theta(s) = KI(s) \quad \textbf{Ec. 1.2}$$

$$LsI(s) - L\cancel{I}(0) + RI(s) = V(s) - Ks\theta(s) + K\theta\cancel{(0)}$$

$$LsI(s) + RI(s) = V(s) - Ks\theta(s) \quad \textbf{Ec. 2.2}$$

A partir de ahora, ya tendremos las ecuaciones para obtener nuestra función de transferencia. Despejamos $I(s)$ de la **Ec. 1.2** y después factorizamos $\theta(s)$ de la siguiente manera:

$$\frac{Js^2\theta(s)+bs\theta(s)}{K} = I(s)$$

$$\frac{\theta(s)\,(Js^2+bs)}{K} = I(s) \quad \textbf{Ec. 1.3}$$

Ahora despejamos $V(s)$ de la **Ec. 2.2**

$$LsI(s) + RI(s) + Ks\theta(s) = V(s) \quad \textbf{Ec. 2.3}$$

Posteriormente sustituimos I(s) de la **Ec. 1.3** en la ecuación **Ec. 2.3,** factorizamos $\theta(s)$ y realizamos la suma de fracciones:

$$Ls\left(\frac{\theta(s)\,(Js^2+bs)}{K}\right) + R\left(\frac{\theta(s)\,(Js^2+bs)}{K}\right) + Ks\theta(s) = V(s)$$

$$\theta(s)\left[\left(\frac{Ls\,(Js^2+bs)}{K}\right) + \left(\frac{R(Js^2+bs)}{K}\right) + Ks\right] = V(s)$$

$$\theta(s)\left[\frac{LJs^3+ Lbs^2+ RJs^2+ Rbs + K^2s}{K}\right] = V(s) \quad \textbf{Ec. 2.4}$$

Ahora recordando que la función de transferencia se define como la transformada de Laplace de el cociente de la salida entre la entrada con condiciones iniciales igual a cero, despejamos **Ec. 2.4** y obtenemos nuestra función de transferencia

$$\frac{\theta(s)}{V(s)}\left[LJs^3 + Lbs^2 + RJs^2 + Rbs + K^2s\right] = K$$

$$\frac{\theta(s)}{V(s)} = \frac{K}{LJs^3+ Lbs^2+ RJs^2+ Rbs + K^2s}$$

$$\frac{\theta(s)}{V(s)} = \frac{K}{LJs^3+ s^2\,(Lb+ RJ) + s(Rb + K^2)} \quad \textbf{Ec. 2.5}$$

Posteriormente, sustituimos los valores de las constantes en nuestra función de transferencia (**Ec. 2.5**)

$$\frac{\theta(s)}{V(s)} = \frac{0.04}{(1.7)(0.05)s^3+ s^2\,((1.7)(0.07)+ (4)(0.05)) + s((4)\,(0.07)+ (0.04)^2)}$$

$$\frac{\theta(s)}{V(s)} = \frac{0.04}{0.085s^3+0.319\,s^2 + 0.2816s}$$

Ya con la función de transferencia de nuestro sistema, debemos observar con cuál método de Ziegler-Nichols podemos usar para sintonizar nuestro controlador. Si observamos la ecuación característica de nuestra función de transferencia, se puede factorizar por factor común “s”, y este al encontrarse en el denominador se obtiene un integrador, por lo tanto, el primer método de Ziegler-Nichols no se puede aplicar, así que usaremos el segundo método de Ziegler-Nichols para sintonizar nuestros controladores PI, PD y PID.

$$\frac{\theta(s)}{V(s)} = \frac{0.04}{s(0.085s^2+0.319s+0.2816)}$$

El segundo método nos dice que debemos de fijar $T_i = \infty$ y $T_d = 0$ y solo utilizar la acción de control proporcional. Debemos incrementar Kp hasta un valor crítico Kcr hasta que nuestro sistema llegue a presentar oscilaciones.

Lo primero que tenemos que hacer es obtener la función de transferencia del sistema en lazo cerrado con nuestro control proporcional como se observa en la **Figura 2.1**.

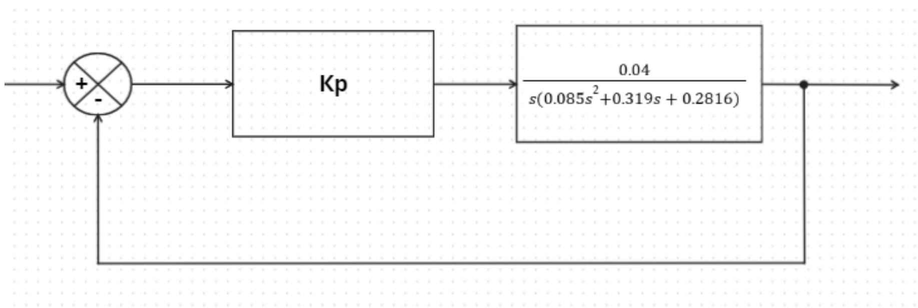


Figura 2.1 Planta en lazo cerrado con un control proporcional

$$\frac{\theta(s)}{V(s)} = \frac{\dfrac{0.04\,Kp}{0.085s^3+0.319\,s^2+0.2816s}}{1+\dfrac{0.04\,Kp}{0.085s^3+0.319\,s^2+0.2816s}} = \frac{\dfrac{0.04\,Kp}{\cancel{0.085s^3+0.319\,s^2+0.2816s}}}{\dfrac{0.085s^3+0.319\,s^2+0.2816s+0.04\,Kp}{\cancel{0.085s^3+0.319\,s^2+0.2816s}}} =$$

$$= \frac{0.04\,Kp}{0.085s^3+0.319\,s^2+0.2816s+0.04\,Kp}$$

Del resultado obtenemos nuestra ecuación característica, que es la que se encuentra en el denominador.

$$0.085s^3 + 0.319\,s^2 + 0.2816s + 0.04\,Kp = 0 \qquad \textbf{Ec. 2.6}$$

Para encontrar el valor de Kp debemos aplicar el criterio de estabilidad de Routh-Hurwitz a la **Ec. 2.6**:

$$\begin{array}{lll} s^3 & 0.085 & 0.2816 \\ s^2 & 0.319 & 0.04\,Kp \\ s^1 & b1 & 0 \\ s^0 & Kp & \end{array}$$

El valor de $b1$ corresponde a lo siguiente:

$$b1 = \frac{(0.319)(0.2816)-(0.085)(0.04Kp)}{0.319}$$

$$= \frac{0.0898304 - 0.0034Kp}{0.319}$$

Para obtener el valor de Kp igualamos a cero el valor de $b1$ y despejamos:

$$\frac{0.0898304 - 0.0034Kp}{0.319} = 0$$

$$0.0898304 - 0.0034Kp = 0(0.319)$$

$$0.0898304 - 0.0034Kp = 0$$

$$0.0898304 = 0.0034Kp$$

$$\frac{0.0898304}{0.0034} = Kp$$

$$Kp = 26.4207$$

Ocurrirá una oscilación sostenida cuando Kp sea igual a 26.4207, por lo que la ganancia crítica Kcr es igual a 26.4207, por lo tanto, nuestra ecuación característica es:

$$0.085s^3 + 0.319\,s^2 + 0.2816s + 1.0568 = 0 \quad \textbf{Ec. 2.7}$$

Lo siguiente que tenemos que hacer es encontrar la frecuencia de la oscilación, para eso sustituimos la variable s de la **Ec. 2.7** por $j\omega$.

$$0.085(j\omega)^3 + 0.319\,(j\omega)^2 + 0.2816(j\omega) + 1.0568 = 0$$

$$0.085j^3\omega^3 + 0.319j^2\omega^2 + 0.2816j\omega = 0 + 1.0568 = 0$$

Nota: Recordemos que $j^2 = -1$ y que $j^3 = -j$

$$-\ 0.085j\omega^3 - 0.319\omega^2 + 0.2816j\omega + 1.0568 = 0$$
$$0.085j\omega(-\ \omega^2 + 3.3128j\omega) + 0.319(-\ \omega^2 + 3.3128) = 0$$

La frecuencia de oscilación se encuentra en:

$$-\ \omega^2 + 3.3128 = 0$$
$$-\ \omega^2 =-\ 3.3128$$
$$\omega^2 = 3.3128$$
$$\sqrt{\omega^2} = \sqrt{3.3128}$$
$$\omega =\sqrt{3.3128}$$

A partir de obtener la frecuencia de oscilación sostenida y con esto se puede calcular el periodo de oscilación sostenida de la siguiente forma:

$$P_{cr} = \frac{2\pi}{\omega} = \frac{2\pi}{\sqrt{3.3128}} = 3.4520$$

Con estos resultados obtenidos podremos calcular las ganancias de nuestros controladores.

