Sistema Realimentados

EP4 - Síntese direta para $G(S) = \frac{K}{\tau s + 1}$ **e modelo de referência de ordem 2**

Data: 2 de abril

Componentes: Dionatas Santos Brito e Filipe Ferreira de Oliveira

Projete um controlador para a FT $G(s) = \frac{0.2}{0.25s + 1}$

1) Escolha os parâmetros do modelo de referência $T(s) = \frac{\omega^2}{s^2 + 2\zeta\omega s + \omega^2{}_n}$ para ter sobreelevação inferior a 5% e tempo de estabelecimento menor que 0.8s.

Os parâmetros de um modelo de segunda ordem têm a seguinte relação com as especificações de sobreelevação e tempo de estabelecimento:

$$MP = \frac{c(t_p) - c(\infty)}{c(\infty)} = 100e^{-\pi\frac{\zeta}{\sqrt{1-\zeta^2}}} < 5\%$$

Logo: $\zeta = \sqrt{\frac{\ln(\frac{M_p}{100})^2}{\pi^2 + \ln(\frac{M_p}{100})^2}}$

E para o tempo de estabelecimento temos:

$$t_s = \frac{4}{\omega_n \zeta} < 0.8s$$

```
MP=5;
a = (log(MP/100)).^2;
zeta = sqrt(a/(pi.^2+a)) % 0.6901
```

```
zeta = 0.6901
```

```
ts=0.8;
omega_n=4/(ts*zeta) % 7.2453
```

```
omega_n = 7.2453
```

Para sobreelevação < 5% → $\zeta > 0.69$;

E para tempo de estabelecimento < 0.8s → $\omega_n > 7.245$;

2) Calcule os parâmetros do controlador $C(s)$ tal que $\dfrac{C(s)G(s)}{1+C(s)G(s)} = T(s)$

Considerando um controlador PI: $C(s) = K_p\left(1 + \dfrac{1}{T_i s}\right)$ e $G(s) = \dfrac{K}{\tau s + 1}$

A FT de malha aberta será $C(s)G(s) = \dfrac{KK_p(1 + T_i s)}{sT_i(\tau s + 1)}$

Fechando a malha, temos: $M(s) = \dfrac{KK_p(1 + T_i s)}{s^2 + \dfrac{1 + KK_p}{\tau}s + \dfrac{KK_p}{T_i \tau}}$

Comparando a equação cararcterística de $M(s)$ com o modelo de segunda ordem:

$$s^2 + \frac{1 + KK_p}{\tau}s + \frac{KK_p}{T_i\tau} = s^2 + 2\zeta\omega_n s + \omega_n^2$$

Daí, temos as equações para os parâmetros do controlador:

$$K_p = \frac{2\zeta\omega_n\tau - 1}{K}$$
$$T_i = \frac{2\zeta\omega_n\tau - 1}{\omega_n^2\tau}$$

Calculando a equação do controlador:

```
% Escolhendo zeta e omega_n que atendam a especificação
zeta=0.71; % zeta > 0.69
ts=0.65;    % ts  < 0.8
omega_n=4/(ts*zeta); % omega_n = 8.6674

tau=0.25;
K=0.2;
Kp=(2*zeta*omega_n*tau-1)/K % Kp = 10.3846
```

```
Kp = 10.3846
```

```
Ti=(2*zeta*omega_n*tau-1)/(omega_n^2*tau) % Ti = 0.1106
```

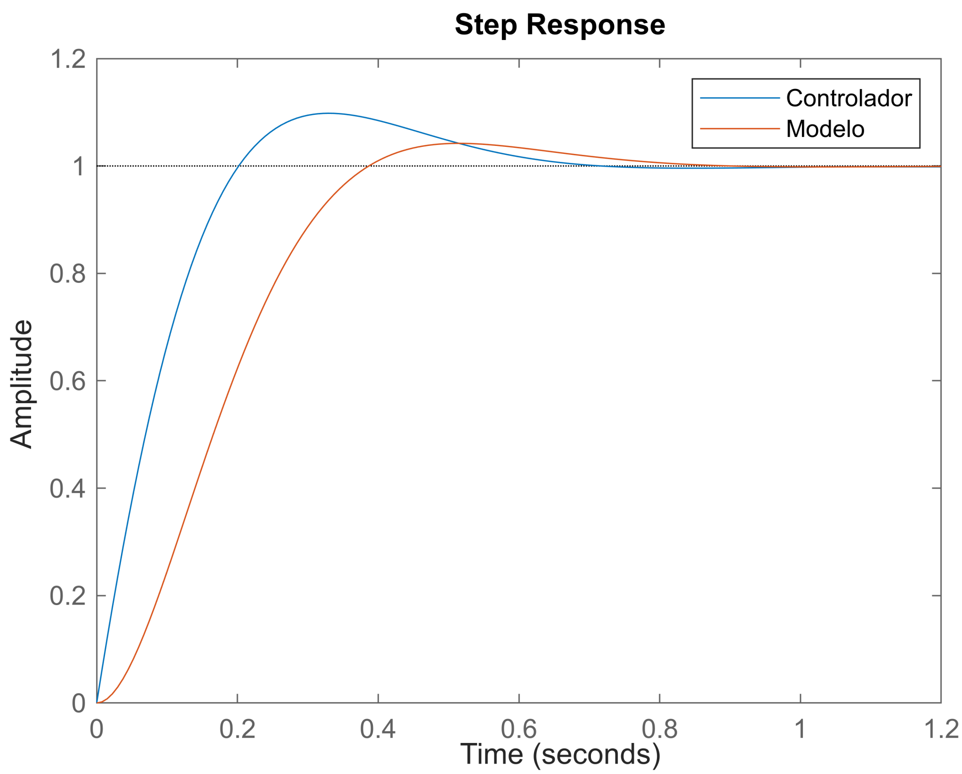
```
Ti = 0.1106
```

```
C = tf(Kp*[Ti 1], [Ti 0]);
```

