UAB
Universitat Autònoma de Barcelona

escola d'enginyeria

Control de Sistemas

Práctica 5 - Diseño de Controladores PID

Objetivo

El objetivo de ésta práctica consiste en aprender a diseñar y sintonizar diferentes controladores PID. Para ello se utilizarán diferentes técnicas de sintonización, desde las más sencillas hasta las que nos permiten diseñar el controlador PID teniendo en cuenta la estabilidad deseada. Para simplificar la práctica y el diseño en sí, se otorgará un código esqueleto de Matlab (un archivo .m) el cual se deberá rellenar correctamente para poder llevar a cabo el diseño de los controladores. Por otro lado se proporcionará un conjunto de archivos Simulink los cuales se deberán acabar de completar. **Importante: Respetad el nombre de las variables, en caso contrario no se puede asegurar el buen funcionamiento del script**.

1. Introducción

El controlador Proporcional, Integral Derivativo (PID) es el más utilizado en el ámbito del control de procesos industriales. A pesar de que fue introducido hace más de 70 años, aún sigue siendo objeto de un estudio intenso en el área académica. El lazo de control realimentado más simple vendría dado por:

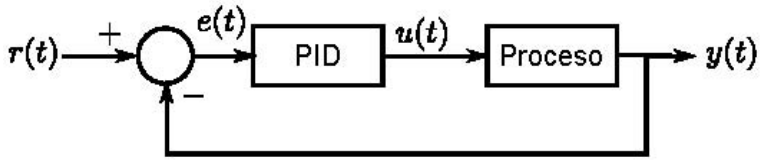

Figura 1: Sistema de control por realimentación

En términos generales, el controlador PID tiene le siguiente algoritmo de control:

$$u(t) = K_p(e(t) + \frac{1}{T_i}\int_0^t e(t)dt + T_d\frac{de(t)}{dt}) \quad (1)$$

donde

- $e(t)$ es la señal de error, es decir, la diferencia entre la salida medida de la planta y el valor deseado $e(t) = r(t) - y(t)$
- $u(t)$ es la señal del controlador y la entrada de la planta
- K_p, T_i y T_d son los parámetros del controlador. Ganancia proporcional, tiempo integral y tiempo derivativo respectivamente.

Al proceso de seleccionar los parámetros K_p, T_i y T_d del controlador se le denomina sintonización. El desarrollo de métodos de sintonización ha sido extenso. En esta práctica se estudiarán 3 de estos métodos. Para su aplicación tomaremos como marco de trabajo tanto el sistema no-lineal como el modelo lineal utilizados en prácticas anteriores.

2. Métodos de sintonización sin tener en cuenta la estabilidad

Los métodos de sintonía que se presentarán a continuación se tratan de métodos que no tienen en cuenta la estabilidad del sistema de control. En total vamos a analizar el funcionamiento de tres métodos diferentes:

- Ziegler-Nichols
- López
- Rovira

Todos ellos se basan en una descripción de la curva de reacción del proceso $Pn(s)$. Considerando que el proceso puede ser identificado por un modelo de primer orden más tiempo muerto dado por:

$$Pn(s) = \frac{k \cdot e^{-Ls}}{\tau s + 1} \quad (2)$$

donde k es la ganancia del proceso, τ es la constante de tiempo y L es el retardo del proceso. En el caso de los métodos comentados previamente veremos que se corresponden

con reglas sencillas que nos relacionan directamente los parámetros del modelo definido en (2) y los parámetros del controlador (1).

