

Sistemas de Control en robótica.

Sintonización de Controladores PID

**Compilado por
Jhon Sossa & Johan Sanabria**

Profesor: Mauricio Arias Correa

**Instituto Tecnológico Metropolitano.
Antioquia.**

Identificación del sistema:

Se obtuvieron los datos de una planta de velocidad para un motor DC. Los resultados se presentan en la figura 1:

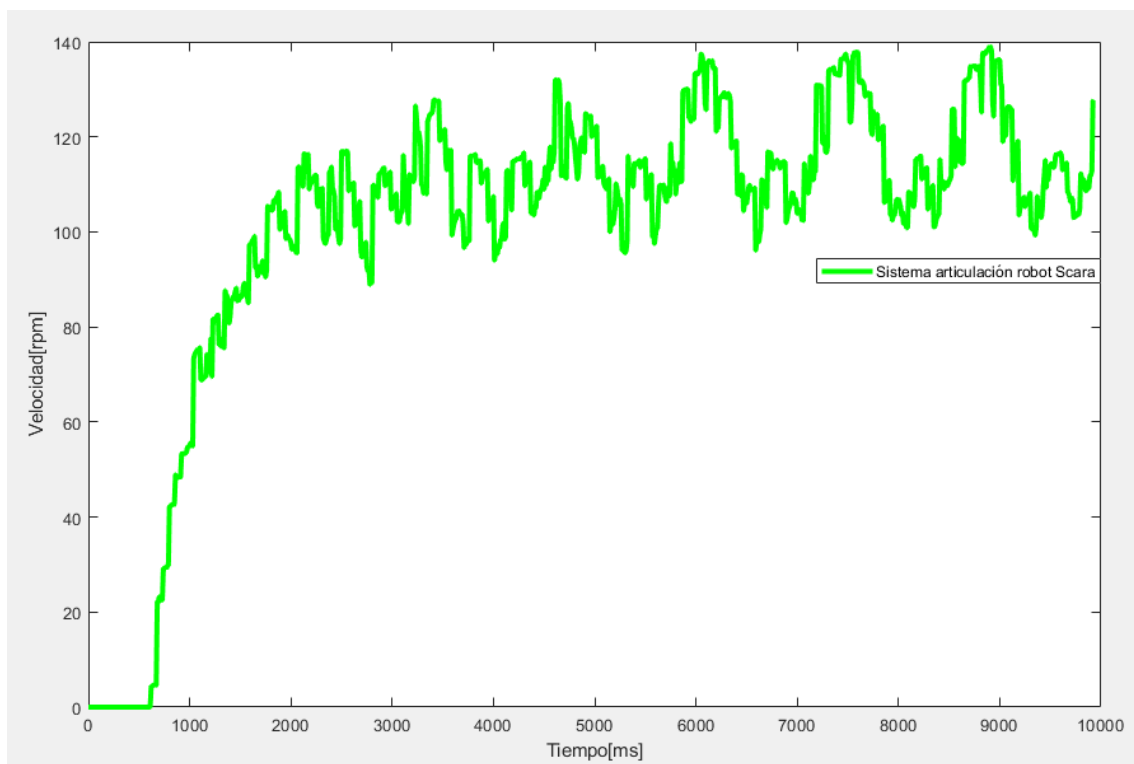


Figura 1. Gráfica del comportamiento del sistema.

Ahora se obtendrán los sistemas POR y SOR

Sistema POR:

