

Ejemplo criterios de elección de metodología de diseño IMC/S-IMC y especificaciones razonables (5)

Enunciado del problema

Considérese la siguiente función de transferencia:

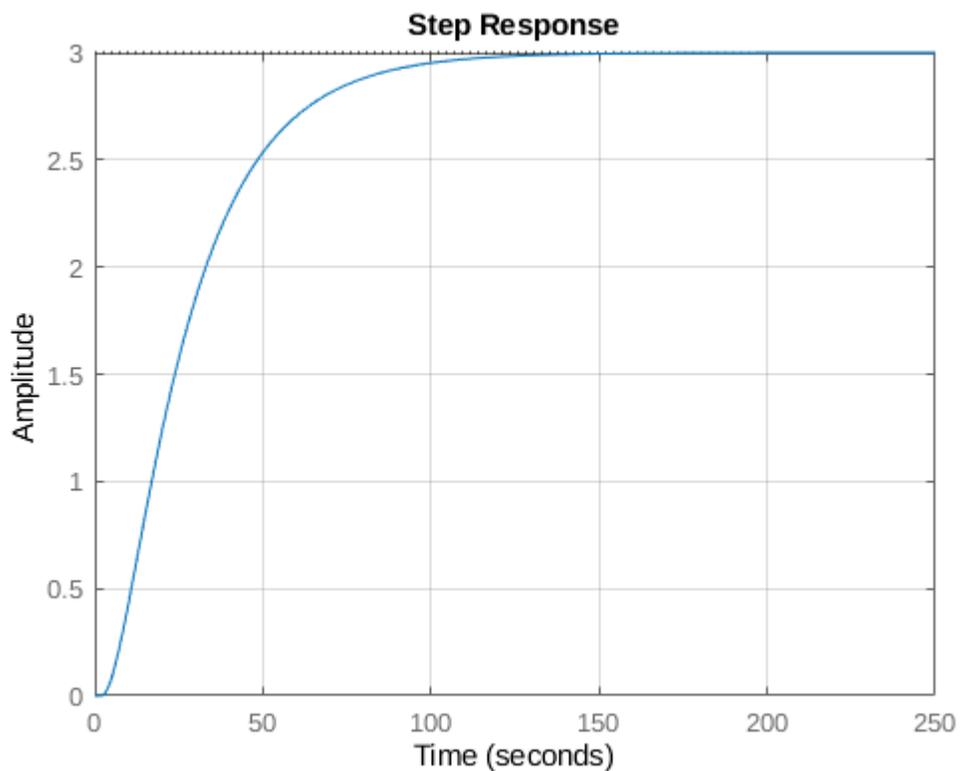
```
s=tf('s');  
G_sinretardo=3/(6*s+1)/(22*s+1);  
G=G_sinretardo*exp(-2*s);zpk(G)
```

ans =

$$\exp(-2*s) * \frac{0.022727}{(s+0.1667)(s+0.04545)}$$

Continuous-time zero/pole/gain model.

```
step(G), grid on
```



Justificar qué control IMC/S-IMC/ninguno de ellos se aconsejaría aplicar, y si se podría sintonizar un PID basado en dichas metodologías, según el rango de tiempo de establecimiento deseado.

Solución:

Como el retardo es de 2, para que la aproximación del retardo del PID obtenido con S-IMC (Taylor) sea aceptable, se recomienda $\tau_c \geq 2$, esto es $t_{est,objetivo} = 4\tau_c + \theta \geq 5\theta = 10$ segundos. Por tanto, para tiempo de establecimiento objetivo mayor de 10 se recomienda un PID por S-IMC.

El IMC se podría diseñar teóricamente para cualquier $\tau_c > 0$, esto es para $t_{est,objetivo} \geq 2$. Pero sólo "teóricamente":

- El regulador IMC implícito tendría retardo interno. No sería equivalente a un PID. Por tanto, aunque perfectamente válido teóricamente, la preferencia industrial por los PID recomendaría el PID/S-IMC para $\tau_c \geq 2$, $t_{est,objetivo} \geq 10$, como se ha dicho antes.
- Si se acelera "demasiado" la respuesta ante referencia en reguladores de cancelación, las perturbaciones a la entrada tardan a compensarse mucho más que la referencia (aparte, se amplifica mucho el ruido de medida). En concreto, S-IMC propone abandonar la filosofía de cancelación cuando el segundo término del mínimo en $\tau_l = \min(\tau_1, 4(\tau_c + \theta))$ es el menor, que en este caso se produce con $\tau_c = 22/4 - 2 = 3.5$, que equivale a $t_{est,objetivo} = 16$ s.