- Para el controlador PI, requerimos su función de transferencia, la cual es la siguiente:

$$G_c(s) = K_p\left(1 + \frac{1}{T_i s}\right)$$

Determinamos K_p y T_i con los valores de referencia de las tablas:

$$K_p = 0.45(26.4207) = 11.8893$$
$$T_i = \frac{1}{1.2}(3.4520) = 2.8766$$

Sustituimos los valores en la función de transferencia para obtener K_i:

$$11.8893\left(1 + \frac{1}{2.8766s}\right) = 11.8893 + \frac{11.8893}{2.8766s} = 11.8893 + \frac{4.1331}{s}$$

Por lo tanto:

$$K_i = 4.1331$$

Para analizar la respuesta de nuestro sistema a este controlador, lo graficamos ante una entrada escalón con el siguiente código en Matlab.

```
%% Sistema
planta = tf(0.04,[0.085 0.319  0.2816 0]);

%% Controlador PI
kp= 11.8893;
ki= 4.1331;
Gc=pid(kp,ki,0);
T= feedback(Gc*planta,1);
t= 0:0.01:50;
step(T,t)
```

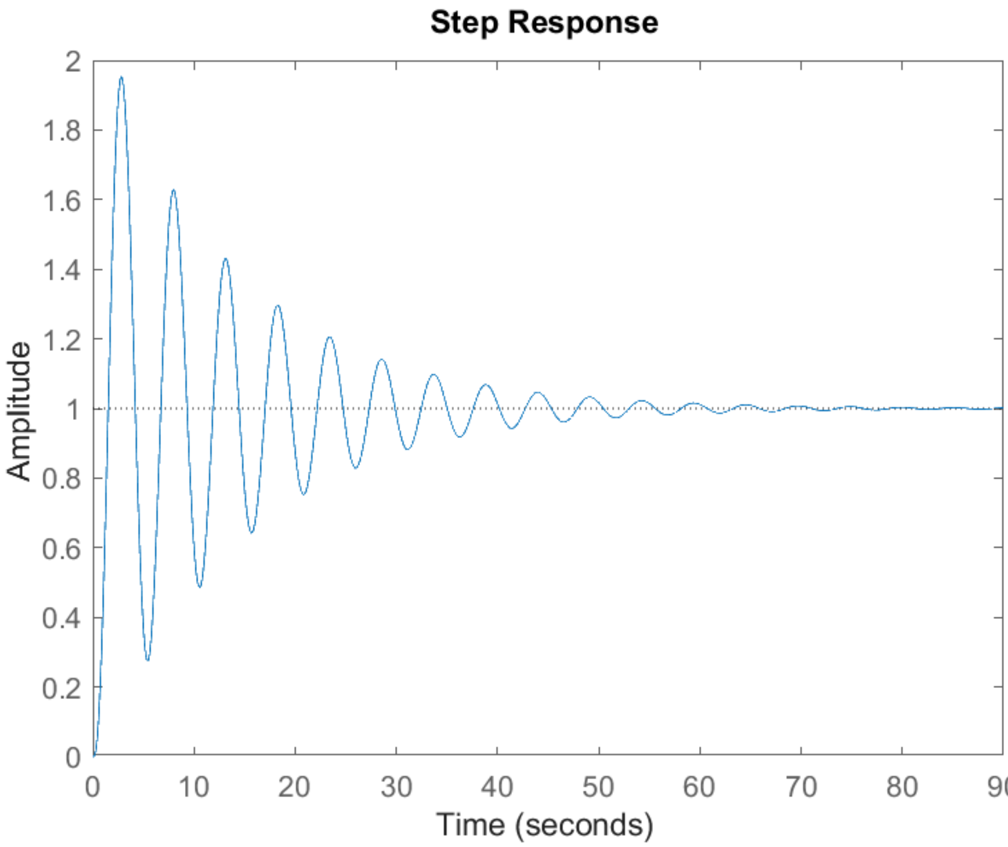


Figura 2.2 Respuesta de la planta en lazo cerrado con el controlador PI ante una entrada escalón

Se observa en la **figura 2.2** que aunque la respuesta cumple con el criterio de error en estado estacionario menor al 2%, presenta un tiempo de asentamiento superior a los 15 segundos y un sobreimpulso mayor al 20%. Esto podría deberse a una ganancia alta en el integrador, lo cual sabemos que incrementa el sobreimpulso. Este valor alto en la ganancia del integrador reduce el error en estado estacionario.

Para mejorar el desempeño y lograr una respuesta que cumpla con los requerimientos, es necesario ajustar tanto la ganancia del integrador como la ganancia proporcional. Reducir la ganancia del integrador ayudaría a disminuir el sobreimpulso y el tiempo de asentamiento, mientras que una ganancia proporcional ajustada puede compensar para mantener el error en estado estacionario aceptable.

Para el ajuste de las ganancias se tomará en cuenta la **tabla 1.1** la cual dicta cómo afecta el incremento independiente de cada ganancia ante el tiempo de respuesta, el sobre impulso, el tiempo de asentamiento, el error en estado estacionario y la estabilidad.

	Tiempo de respuesta	Sobreimpulso	Tiempo de asentamiento	Error en estado estacionario	Estabilidad
Incremento de K_P (Proporcional)	Disminuye	Aumenta	Leve aumento	Disminuye	Degrada
Incremento de K_I (Integral)	Disminuye levemente	Aumenta	Aumenta	Gran disminución	Degrada
Incremento de K_D (Derivativo)	Disminuye levemente	Disminuye	Disminuye	No afecta	Mejora

Tabla 1.1 Efectos del ajuste independiente de las ganancias Kp, Ki, Kd en la respuesta en lazo cerrado

Para el ajuste manual de este controlador fue necesario reducir tanto la ganancia del integrador como la ganancia proporcional, ya que, al aumentar solamente la acción integral, el sobreimpulso y el tiempo de asentamiento aumentan, mientras que si solo se aumenta la acción proporcional, el tiempo de asentamiento aumenta un poco más y por ende el sistema se vuelve más lento. Esto indica que si no se reducen ambas ganancias la respuesta no cumplirá con lo requerido. Al disminuir la acción integral, el sobreimpulso y el tiempo de asentamiento disminuyen, y al disminuir la acción proporcional sobreimpulso también disminuye, además el error en estado estacionario no presentaría un gran cambio. De esta forma se cumplirá con lo requerido.