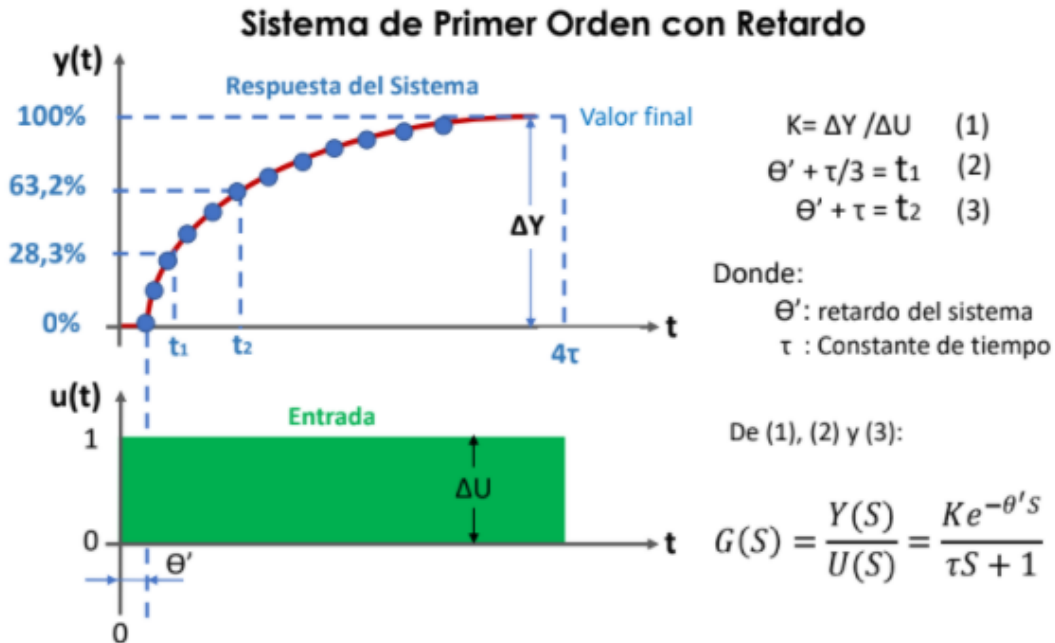


Figura 2. Cálculo de un sistema de primer orden con retardo.

Sistema POR	Resultados
$t_1 \rightarrow 28.3\%(interpolación)$	0.798 s
$t_2 \rightarrow 63.2\%(interpolación)$	1.174 s
$\tau(constante\ de\ tiempo)$	0.564 s
$\theta'(tiempo\ de\ retardo)$	0.610 s
ΔY	114.17724 rpm
ΔU	180 rpm
$K(Ganancia\ del\ sistema)$	0.63432
Perdidas del sistema	36.568 %
$T_s(tiempo\ de\ muestreo)$	0.010
$4\tau(punto\ de\ estabilización)$	2.256 s
$G(S) \rightarrow Función\ de\ transferencia$	$\frac{0.63432e^{-0.610S}}{0.564S + 1}$

Tabla 1. Sistema POR

Sistema de SOR:

$$G_p(S) = \frac{K e^{-\theta' s}}{(\tau_1 S + 1)(\tau_2 S + 1)} \rightarrow \xi \geq 1$$

$$\tau_{1,2} = \frac{\xi \pm \sqrt{\xi^2 - 1}}{W_n} \quad K = \frac{\Delta Y}{\Delta U}$$

$$x = \frac{t_2 - t_1}{t_3 - t_1} \rightarrow \xi = \frac{0.0805 - 5.547(0.475 - x)^2}{x - 0.356}$$