Resumen:

- $t_e \leq 2$: imposible, no se puede ser más rápido que el retardo físico del proceso.
- $t_e \in [2, 10]$: en teoría, sería correcto usar IMC, resultando regulador con retardo interno \neq PID... seguramente diseño no satisfactorio en la práctica: la respuesta ante perturbaciones de entrada durará siempre 90 segundos, y también amplificará ruido medida en acción de control excesiva.
- $t_e \geq 10$: IMC será aproximado con prestaciones prácticamente idénticas por un PID/S-IMC, aconsejable este último, pues... aunque internamente no será "realmente" IMC para $t_e \in [10, 16]$.

Detalle/simulación:

1. Vamos a simular un ejemplo de dicha propuesta con tiempo de establecimiento "alto" >16.

```
proc.tau1=22;proc.tau2=6;proc.kp=dcgain(G);proc.theta=2;
tauc=8;
t_est_objetivo=4*tauc+proc.theta
```

```
t_est_objetivo = 34
```

```
PID=SIMC_Aux.S_IMC(proc,tauc,7); %filtro derivada en tau2/7
```

```
Diseñado S_IMC con K_c=7.333333e-01, tau_I=22, tau_d=6
```

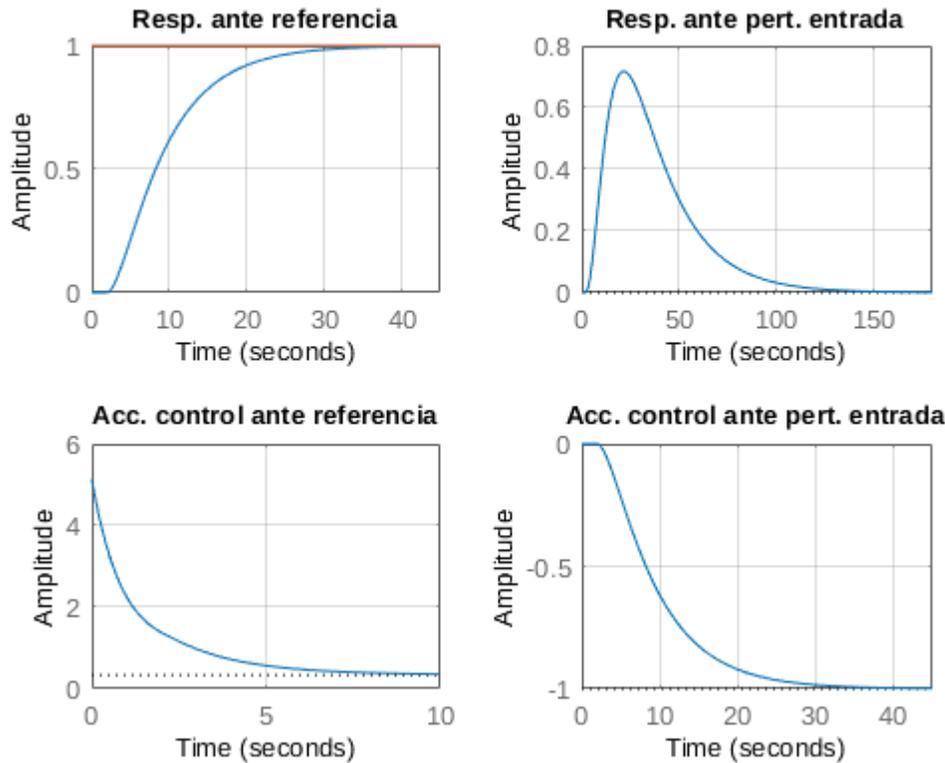
```
zpk(PID)
```

```
ans =
```

$$\frac{5.1333 (s+0.1667) (s+0.04545)}{s (s+1.167)}$$

Continuous-time zero/pole/gain model.

SIMC_Aux.SimulaRefYEntrada (G, PID)



2. Ahora, con tiempos de establecimiento deseado "intermedios", un PID S-IMC resultaría:

```
tauc=2.2;
t_est_objetivo=4*tauc+proc.theta
```

```
t_est_objetivo = 10.8000
```

```
PID=SIMC_Aux.S_IMC(proc,tauc,15); %filtro derivada en tau2/15
```

S-IMC: evitamos cancelación polo dominante cerca de origen
 Diseñado S_IMC con $K_c=1.746032e+00$, $\tau_I=1.680000e+01$, $\tau_d=6$