Después de realizar varias pruebas, finalmente las ganancias ajustadas quedaron de la siguiente manera:

$$K_p = 3.9631$$
$$K_i = 0.0322$$

```
num = 0.04;
den = [0.085 0.319 0.2816 0];

Gcs = tf(num,den);
step(Gcs)

%PI ajustado
kp = 3.9631;
ki = 0.0322;
c = pid(kp,ki,0);
T = feedback(c*Gcs,1);
t = 0:0.01:20;
step(T,t);
```

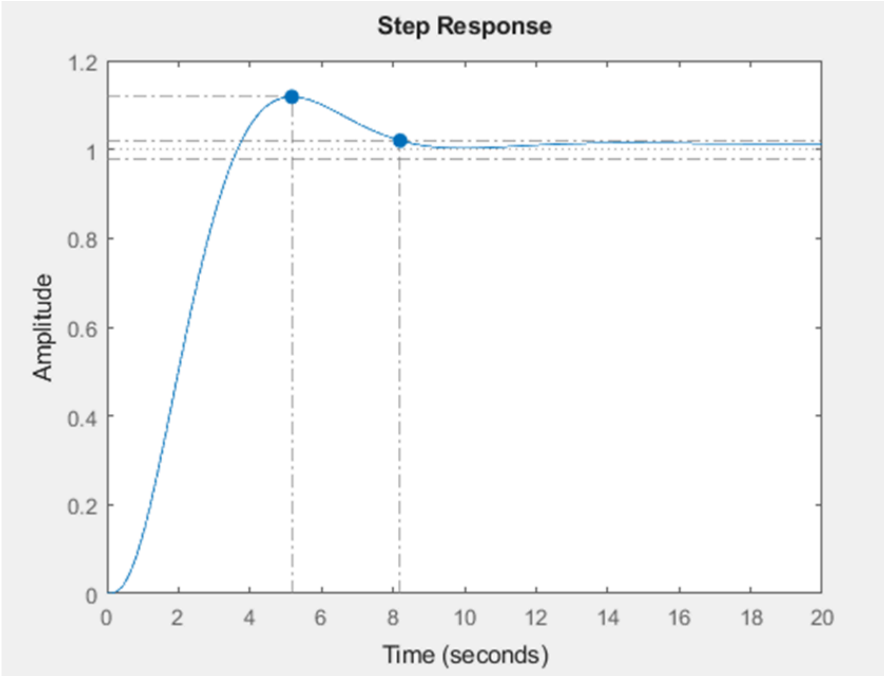


Figura 2.3 Respuesta de la planta en lazo cerrado con el controlador PI ante una entrada escalón y ganancias ajustadas

Como se puede observar en la **Figura 2.3**, nuestro sistema cumple con un error en estado estacionario menor al 2%, el tiempo de asentamiento es de 8.2 segundos, por lo que cumple con ser menor a 15 segundos, y finalmente el sobre impulso es de 11.8%, por lo que cumple con ser menor al 20%.

- Para comenzar con el PD recordemos que la función de transferencia del controlador es:

$$G_c(s) = K_p(1 + T_d s)$$

Por lo que se necesita encontrar la el valor de K_p y de K_d lo cuales se calculan a partir de la siguiente tabla:

Control	Kp	Ki	Kd
P	0,50*Kc		
PI	0,45*Kc	0,54*Kc/Tc	
PD	0,80*Kc		0,075*Kc*Tc

Por lo tanto:

$$K_p = 0.80(26.4207) = 21.1365$$

$$K_d = 0.075(26.4207)(3.4520) = 6.8403$$

Para analizar la respuesta de nuestro sistema a este controlador, lo graficamos ante una entrada escalón con el siguiente código en Matlab.

```
%% Sistema
planta = tf(0.04,[0.085 0.319  0.2816 0]);

%% Controlador PD
kp= 21.1365;
kd= 6.8403;
Gc=pid(kp,0,kd);
T= feedback(Gc*planta,1);
t= 0:0.01:25;
step(T,t)
```

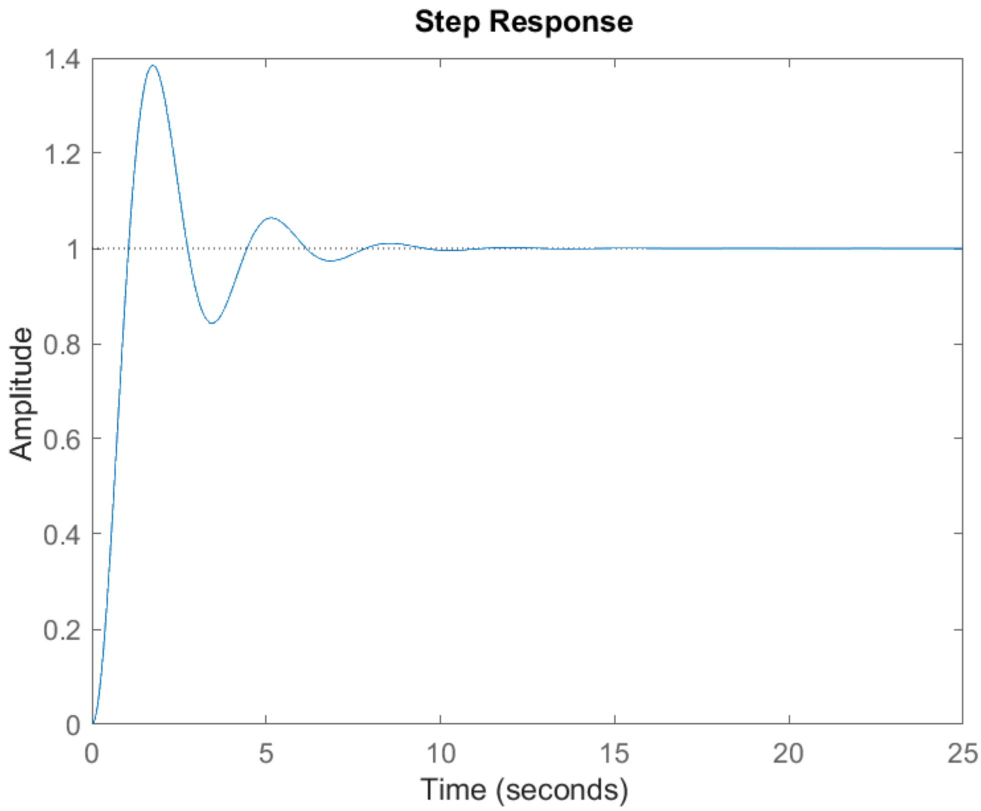


Figura 2.4 Respuesta de la planta en lazo cerrado con el controlador PD ante una entrada escalón

En la **figura 2.4**, se observa que la respuesta de la planta con el controlador PD. Cumple con el requerimiento de un error en estado estacionario menor al 2% y un tiempo de asentamiento menor a 15 segundos. Sin embargo, presenta un sobreimpulso superior al 20%. Esto podría deberse a una ganancia proporcional muy alta o una ganancia derivativa muy baja ya que recordando el derivador sirve para disminuir el sobre impulso.

Según la **tabla 1.1** en la **página 10**, el ajuste debería centrarse en aumentar la ganancia derivativa más que en reducir la ganancia proporcional. Si se disminuye demasiado la ganancia proporcional, el error en estado estacionario podría aumentar. Por lo tanto, un incremento en la ganancia derivativa es la mejor opción para reducir el sobreimpulso sin comprometer el cumplimiento de los otros requerimientos.

Para el ajuste manual de este controlador, nos centramos más en aumentar la acción derivativa, ya que si aumentamos la acción proporcional provocará que el sobreimpulso crezca más, por lo que nuestro sistema no cumplirá con lo requerido.