$$F_2(\xi) = 2.6\xi - 0.6 \rightarrow W_n = \frac{F_2(\xi)}{t_3 - t_1}$$

K = Ganancia de la planta

W_n = Frecuencia natural

ξ = Coeficiente de amortiguamiento

θ' = Tiempo muerto de la planta

τ_1 y τ_2 = Constantes de tiempo

ΔY = Cambio total en la salida de la planta

ΔU = Magnitud del escalon aplicado

t_1 = tiempo requerido para que la salida alcance el 15% de su valor final

t_2 = tiempo requerido para que la salida alcance el 45% de su valor final

t_3 = tiempo requerido para que la salida alcance el 75% de su valor final

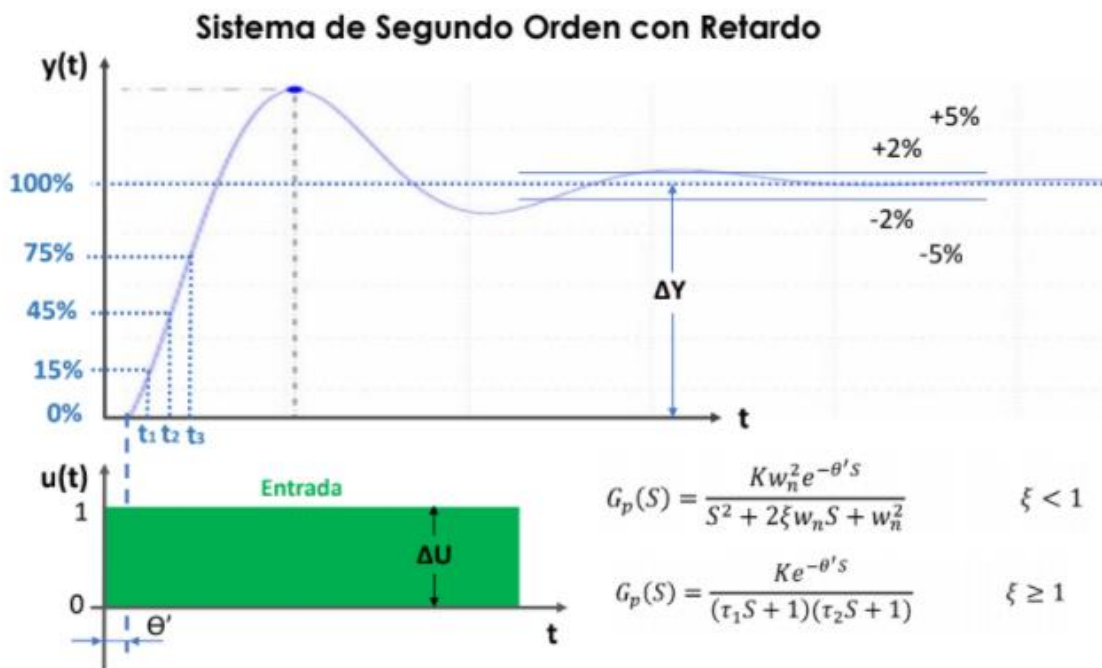


Figura 3. Cálculo de un sistema de segundo orden con retardo.

Sistema POR	Resultados
$t_1 \rightarrow 15\%(interpolación)$	0.679 s
$t_2 \rightarrow 45\%(interpolación)$	1.034 s
$t_3 \rightarrow 45\%(interpolación)$	1.769 s
$\tau_1(constante\ de\ tiempo)$	0.856 s
$\tau_2(constante\ de\ tiempo)$	0.144 s
$\theta'(tiempo\ de\ retardo)$	0.610 s
ΔY	114.93121 rpm
ΔU	180 rpm
$K(Ganancia\ del\ sistema)$	0.63851
Perdidas del sistema	36.149 %
$T_s(tiempo\ de\ muestreo)$	0.010
$\xi(Constante\ amortiguamiento)$	1.424
$F_2(\xi) \rightarrow \xi \geq 1$	3.1024
$W_n(frecuencia\ natural)$	2.84624 Hz
$G(S) \rightarrow Función\ de\ transferencia$	$\frac{0.63851e^{-0.610S}}{(0.856S + 1)(0.144S + 1)}$

Tabla 2. Sistema SOR

Sintonización del controlador PID en Matlab

Procedimiento:

1. Se cargan los datos obtenidos del sistema a partir de una base de datos:

```

tiempo = xlsread('DatosE.xlsx','A2:A995');
velocidad = xlsread('DatosE.xlsx','B2:B995');
t=tiempo; y=velocidad;
datos = iddata(y,t,0.010)
datos=iddata(y, t, 10) %muestreo cada 10mS

```

2. Luego en el command windows de Matlab se digita pidTuner para abrir la aplicación de PID Tuner donde identificaremos la planta y luego sintonizaremos los parámetros del controlador PID.

```
%cargamos pidTuner  
pidTuner
```