2.1. Método de Ziegler-Nichols

Los parámetros del controlador PID utilizando la regla de sintonía de Ziegler-Nichols se pueden encontrar siguiendo la siguiente tabla:

Controlador	K_p	T_i	T_d
P	$\frac{\tau}{kL}$	∞	0
PI	$\frac{0{,}9\tau}{kL}$	$\frac{L}{0{,}3}$	0
PID	$\frac{1{,}2\tau}{kL}$	$2L$	$0{,}5L$

Tabla 1: Parámetros del método Ziegler-Nichols

2.2. Métodos Óptimos Integrales (López y Rovira)

Estos criterios se basan en determinar los parámetros del controlador PID de manera que la función de error en lazo cerrado haga que un determinado criterio de optimalidad sea mínimo. Los más usuales son:

$$IAE = \int_0^\infty |e(t)|dt \tag{3}$$

$$ITAE = \int_0^\infty |te(t)|dt \tag{4}$$

Para estos criterios de sintonía, los valores óptimos de los parámetros acostumbran a proporcionarse de manera separa para el comportamiento en **servo (queremos que la salida sea igual a una entrada determinada) y en regulación (queremos que las perturbaciones que entran en nuestro sistema afecten lo mínimo posible)**. Así, para el criterio ITAE los parámetros de un controlador PI se calculan en base a aplicar una fórmula del tipo:

$$Y = A \cdot (\frac{L}{\tau})^B \tag{5}$$

dónde los valores de A y B se obtienen de la siguiente tabla en función de si estamos en un problema servo o en un problema de regulación. El valor de τ se corresponde con el valor encontrado en (2).

	Servo (Rovira)		Regulación (López)	
	A	B	A	B
Término de ganancia (K_p)	0.585	-0.916	0.859	-0.977
Término integral (T_i)	1.03	-0.165	0.674	-0.680

Para el caso de regulación se aplica:

$$Y = K_p \cdot k \tag{6}$$

$$Y = \frac{\tau}{T_i} \tag{7}$$

Para el caso servo el término de ganancia se calcula como antes, pero el término integral se calcula en base a:

$$\frac{\tau}{T_i} = A + B \cdot \frac{L}{\tau} \tag{8}$$

Si por contra, lo que queremos es sintonizar un controlador PID, la tabla a aplicar es

	Servo (Rovira)		Regulación (López)	
	A	B	A	B
Término de ganancia (K_p)	0.965	-0.85	1.357	-0.947
Término integral (T_i)	0.7964	-0.1465	0.842	-0.738
Término derivativo (T_d)	0.308	0.929	0.381	0.995

donde la constante derivativa se calcula en base a:

$$Y = \frac{T_d}{\tau} \tag{9}$$

2.3. Ejercicios

Para poder empezar a aplicar las reglas de sintonía necesitamos identificar el proceso descrito en la práctica 2, sí, el famoso sistema de los dos tanques interconectados con un caudal de entrada $Q = 2$. En este caso, tened en cuenta que si trabajamos con un modelo lineal, las variables que tendremos son variables incrementales, con lo que representarán incrementos respecto al punto de operación. En este caso, y debido a que ya se realizó la práctica correspondiente a la identificación de sistemas, se da el sistema linealizado. Este sigue la siguiente función de transferencia:

$$Pn(s) = \frac{0{,}2294e^{-0{,}4563s}}{6{,}5201s + 1} \tag{10}$$

2.3.1. Obtención de los parámetros del controlador - (2 puntos)

Una vez que sabemos cual es el modelo lineal de nuestro sistema a controlar, aplicaremos las diferentes reglas de sintonía para la obtención de los parámetros del controlador PID. En este caso, escogeremos aquel controlador que nos ofrezca el mejor comportamiento en términos de IAE:

$$IAE = \int_0^\infty |e(t)|dt \tag{11}$$

donde $e(t)$ se corresponde con el error de realimentación, es decir, con $e(t) = r(t) - y(t)$.

Por lo tanto, en este apartado se pide generar una tabla con los parámetros de los controladores PID que se obtienen para los métodos considerando la función de transferencia previamente mostrada:

- **Ziegler-Nichols (0.5 puntos)**
- **Rovira (0.75 puntos)**
- **López (0.75 puntos)**

Como plantilla de la tabla a generar utilizar la siguiente tabla:

Controladores			
Parámetro	Ziegler-Nichols	Rovira	López
K_p			
T_i			
T_d			

Tabla 2: Plantilla para la tabla a generar

2.3.2. Simulación del sistema de control lineal (1 punto)

Con la finalidad de poder evaluar las prestaciones del controlador, simularemos el funcionamiento del mismo ante un cambio unitario en la referencia y una perturbación unitaria en carga (es decir, a la entrada del proceso). En primer lugar abrid el archivo control_pid_lineal.mdl y realizar el esquema de control para el sistema lineal.