Después de varias pruebas, las ganancias ajustadas nos quedan de esta manera:

$$K_p = 21.1365$$

$$K_d = 20.5209$$

```
num = 0.04;
den = [0.085 0.319 0.2816 0];

Gcs = tf(num,den);
step(Gcs)

%PD ajustado
kp = 21.1365;
kd = 20.5209;
c = pid(kp,0,kd);
T = feedback(c*Gcs,1);
t = 0:0.01:20;
step(T,t);
```

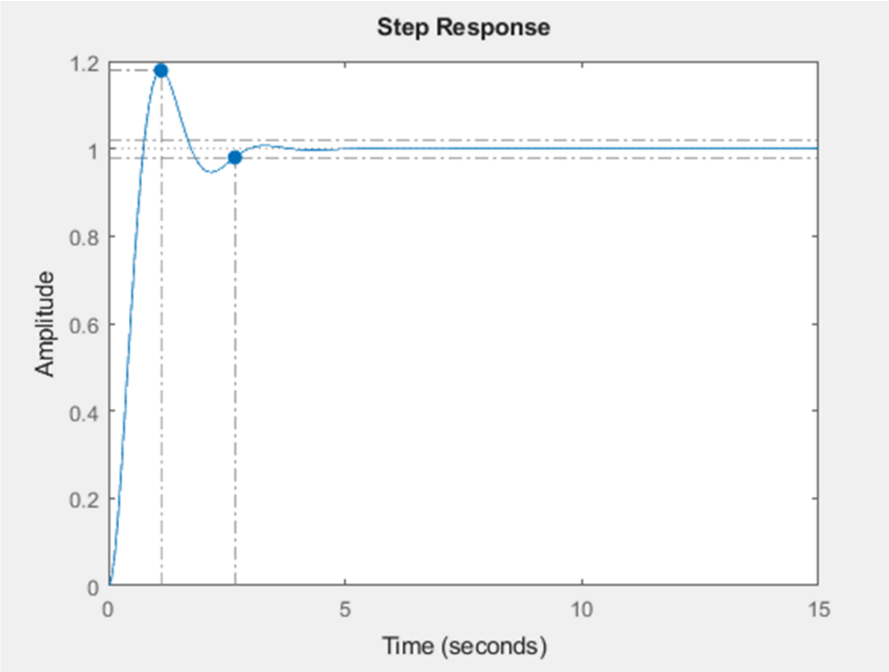


Figura 2.5 Respuesta de la planta en lazo cerrado con el controlador PD ante una entrada escalón y ganancias ajustadas

Si observamos la **Figura 2.5** el error en estado estacionario es menor al 2%, por lo tanto se cumple con este requisito, el tiempo de asentamiento es de 2.67 segundos, por lo que es menor a 15 segundos, y finalmente el sobreimpulso es de 17.9%, por lo que es menor de 20%.

- Para el controlador PID también requerimos su función de transferencia:

$$G_c(s) = K_p\left(1 + \frac{1}{T_i s} + T_d s\right)$$

Obtenemos los valores de K_p, T_i y T_d:

$$K_p = 0.6(26.4207) = 15.8524$$
$$T_i = 0.5(3.4520) = 1.726$$
$$T_d = 0.125(3.4520) = 0.4315$$

Para obtener K_i y K_d sustituimos los valores calculados en la función de transferencia:

$$15.8524\left(1 + \frac{1}{1.726s} + 0.4315s\right)$$
$$= 15.8524 + \frac{15.8524}{1.726s} + (0.4315s)(15.8524)$$
$$= 15.8524 + \frac{9.1844}{s} + 6.8403s$$

Por lo tanto:

$$K_i = 9.1844$$
$$K_d = 6.8403$$

Para analizar la respuesta de nuestro sistema a este controlador, lo graficamos ante una entrada escalón con el siguiente código en Matlab.

```
%% Sistema
planta = tf(0.04,[0.085 0.319  0.2816 0]);

%% Controlador PID
kp= 15.8524;
ki = 9.1844;
kd= 6.8403;
Gc=pid(kp,ki,kd);
T= feedback(Gc*planta,1);
t= 0:0.01:25;
step(T,t)
```

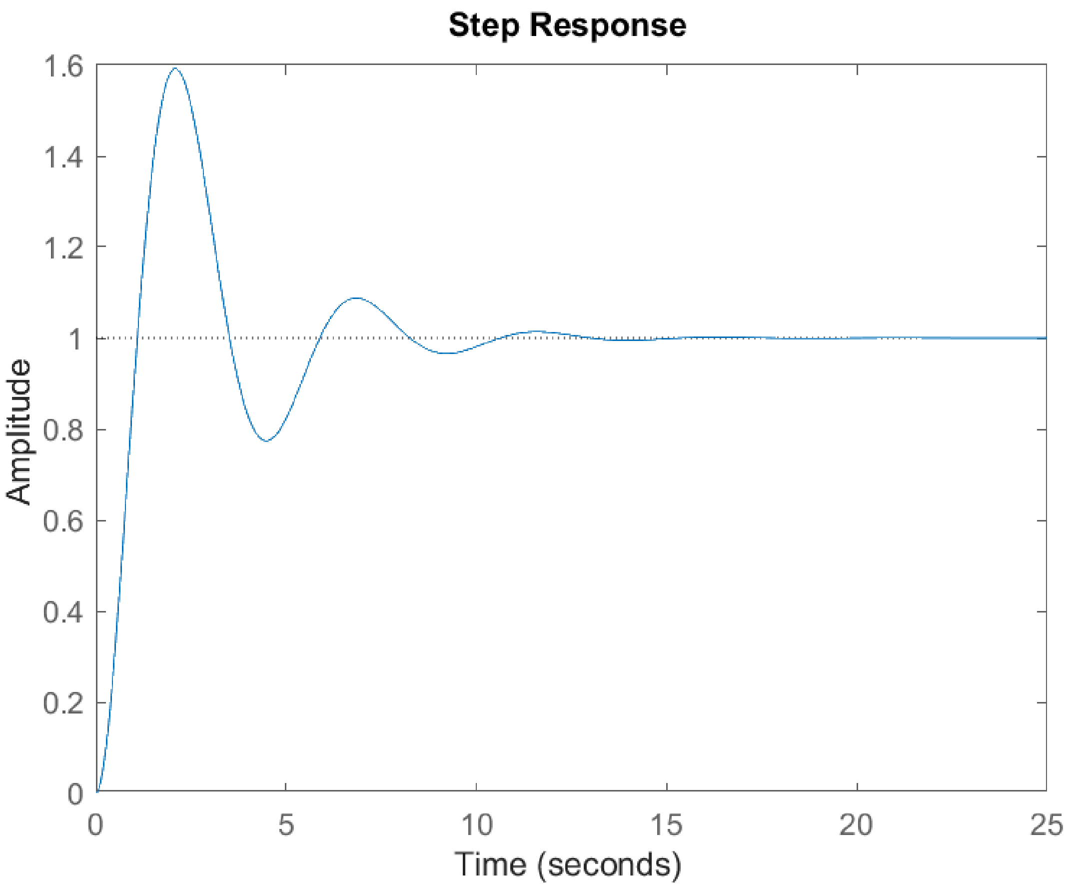


Figura 2.6 Respuesta de la planta en lazo cerrado con el controlador PID ante una entrada escalón

Se puede observar en la **figura 2.6** que la respuesta de la planta con el controlador PID si cumple con el requerimiento de un error en estado estacionario menor al 2% y un tiempo de asentamiento menor a 15 segundos. Por otro lado, el requerimiento de un sobreimpulso menor al 20% no se cumple esto puede deberse principalmente a que la ganancia del integrador es muy alta por lo que crea más sobreimpulso o también la ganancia proporcional es elevada por lo que hace que haya más sobreimpulso, otro motivo puede ser que la ganancia del derivador es muy baja por lo que su efecto para reducir el sobreimpulso no es suficiente para cumplir el requerimiento.

Consultado la **tabla 1.1** en la **página 10**, el ajuste debería enfocarse principalmente en incrementar la ganancia derivativa y ajustar a la proporcional ya que el integrador es más efectivo para reducir el error en estado estacionario que la ganancia proporcional.

Para el ajuste manual de este controlador fue necesario reducir la ganancia integral y la ganancia proporcional, y aumentar la ganancia derivativa, ya que al reducir solamente estas dos ganancias, la acción derivativa seguiría siendo baja y por ende el sobreimpulso no va a disminuir significativamente, por lo que nuestro sistema no cumplira con los requerimientos. Al aumentar la acción derivativa, se reduce el tiempo de asentamiento y el sobreimpulso, así cumpliremos con lo requerido.