3. Identificamos la planta.

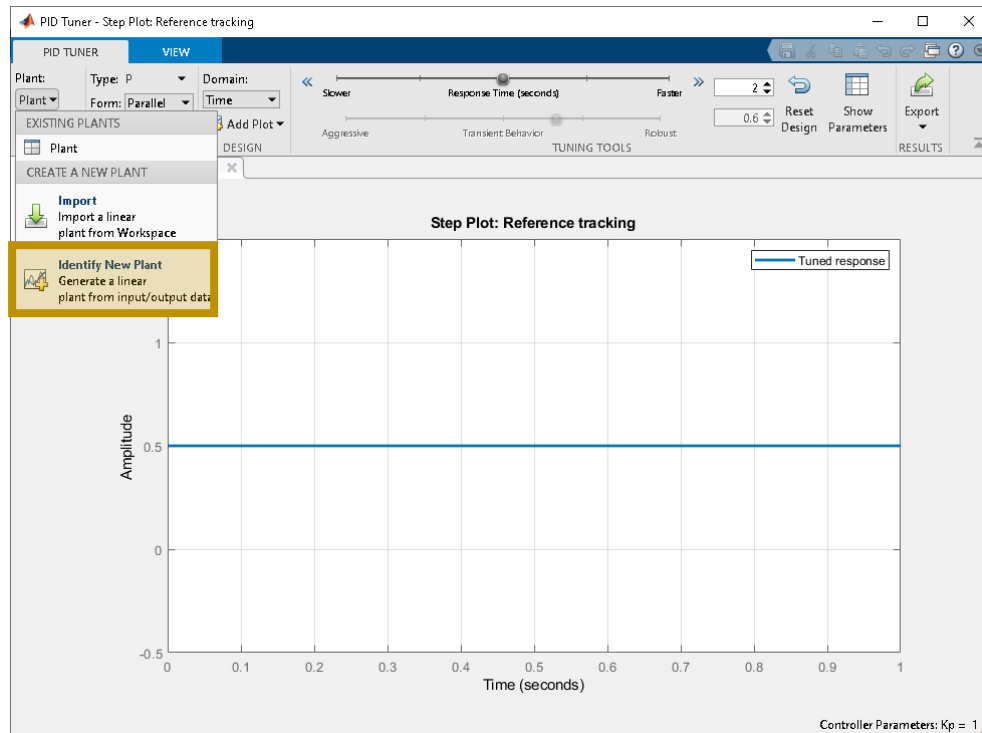


Figura 4. Identificación de la nueva planta.