Una vez realizado esto, simular el funcionamiento ante un cambio unitario en referencia y una perturbación unitaria en carga. Mostrar los resultados en una figura en la que se muestren dos gráficas (utilizad subplots):

1. **las señales de referencia y salida**
2. **la señal de control que genera el controlador PID, esto es $u(t)$.**

La señal de control hay que interpretarla en términos de la magnitud física que representa (en nuestro caso caudal de entrada) en el correspondiente diagrama de control. **Al mostrar las gráficas de las respuestas temporales, tener en cuenta que estas deben mostrar, en una misma figura, la respuesta del sistema ante un cambio en referencia y ante la presencia de una perturbación.**

Por último, añadir una fila extra a la tabla anteriormente generada con el valor de IAE obtenido. La tabla ejemplo se corresponde con la siguiente:

Controladores			
Parámetro	Ziegler-Nichols	Rovira	López
K_p			
T_i			
T_d			
IAE			

Tabla 3: Plantilla para la tabla a generar

2.3.3. Simulación del sistema de control no-lineal (1 punto)

A continuación evaluaremos el funcionamiento del controlador finalmente ajustado en el apartado anterior cuando lo aplicamos al sistema no-lineal original (el de la práctica 3 y 4). Para ello, **abrid el documento control_pid_nolineal.mdl y realizad el esquema de control del modelo no-lineal.**

Simularemos el funcionamiento del controlador ante un cambio en la altura para el segundo tanque de 0.2 unidades y una perturbación en carga de dos unidades. Mostrar los resultados en el mismo formato que en el apartado anterior.

2.3.4. Comparación de prestaciones (1 punto)

Puesto que el controlador lo hemos diseñado en base a una aproximación lineal del sistema pero nuestro objetivo final es realizar el control del sistema no-lineal, conviene realizar una comparación del rendimiento que ofrece el controlador cuando lo aplicamos a:

1. El modelo nominal, $Pn(s)$, utilizado para el ajuste del controlador.
2. El sistema original no-lineal

Comparar las respuestas que genera el mismo controlador PID (es decir la salida del sistema) aplicado a cada uno de estos dos casos (modelo nominal ($Pn(s)$) y sistema no-lineal). Poner todas las gráficas en la misma figura indicando a que caso corresponde cada una. En esta comparación se deberá elaborar una tabla donde para cada uno de los controladores se muestre el IAE ante un cambio en referencia (de 0 a 0.2) y el IAE ante la presencia de una perturbación de 2 unidades. Esto debe hacerse por separado, es decir, calculad el IAE cuando solo hay un cambio en referencia (de 0 a 0.2) y solo

cuando hay una perturbación de 2 unidades. En base a esta tabla, elegir el que se consideraría mejor controlador.

IAE de los diferentes controladors PID				
	Cambio en referencia		Perturbación	
	$Pn(s)$	Modelo No-Lineal	$Pn(s)$	Modelo No-lineal
Ziegler-Nichols				
Rovira				
López				

Tabla 4: Plantilla de la tabla a rellenar

Importante: Los valores que cumplimentarán las tablas a rellenar se obtendrán todos de Matlab. Serán valores que se os devuelvan por pantalla!

3. Métodos de sintonización considerando Robustez

En las reglas de sintonía anteriores, no se considera la robustez del sistema de control. Por tanto, la robustez que acabe teniendo el sistema de control resultante será la que surja de aplicar el controlador PI o PID con el ajuste correspondiente. En este apartado veremos la robustez de los métodos anteriores así como un nuevo método de sintonía muy potente que sí incluye como consideración explícita el conseguir un determinado nivel de robustez para el sistema en lazo cerrado. Este método se llama uSORT y nos permite diseñar el controlador PID partiendo del valor máximo de la función de sensibilidad, el famoso M_s.