Después de varias pruebas, las ganancias ajustadas son las siguientes:

$$K_p = 13.2103$$

$$K_i = 4.5922$$

$$K_d = 13.6806$$

```
num = 0.04;
den = [0.085 0.319 0.2816 0];

Gcs = tf(num,den);
step(Gcs)

%PID ajustado
kp = 13.2103;
ki = 4.5922;
kd = 13.6806;
c = pid(kp,ki,kd);
T = feedback(c*Gcs,1);
t = 0:0.01:15;
step(T,t);
```

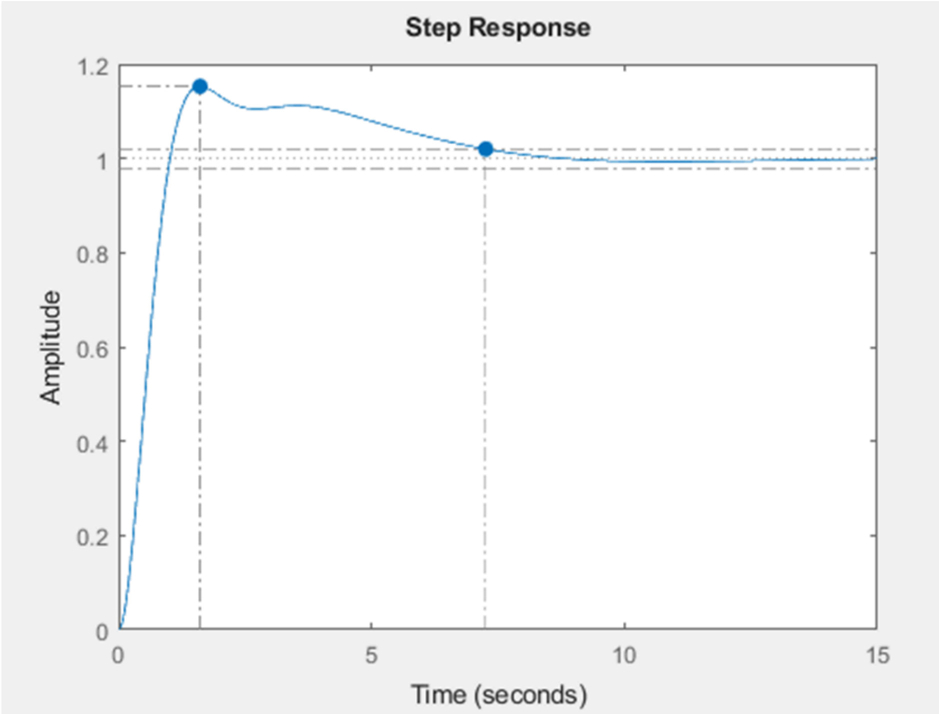


Figura 2.7 Respuesta de la planta en lazo cerrado con el controlador PID ante una entrada escalón y ganancias ajustadas

Se puede observar en la **Figura 2.7** que el sistema cumple con el error en estado estacionario menor al 2%, el tiempo de asentamiento es de 7.26 segundos, por lo tanto cumple con ser menor a 15 segundo, y finalmente el sobreimpulso es de 15.3%, por lo que cumple con ser menor al 20%.

3.- Diseñe e implemente un PID para la articulación 1 del robot Gen3-lite de la compañía Kinova, para ello use la función de transferencia (1). Pruebe el controlador llevando la articulación 1 a las siguientes 3 posiciones: ref1=30°, ref2=50° y ref3=70°. Grafique y comente los resultados. Si el sobreimpulso es más del 5% y el tiempo de asentamiento es de más de 2 segundos, mencione que cambios debe hacer al controlador para mejorar el comportamiento del sistema. Para la sintonización use el método de Ziegler-Nichols que se requiera.

$$G(s) = \frac{357.5}{s^2 + 1.094s + 0.4379} \qquad (1)$$

Para saber qué método utilizar en para realizar una sintonización adecuada podemos aplicar el criterio que nos menciona en el **capítulo 8**, **apartado 8.2** del libro de Ogata, la cual nos dice que si nuestra función de transferencia a la cual nosotros queremos sintonizar podemos aplicar el primer método **NO CONTENGA INTEGRADORES ni POLOS DOMINANTES COMPLEJOS CONJUGADOS** este método se puede aplicar. Y como podemos observar en nuestra función de transferencia de la planta, cumple con el primer criterio entonces este método si se puede aplicar.

Lo siguiente es ver la respuesta en lazo abierto ante una entrada escalón como se muestra en la **FIGURA 3.1**

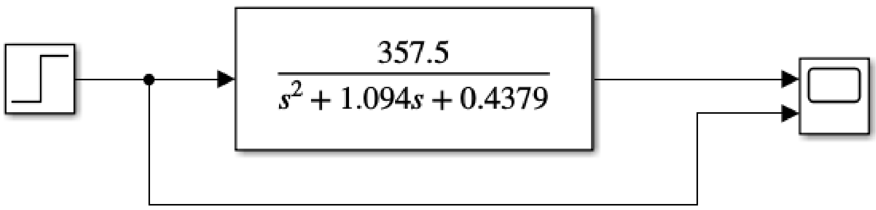


Figura 3.1 Diagrama de bloques del sistema en lazo abierto ante una entrada tipo escalón unitario.

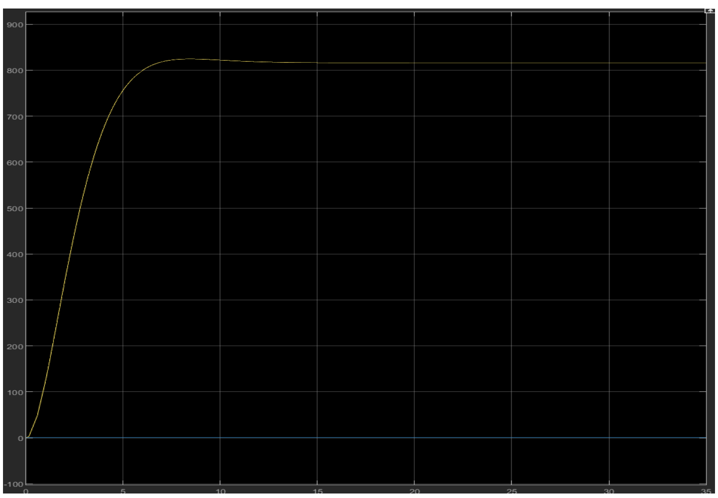


Figura 3.2 Respuesta de nuestro sistema ante una entrada escalón unitario.

Como se observa ahora en la **FIGURA 3.2** la respuesta ante una entrada tipo escalón unitario tiene una forma sigmoidal la cual nos indica que este método sí es aplicable.