4. Agregar datos

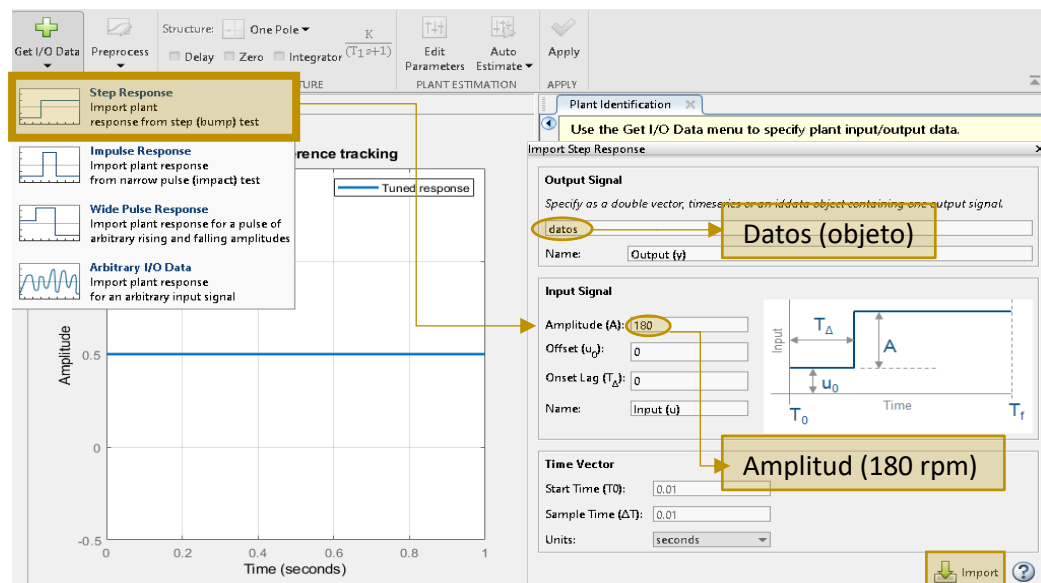


Figura5. Identificación de la planta

5. Se analiza el sistema de primer orden

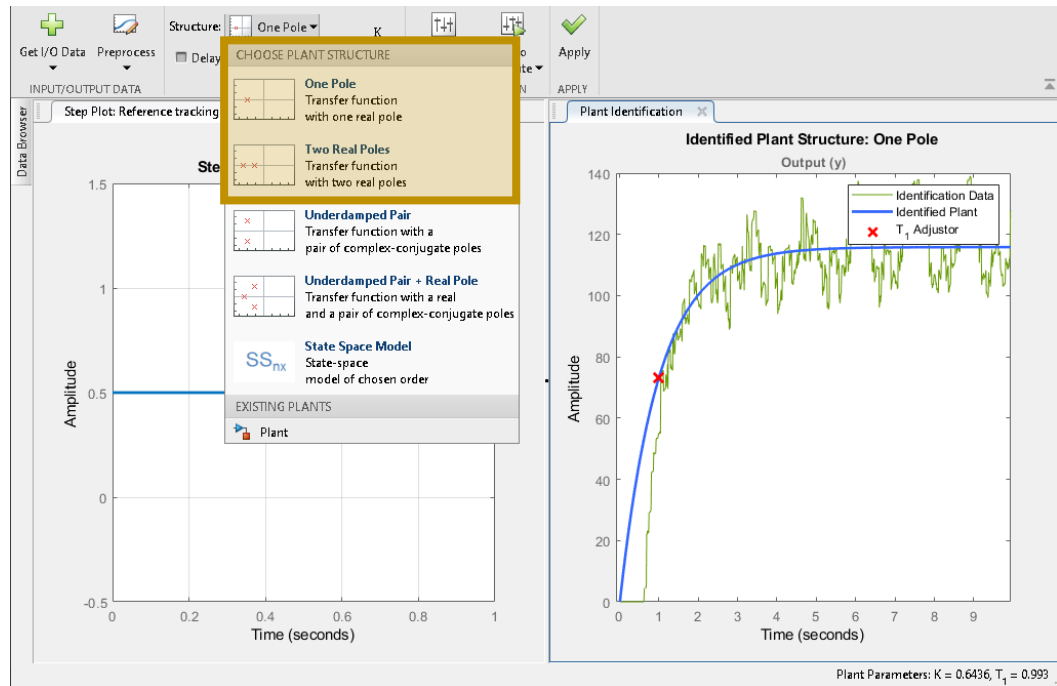


Figura 6. Analizar la planta como sistema POR o sistema SOR

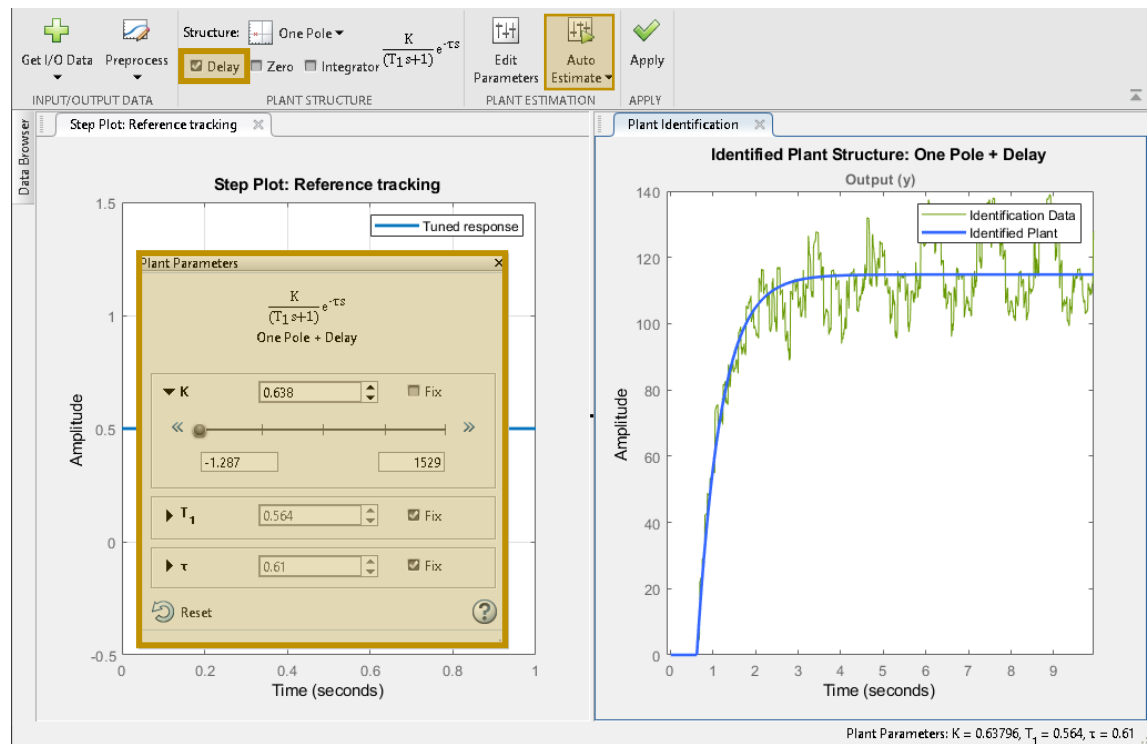


Figura 8. Configuración del sistema POR en la planta

6. Se analiza el sistema de segundo orden

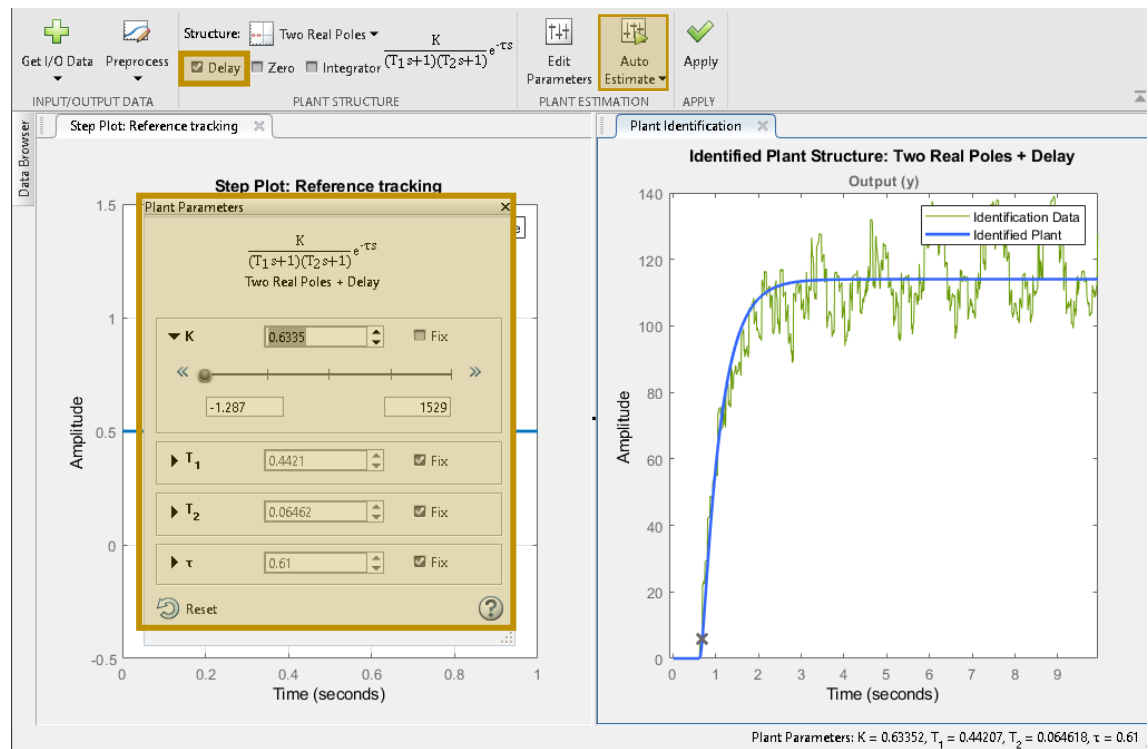


Figura 9. Configuración del sistema SOR en la planta

7. Ahora luego de haber configurado el sistema que identifica la planta vamos a seleccionar la opción Apply en PID Tuner.

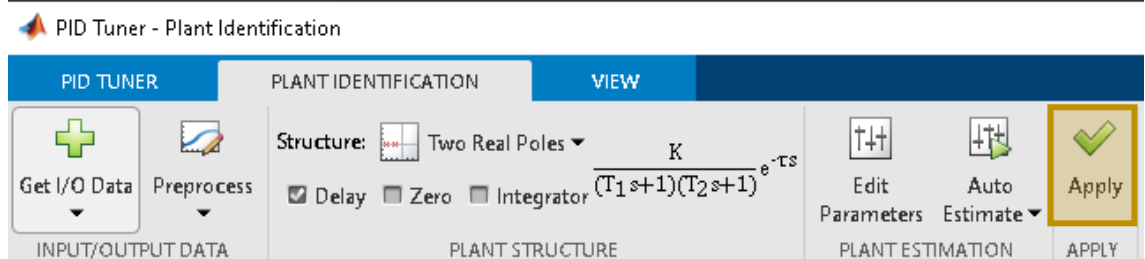


Figura 10. Aplicar la planta al sistema de datos para sintonizar un PID

8. Por último, vamos a sintonizar nuestro PID para obtener los parámetros óptimos para estabilizar la planta en un Setpoint deseado, la gráfica de la izquierda nos muestra el comportamiento del controlador PID, en la gráfica de la derecha vamos a observar la planta, también si seleccionamos la opción Show Parameters observaremos los parámetros K_p , K_i y K_d del controlador los cuales podemos ir variando con la ayuda de los deslizadores que se observan en la parte superior de la imagen, en los cuales estamos manipulando la velocidad de respuesta del controlador y la robustez del mismo, por tanto, podemos ir variando y jugando con cada uno de estos

parámetros hasta obtener el controlador óptimo que me garantice que mi planta en este caso la articulación alcance en un tiempo una velocidad determinada que será el Setpoint.