3.1. Obtención de la robustez para las sintonías aplicadas (1 punto)

En la práctica anterior, vimos como calcular el valor de Ms como indicador de robustez, para un sistema de control. Para calcular la robustez utilizaremos una función llamada *ms_basic_PID.m*.

- ms_basic_PID(K_p, T_i, T_d, Proc_tf)

Abrid la función y acabad de completarla. Ahora deberemos aplicar este procedimiento a cada uno de los lazos de control que resulta de la tabla que hemos construido en el apartado anterior.

1. **Completar la tabla anterior añadiendo una nueva columna con el valor de Ms de cada uno de los sistemas de control. Esto nos indicará el nivel de robustez de cada sistema de control.**
2. **En base al valor de Ms obtenido, indica que controladores calificarías como robustos y cuales no, indicando el motivo.**

3.2. Aplicación de una regla de sintonía robusta - Método M_s

Hasta este punto hemos estado trabajando con el controlador PID definido en (1). Dicho controlador se puede definir como el controlador ideal o básico por definición en el cual solo se tienen en cuenta los términos proporcional, integral y derivativo. En la realidad, dicho controlador no se utiliza con mucha asiduidad. Lo que realmente se utiliza es una modificación que nos añade dos grados de libertad, el controlador denominado PID estándar con 2 grados de libertad. Éste se define según (12).

$$u(s) = K_p(\beta r(s) - y(s) + \frac{1}{T_i s}e(s) - \frac{T_d s}{\alpha T_d s + 1}y(s)) \tag{12}$$

donde α se corresponde con el parámetro de corrección de la parte derivativa y es el primer grado de libertad. Normalmente, este parámetro adopta el valor de $\alpha = 0{,}1$. β se corresponde con el segundo orden de libertad que afecta únicamente a la señal de referencia. Por ahora, este valor se dejará en $\beta = 1$. Más adelante trataremos con su forma de calcularlo y para que nos servirá.

Lo que realizaremos ahora es completar la tabla anterior en base a la aplicación de una regla de sintonía que tiene en cuenta la robustez deseada para el sistema de control. Se trata de la regla uSORT (Unified Simple Robust Tuning). Esta regla se basa en las siguientes fórmulas para un ajuste servo o regulación y está pensado para procesos descritos en base a un modelo de la forma:

$$G(s) = \frac{ke^{-Ls}}{(\tau s + 1)(\tau \alpha s + 1)} \tag{13}$$

Este proceso es más genérico ya que incluye el modelo $Pn(s)$. Si $\alpha = 0$ lo que obtendremos al final es el mismo proceso $Pn(s)$. Definiendo $\tau_0 = L/\tau$, las fórmulas de sintonía son las siguientes:

- Para el caso servo
  1. $Kp \cdot K = a_o + a_1 \cdot \tau_0^{a_2}$
  2. $\frac{T_i}{\tau} = \frac{b_0 + b_1 \cdot \tau_0 + b_2 \cdot \tau_0^2}{b_3 + \tau_0}$
  3. $\frac{T_d}{\tau} = c_0 + c_1 \cdot \tau_0^{c_2}$
- Para el caso regulación
  1. $Kp \cdot K = a_o + a_1 \cdot \tau_0^{a_2}$
  2. $\frac{T_i}{\tau} = b_0 + b_1 \cdot \tau_0^{b_2}$
  3. $\frac{T_d}{\tau} = c_0 + c_1 \cdot \tau_0^{c_2}$

Las constantes de las fórmulas de sintonía se proporcionan en las tablas de la Figura 2 de acuerdo al controlador y robustez deseada. En caso de que el valor a no se encuentre en las tablas, se considera el valor interpolado entre los dos valores inmediatamente inferior y superior (No os preocupéis, en estas prácticas no tendréis que interpolar valores).