Lo siguiente es encontrar 3 parámetros que nos indican el moteado, la cual las llamamos con las variables **T, L y K.** Como se muestra en la **Figura 3.3.**

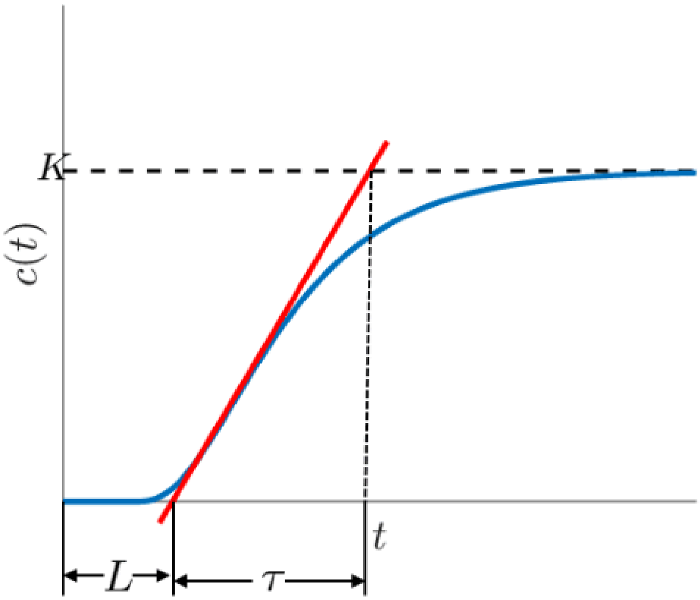


Figura 3.3. Gráfica para hallar los valores K, L y T

El único reto para este método es hallar esa pendiente tangente que pasa por el punto de inflexión de nuestra gráfica, Pero para este caso utilizaremos el siguiente código elaborado en MATLAB que se muestra en la **Figura 3.4**

```
clc
clear all
close all
%Modelo del sistema
G=tf([357.5],[1 1.094 0.4379]);
G.iodelay=0;
%tiempo
dt=0.01;
t=0:dt:35;
%Entrada
u(1:length(t))=1;
%Salida
y=lsim(G,u,t);
dy=diff(y)/dt; %Derivada
dy2=diff(dy)/dt;
%Encontrar el punto de inflexion y su derivada
% el punto donde la pendiente de la respuesta escalón tiene su valor máximo (punto de inflexión)
[m,p]=max(dy);
yp=y(p);
tp=t(p);
tm=0:20;
ym=m*(tm-tp)+yp; %Ecuacion de la recta
figure
plot(t,y,tp,yp,'o',tm,ym,'-r','linewidth',3);
axis([0 35 0 1000]);
box off
ylabel('$$c(t)$$','FontSize',20,'Interpreter','latex')
xlabel('$$t$$','FontSize',20,'Interpreter','latex')
set(gca,'FontSize',(20) )
```

Figura 3.4. Código elaborado en matlab que nos ayuda a graficar esa recta tangente que pasa por el punto de inflexión.

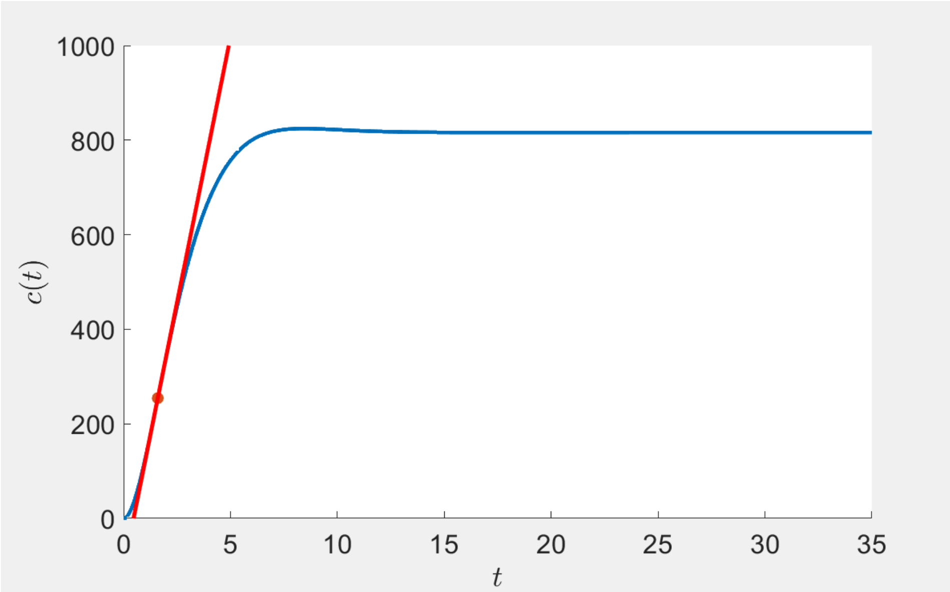


Figura 3.5. Gráfica de la recta tangente que pasa por el punto de inflexión de nuestro sistema.

Con esto ya podemos aplicar el criterio de cómo obtener los valores de K, L y T que se muestran en la **Figura 3.3.** Para esto utilizamos la herramienta en Matlab para colocar puntos sobre la línea de nuestra gráfica y así sacar dichos valores.

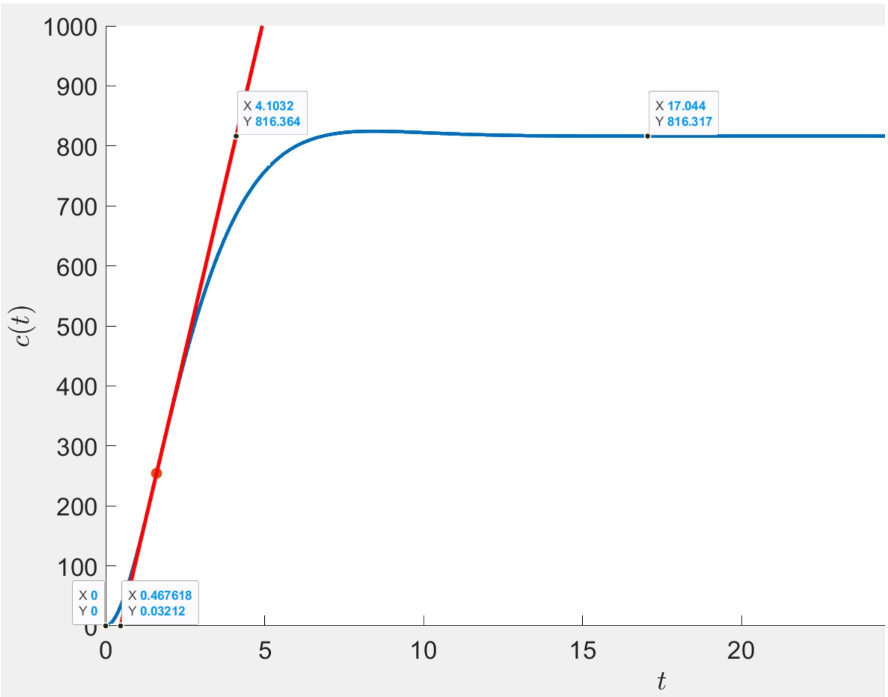


Figura 3.6. Gráfica de la recta tangente que pasa por el punto de inflexión de nuestro sistema y mostrando los puntos importantes para sacar nuestros valores de K, L y T.

Como se muestra en la **Figura 3.6** la información que nos da esta gráfica es de acuerdo a la gráfica que nos muestra el método y los parámetros que tenemos que rescatar, Para esto decimos que el valor de **K** es aquella amplitud donde el sistema se mantiene estable, en este caso nos da un valor de **816.3**. Después para sacar nuestro valor **L** tenemos que ver la distancia donde nuestro sistema comienza, en este caso es en la coordenada o en el punto (cero, cero) o el tiempo que tardó o en este caso en el eje de la abscisa hasta tocar nuestra pendiente tangente (linea de color rojo), y se observa que el tiempo o el valor de **L** es de **0.467.** Lo siguiente es determinar nuestro valor de T para esto tomamos el punto donde nuestra pendiente toca el valor de K la cual se muestra también en nuestra **figura 3.6** y retrocedemos hasta llegar a nuestro punto donde medimos **L** y notamos que es la diferencia de nuestro punto en **4.103** s y **0.467.**

Con esto ya contamos con nuestros valores de K y L, los cuales son los únicos que necesitamos para calcular nuestras ganancias.

$$L = 0.467$$
$$T = 4.103 - 0.467 = 3.636$$

Lo siguiente es calcular el valor de Kp TI y Td a partir de la tabla que se muestra en la **Tabla 1.2**.