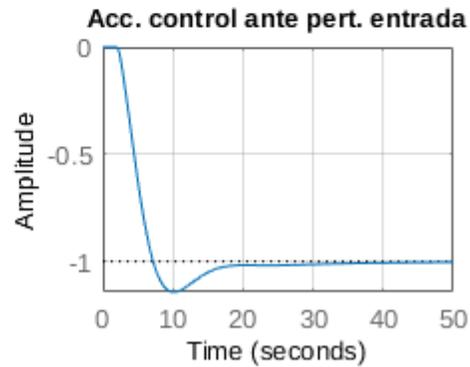
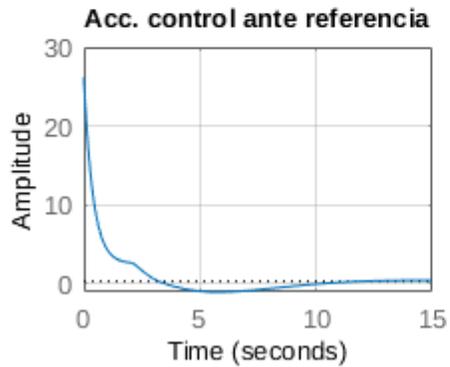
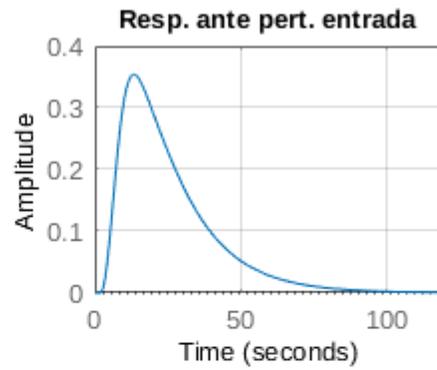
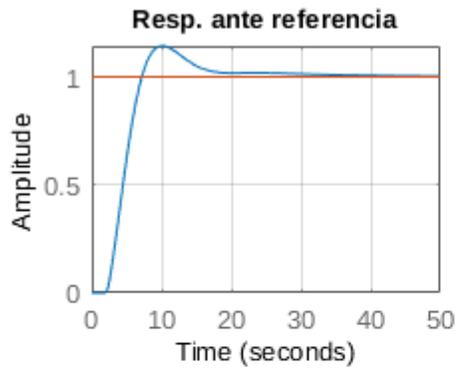
```
zpk(PID)
```

```
ans =
```

$$\frac{26.19 (s+0.1667) (s+0.05952)}{s (s+2.5)}$$

Continuous-time zero/pole/gain model.

SIMC_Aux.SimulaRefYEntrada (G, PID)



3. Un IMC "puro" con respuesta con t_e objetivo muy rápido es poco probable que funcione en la práctica:

```
tauc=1.8
```

```
tauc = 1.8000
```

```
t_est_objetivo=4*tauc+proc.theta
```

```
t_est_objetivo = 9.2000
```

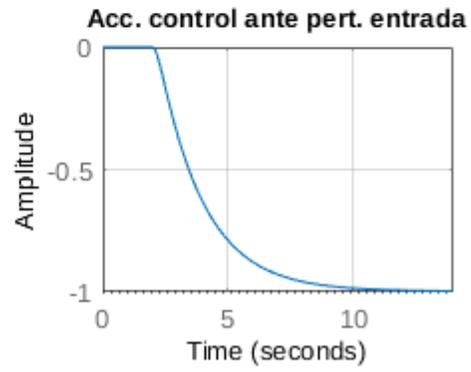
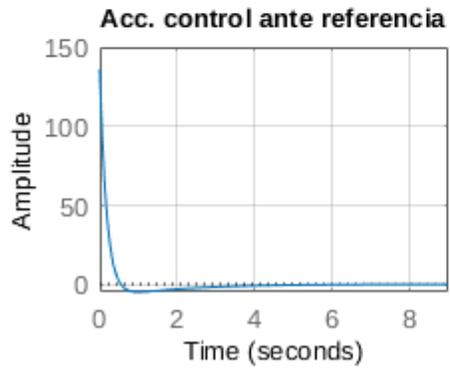
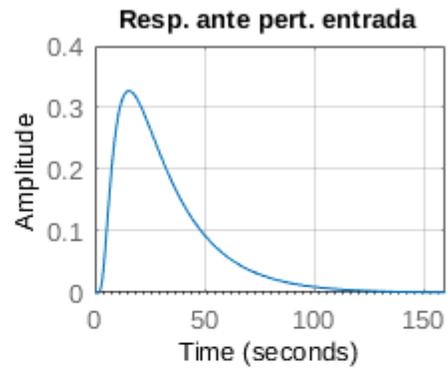
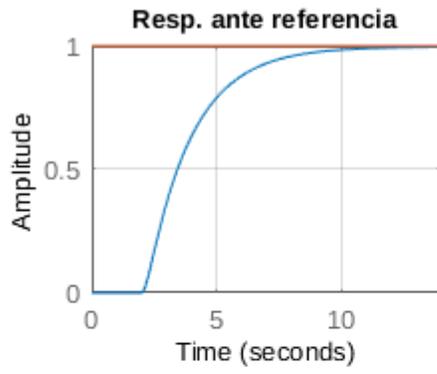
```
Q=inv(G_sinretardo)*1/(tauc*s+1)/(tauc/10*s+1);
zpk(Q) %realizable
```

```
ans =
```

```
135.8 (s+0.1667) (s+0.04545)
-----
(s+5.556) (s+0.5556)
```

```
Continuous-time zero/pole/gain model.
```

```
Reg=SIMC_Aux.SimulaIMC(G,Q);
```



Reg.InternalDelay

ans = 2