Table 1. Regulatory Control PI Tuning

	Controlled process time constants ratio a				
	0.0	0.25	0.50	0.75	1.0
Target robustness $M_S^t = 2.0$					
a_0	0.265	0.077	0.023	-0.128	-0.244
a_1	0.603	0.739	0.821	1.035	1.226
a_2	-0.971	-0.663	-0.625	-0.555	-0.517
Target robustness $M_S^t = 1.8$					
a_0	0.229	0.037	-0.056	-0.160	-0.289
a_1	0.537	0.684	0.803	0.958	1.151
a_2	-0.952	-0.626	-0.561	-0.516	-0.472
Target robustness $M_S^t = 1.6$					
a_0	0.175	-0.009	-0.080	-0.247	-0.394
a_1	0.466	0.612	0.702	0.913	1.112
a_2	-0.911	-0.578	-0.522	-0.442	-0.397
Target robustness $M_S^t = 1.4$					
a_0	0.016	-0.053	-0.129	-0.292	-0.461
a_1	0.476	0.507	0.600	0.792	0.997
a_2	-0.708	-0.513	-0.449	-0.368	-0.317
b_0	-1.382	0.866	1.674	2.130	2.476
b_1	2.837	0.790	0.268	0.112	0.073
b_2	0.211	0.520	1.062	1.654	1.955

(a)

Table 3. Servo-Control PI Tuning

	Controlled process time constants ratio a				
	0.0	0.25	0.50	0.75	1.0
Target robustness $M_S^t = 1.8$					
a_0	0.243	0.094	0.013	-0.075	-0.164
a_1	0.509	0.606	0.703	0.837	0.986
a_2	-1.063	-0.706	-0.621	-0.569	-0.531
Target robustness $M_S^t = 1.6$					
a_0	0.209	0.057	-0.010	-0.130	-0.220
a_1	0.417	0.528	0.607	0.765	0.903
a_2	-1.064	-0.667	-0.584	-0.506	-0.468
Target robustness $M_S^t = 1.4$					
a_0	0.164	0.019	-0.061	-0.161	-0.253
a_1	0.305	0.420	0.509	0.636	0.762
a_2	-1.066	-0.617	-0.511	-0.439	-0.397
b_0	14.650	0.107	0.309	0.594	0.625
b_1	8.450	1.164	1.362	1.532	1.778
b_2	0.0	0.377	0.359	0.371	0.355
b_3	15.740	0.066	0.146	0.237	0.209

(b)

Table 2. Regulatory Control PID Tuning

	Controlled process time constants ratio a				
	0.0	0.25	0.50	0.75	1.0
Target robustness $M_S^t = 2.0$					
a_0	0.235	0.435	0.454	0.464	0.488
a_1	0.840	0.551	0.588	0.677	0.767
a_2	-0.919	-1.123	-1.211	-1.251	-1.273
Target robustness $M_S^t = 1.8$					
a_0	0.210	0.380	0.400	0.410	0.432
a_1	0.745	0.500	0.526	0.602	0.679
a_2	-0.919	-1.108	-1.194	-1.234	-1.257
Target robustness $M_S^t = 1.6$					
a_0	0.179	0.311	0.325	0.333	0.351
a_1	0.626	0.429	0.456	0.519	0.584
a_2	-0.921	-1.083	-1.160	-1.193	-1.217
Target robustness $M_S^t = 1.4$ †					
a_0	0.155	0.228	0.041	0.231	0.114
a_1	0.455	0.336	0.571	0.418	0.620
a_2	-0.939	-1.057	-0.725	-1.136	-0.932
†Valid only for $\tau_o \geq 0.40$ if $a \geq 0.25$					
b_0	-0.198	0.095	0.132	0.235	0.236
b_1	1.291	1.165	1.263	1.291	1.424
b_2	0.485	0.517	0.496	0.521	0.495
c_0	0.004	0.104	0.095	0.074	0.033
c_1	0.389	0.414	0.540	0.647	0.756
c_2	0.869	0.758	0.566	0.511	0.452

(c)