Tipo de controlador	K_p	T_i	T_d
P	$\frac{T}{L}$	∞	0
PI	$0.9\frac{T}{L}$	$\frac{L}{0.3}$	0
PD	$1.2\frac{T}{L}$	$2L$	$0.5L$

Tabla 1.2 Tabla que se utilizara para sacar los valores de Kp, Ti y Td

Utilizando la tabla obtenemos que:

$$K_p = 1.2(3.636/0.467) = 9.343$$
$$T_i = 2(0.467) = 0.934$$

$$T_d = 0.5(0.467) = 0.2335$$

Y el último paso para este método de sintonización es calcular K_d, la cual se obtiene por el producto de K_p y T_d , y para obtener K_i solo es el cociente de K_p y T_i. como se muestra a continuación.

$$K_i = 9.343 / 0.934 = 10.0032$$
$$K_d = 9.343 * 0.2335 = 2.1816$$

Con esto dicho es ahora de probar nuestra ganancias. Para esto utilizamos el siguiente diagrama en Simulink que se ve en la **figura 3.7.**

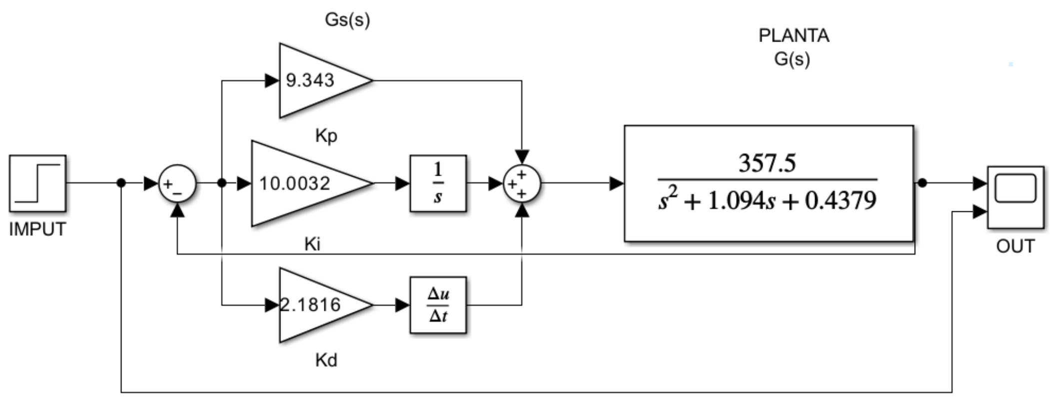


Figura 3.7. Diagrama de bloques de un sistema G(s) compensado por un PID Gc(s).

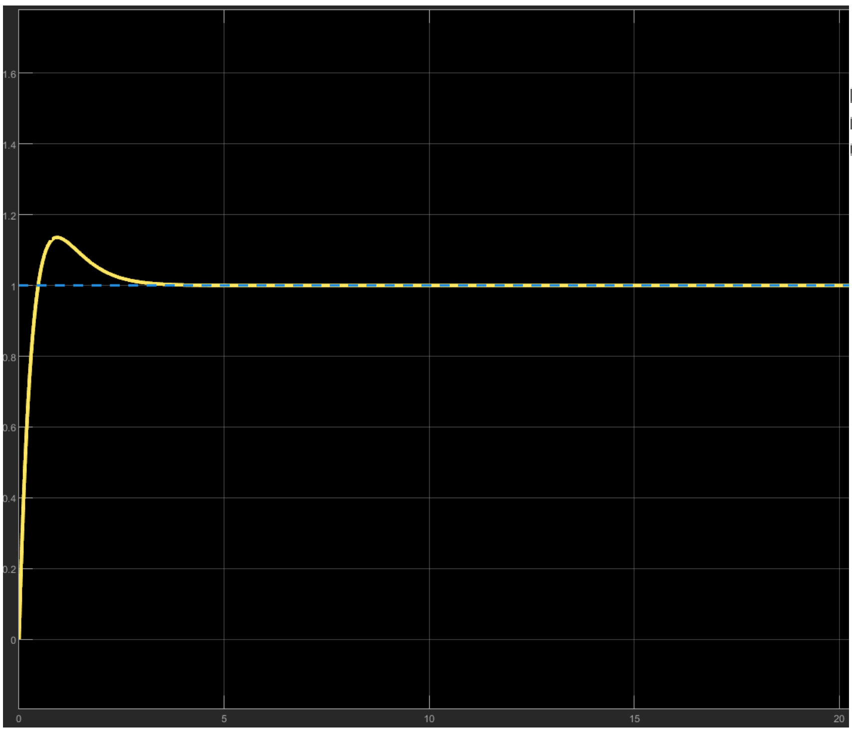


Figura 3.8. Gráfica de la respuesta de nuestro sistema G(s) compensado por un PID Gc(s)

Con esto concluimos este método de sintonización para el controlador PID, Notamos que en comparación al anterior método este método no se puede resolver analíticamente, Lo cual recurrimos a lo visual, a la obtención de la gráfica de la respuesta ante una entrada escalón unitario y de ahí sacamos los parámetros, cosa que no pasó con el anterior caso o en el método dos presentado por **Ziegler-Nichols**

Notamos que en la respuesta mostrada en la **figura 3.8** tenemos a un un sobre impulso de alrededor del 20% esto claramente no cumple con las especificaciones requeridas en el inciso también el tiempo de asentamiento ronda entre los 3 a 4 segundos lo cual también no cumple con las especificaciones para esto decimos que tenemos que ajustar o tunear las ganancias para obtener la respuesta requerida, para esto utilizamos la **tabla 1.1** de la **página 10**.

Con esto el primer ajuste es disminuir ese sobre impulso, para esto reducimos la acción proporcional a la mitad, y en este caso como la acción proporcional es proporcional a las ganancias Ki y Kd, porque al momento de calcularlas fue el producto de su tiempo y la ganancia Kp, entonces bajamos también a la mitad dichas ganancias. Y los resultados se muestran en la **Figura 3.9**

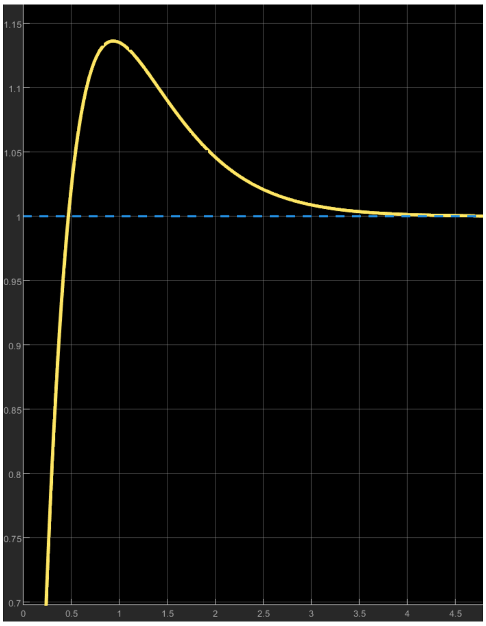


Figura 3.9. Respuesta ante una entrada tipo escalón unitario a nuestro sistema G(s) compensado por un PID Gc(s) con las ganancia Kp ajustada a la mitad.

En la **Figura 3.9** se observa que logramos bajar ese sobre impulso a uno más pequeño, pero aún no es suficiente, así que aplicando la metodología de sintonización manual que se muestra en la **Figura 3.8** para tener una sintonización más fina quedaría de la siguiente manera

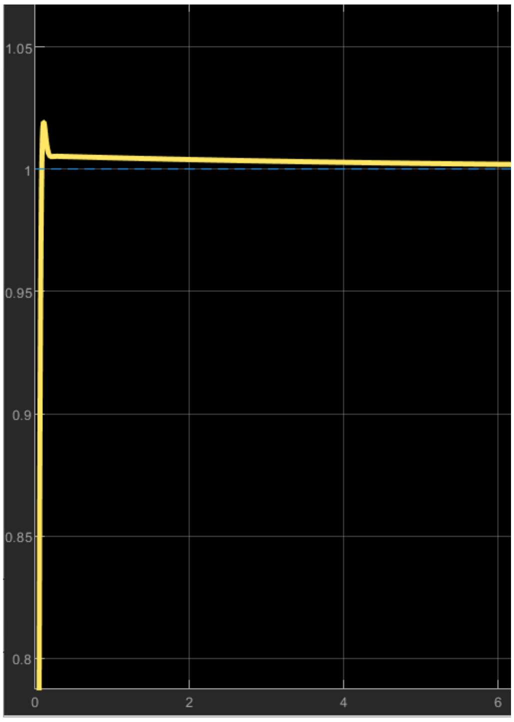


Figura 3.10. Respuesta ante una entrada tipo escalón unitario a nuestro sistema G(s) compensado por un PID Gc(s) con las ganancias ajustadas finamente.