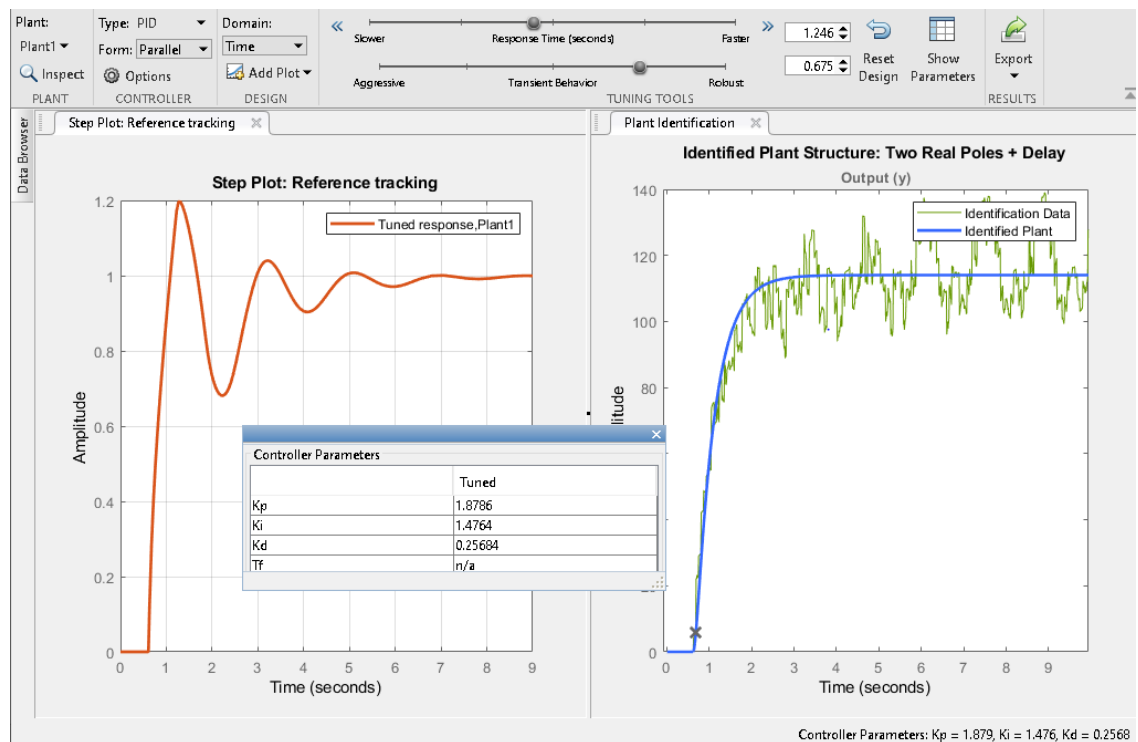


Figura 11. Sintonización de los parámetros del PID

Bibliografía:

Cómo implementar un controlador PID en Arduino. (2020). Retrieved 1 December 2020, from <https://www.luisllamas.es/como-implementar-un-controlador-pid-en-arduino/>

(2020). Retrieved 1 December 2020, from https://www.youtube.com/watch?v=ts-4jYyuQ-8&list=PL6aTr-0DUFPmMl1ro_bQ909dCbWDn0JxP

(2020). Retrieved 1 December 2020, from https://www.youtube.com/watch?v=cahJBH60_P0&list=PL6aTr-0DUFPmMl1ro_bQ909dCbWDn0JxP

Diagrama de flujo. (2020). Retrieved 1 December 2020, from <https://lucid.app/documents#/documents>

Fernandez, F. (2020). Retrieved 1 December 2020, from https://www.researchgate.net/publication/290816594_Uso_y_determinacio

n de los parametros de un controlador PID mediante el metodos de
Ensayo y Error Ziegler-Nichols y Cohen-Coon