Table 4. Servo-Control PID Tuning

	Controlled process time constants ratio a				
	0.0	0.25	0.50	0.75	1.0
Target robustness $M_S^t = 2.0$					
a_0	0.377	0.502	0.518	0.533	0.572
a_1	0.727	0.518	0.562	0.653	0.728
a_2	-1.041	-1.194	-1.290	-1.329	-1.363
Target robustness $M_S^t = 1.8$					
a_0	0.335	0.432	0.435	0.439	0.482
a_1	0.644	0.476	0.526	0.617	0.671
a_2	-1.040	-1.163	-1.239	-1.266	-1.315
Target robustness $M_S^t = 1.6$					
a_0	0.282	0.344	0.327	0.306	0.482
a_1	0.544	0.423	0.488	0.589	0.622
a_2	-1.038	-1.117	-1.155	-1.154	-1.221
Target robustness $M_S^t = 1.4$					
a_0	0.214	0.234	0.184	0.118	0.147
a_1	0.413	0.352	0.423	0.575	0.607
a_2	-1.036	-1.042	-1.011	-0.956	-1.015
b_0	1687	0.135	0.246	0.327	0.381
b_1	339.2	1.355	1.608	1.896	2.234
b_2	39.86	0.333	0.273	0.243	0.204
b_3	1299	0.007	0.003	-0.006	-0.015
c_0	-0.016	0.026	-0.042	-0.086	-0.110
c_1	0.333	0.403	0.571	0.684	0.772
c_2	0.815	0.613	0.446	0.403	0.372

(d)

Figura 2: Tablas del método uSORT

En este apartado se pide realizar dos tablas, una tabla con los valores de K_p, T_i y T_d obtenidos y una tabla similar a la de los apartados anteriores pero considerando únicamente la aplicación, al mismo modelo obtenido ($Pn(s)$), de la sintonía uSORT en los siguientes casos:

- un PI robusto para operar en regulación (0.25 puntos)
- un PI poco robusto para operar en regulación (0.25 puntos)
- un PI robusto para operar en servo (0.25 puntos)
- un PI poco robusto para operar en servo (0.25 puntos)
- un PID robusto para operar en regulación (0.25 puntos)
- un PID poco robusto para operar en regulación (0.25 puntos)
- un PID robusto para operar en servo (0.25 puntos)
- un PID poco robusto para operar en servo (0.25 puntos)

En el caso de los controladores robustos se pide considerar un Ms = 1.6. Para los casos de controladores poco robustos se pide utilizar los valores de Ms que proporcionen el nivel de robustez más bajo. Para cada uno de los casos se deberá evaluar el IAE para regulación y para servo, así como el valor de Ms obtenido (para obtener el valor de Ms se utilizará la función *ms_filter_PID.m*, la cual deberéis acabar de implementar).

Para calcular el valor de M_s utilizaremos una reinterpretación matemática de (12). Si reorganizamos la expresión, el PID se puede representar como:

$$U(s) = K_p \cdot (\beta + \frac{1}{T_i s}) \cdot r(s) - K_p \cdot (1 + \frac{1}{T_i s} + \frac{T_d s}{\alpha T_d s + 1}) \cdot y(s) \tag{14}$$

donde claramente podemos diferenciar entre la parte que afecta a la referencia $C_r(s)$ y la parte que afecta a la señal de feedback $C_y(s)$:

$$U(s) = C_r(s) \cdot r(s) - C_y(s) \cdot y(s) \tag{15}$$

En este caso, la única componente a tener en cuenta para el cálculo del M_s es $C_y(s)$.

Las tablas ejemplos que deberéis completar son las siguientes:

Parámetros para diferentes controladores PID				
Controlador	Servo		Regulation	
	$M_s = 1{,}6$	$M_s = 2{,}0$	$M_s = 1{,}6$	$M_s = 2{,}0$
K_{PI}				
$T_{i_{PI}}$				
$T_{d_{PI}}$				
K_{PID}				
$T_{i_{PID}}$				
$T_{d_{PID}}$				

Tabla 5: Tabla ejemplo para los parámetros del controlador

IAE para controladores PI diferentes y PID standar						
Controlador PID	M_s de diseño	Servo		Regulación		M_s obtenido
		Lineal	No Lineal	Lineal	No Lineal	
PI Servo	1.6					
PI Servo	1.8					
PI Regulación	1.6					
PI Regulación	2.0					
PID Servo	1.6					
PID Servo	2.0					
PID Regulación	1.6					
PID Regulación	2.0					