Como observamos en la **Figura 3.10**, Conseguimos un sobre impulso del al menos 2 por ciento, y cumplimos con el tiempo de asentamiento, esto se debió al ajuste fino elaborado con la anterior tabla de la **Figura 3.8**.
Y con estas ganancias fue que obtuvimos nuestra respuesta satisfactoria.

$$K_i = 5.9215$$
$$K_d = 1.0016$$
$$K_p = 0.1908$$

Bien ahora con nuestras ganancias pasamos a la práctica donde aplicamos el controlador a la planta física la cual decimos que es para la articulación 1 del robot Gen3-lite de la compañía Kinova, la cual no contábamos con que para el sistema físico estaba limitado a cierto rango de valores para las ganancias. en este caso solo la ganancia Kp estaba fuera del rango, así que solo tuvimos que ajustar esa ganancia, y los resultados fueron los siguientes.

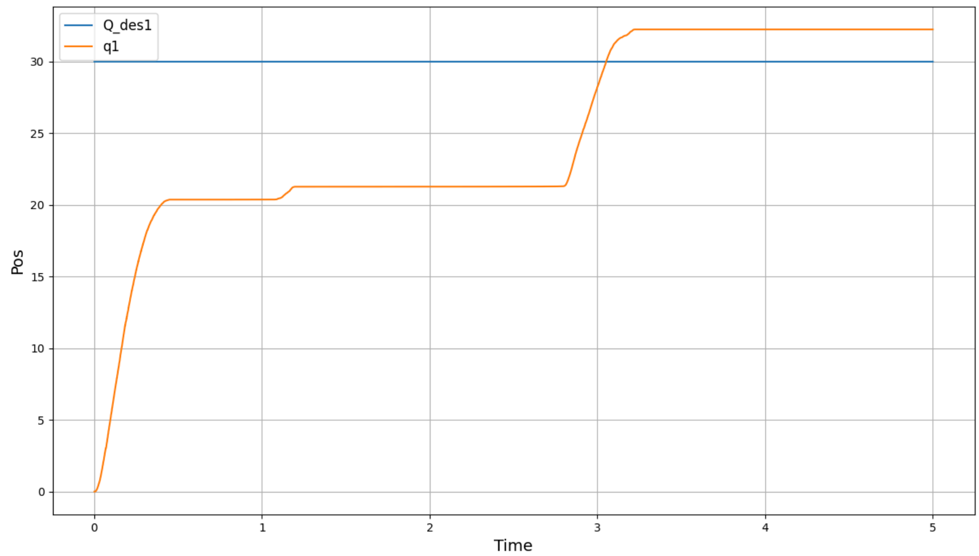


Figura 3.11. Respuesta ante una señal de entrada de 30 grados al sistema G(s) física compensada por un PID Gc(s)

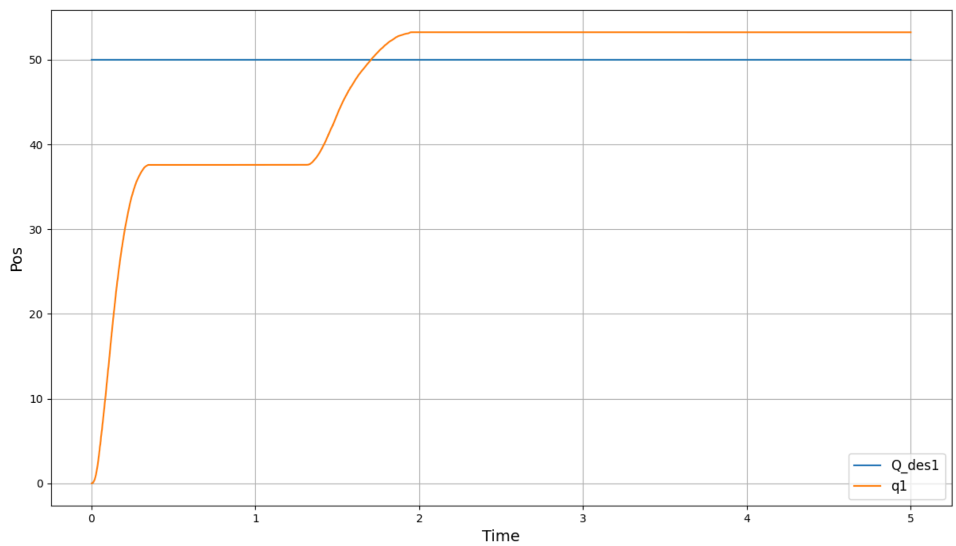


Figura 3.12. Respuesta ante una señal de entrada de 50 grados al sistema G(s) física compensada por un PID Gc(s)

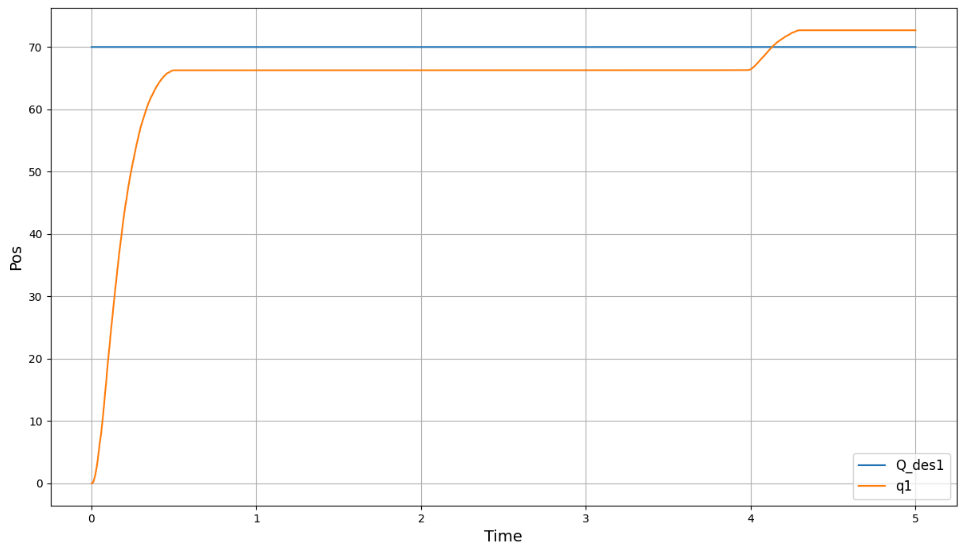


Figura 3.13. Respuesta ante una señal de entrada de 70 grados al sistema G(s) física compensada por un PID Gc(s)

Lo que se muestra en la **Figura 3.11 , 3.12 , y 3.13** son las respuesta ante una entrada de referencia en grados, la cual se nos especifica en el inciso, y lo que podemos rescatar de estas respuestas es que tenemos buenos resultados con muy poco sobre impulsos pero siempre tendremos error en estado estacionario, esto se debe principalmente a que nuestras ganancias calculadas se hicieron en una simulación ideal la cual esto en la vida real no pasa, y estas diferencias no se tomaron en cuenta a la hora de calcular nuestras ganancias.

Bibliografía:

PID Tuner - Ajustar controladores PID - MATLAB - MathWorks América Latina. (s. f.). https://la.mathworks.com/help/control/ref/pidtuner-app.html