Tabla 6: Tabla ejemplo para los valores de IAE de los diferentes controladores

3.3. Consideración de un segundo grado de libertad para la sintonía en regulación (1 punto)

El método uSORT proporciona un ajuste óptimo en el sentido IAE restringido a garantizar una determinada robustez. Cuando el ajuste escogido es el de regulación, la respuesta ante un salto en referencia puede mejorarse añadiendo un segundo grado de libertad al controlador (ya sea un PI o un PID) mediante el término β que comentábamos en (12). Éste se define como el factor de peso en la referencia (nótese que es equivalente al PID estándard de 1 grado de libertad si el valor de β es equivalente a 1).

Este factor de peso se obtiene en el caso del uSORT mediante la expresión

$$\beta = d_0 + d_1(\tau_0)^{d_2} \tag{16}$$

donde las constantes d_x se obtienen de:

	Target robustness M_S^t			
	2.0	1.8	1.6	1.4
	PI Controller			
d_0	0.730	0.658	0.649	0.811
d_1	0.302	0.578	0.900	1.205
d_2	0.368	0.372	0.446	0.608
	PID Controller			
d_0	0.306	0.248	0.255	0.383
d_1	0.416	0.571	0.277	0.921
d_2	0.367	0.362	0.476	0.612

En base a este ajuste adicional se trata de ver

- **como mejora el valor de IAE para el caso de un salto en referencia cuando utilizamos una sintonía para regulación.**
- **Verificar que los valores de IAE para el caso de rechazo de perturbación continúan siendo los mismos, así como el valor de robustez obtenido.**

En este caso, replicad la tabla del apartado anterior modificando aquellos valores que mejoran al aplicar el segundo grado de libertad.

3.4. Elección del mejor controlador (1 punto)

En base a las evaluaciones anteriores, atendiendo tanto a consideraciones de robustez como a la necesidad de tener que operar tanto en regulación como en servo. ¿Qué controlador escogerías? Mostrad en una misma gráfica el comportamiento para el sistema lineal y el sistema no-lineal.

A modo informativo, el controlador PID utilizado en este apartado se basa en la implementación mostrada en la Figura 3. En este caso, los valores de β pueden ser 1 en el caso de utilizar el método uSORT sin considerar la corrección de β o bien el valor calculado utilizando (16). α se mantiene en 0.1 mientras que aparece una variable extra, γ. la cual dejaremos siempre a 0.

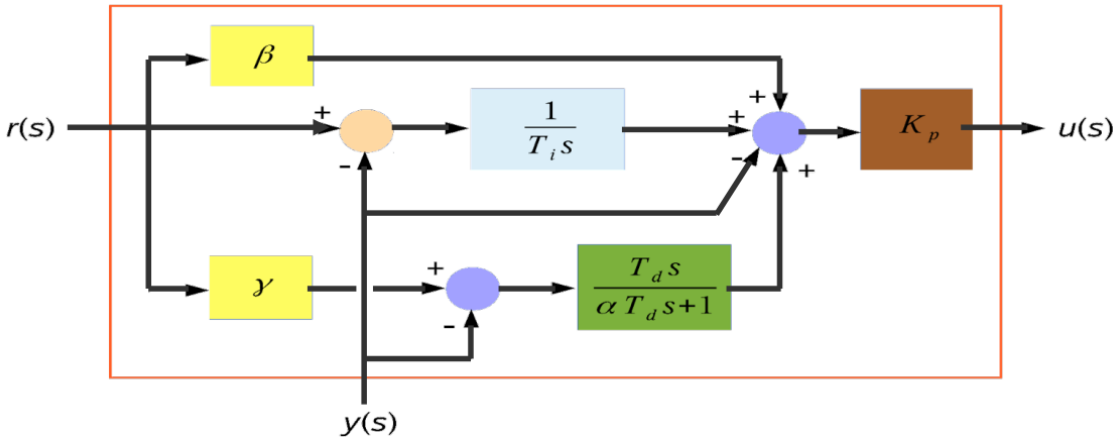

Figura 3: Implementación PID estándar