3) Plote a resposta ao degrau em malha fechada e compare com T(s), comentando as diferenças.

```
s = tf('s');
G=(0.2/(0.25*s+1));
T=omega_n^2/(s^2+2*zeta*omega_n*s+omega_n^2);
M=feedback(C*G,1);
step(M,T)
```

```
legend('Controlador','Modelo')
```




```
cont=stepinfo(M);
mod=stepinfo(T);
```

Especificações do modelo:

```
mod_ts =mod.SettlingTime % 0.6870
```

```
mod_ts = 0.6870
```

```
mod_overshoot =mod.Overshoot % 4.2106
```

```
mod_overshoot = 4.2106
```

Especificações com controlador:

```
const_ts =cont.SettlingTime % 0.5884
```

```
const_ts = 0.5884
```

```
cont_overshoot =cont.Overshoot % 9.8131
```

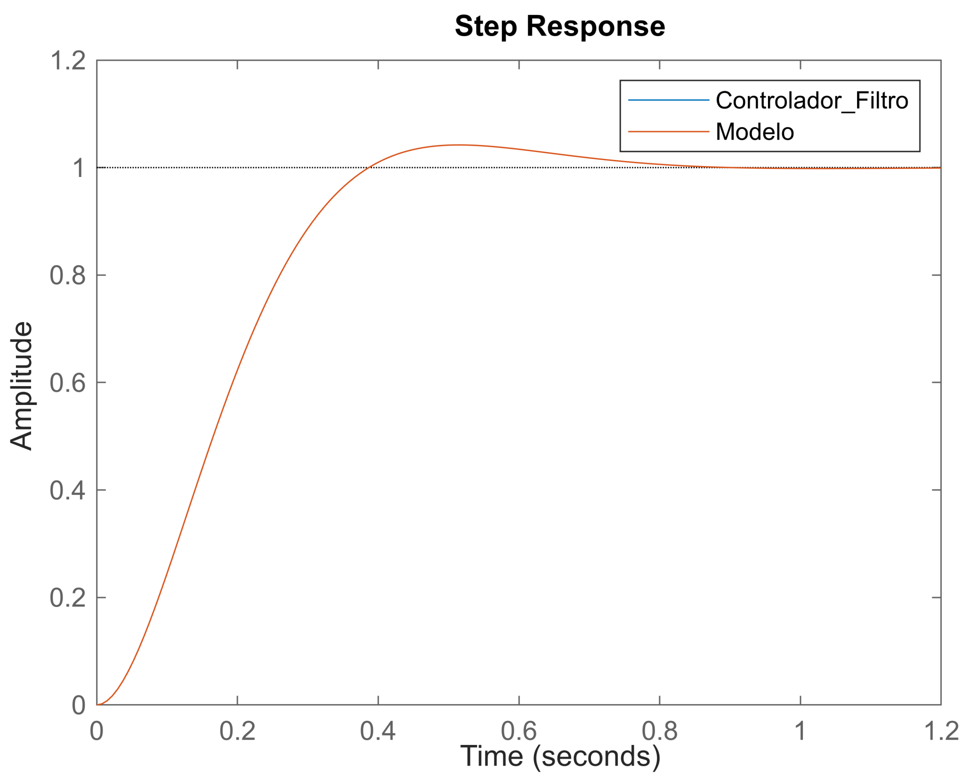
```
cont_overshoot = 9.8131
```

Ao analisar o sistema com o controlador, é possível retirar as seguintes informações: tempo de estabelecimento de 0.5884 segundos e overshoot de 9.8%.

Isso se vê refletido ao análisar o gráfico de resposta, onde o controlador teve um tempo de subida menor e chegou no valor de regime mais rápido, ao se comprar com o modelo

Vemos que o sistema não atingiu a especificação de sobreelevaçãoo, logo podemos adicionar um filtro afim de cancelar o zero introduzido pelo PI

```
F = tf(1,[Ti 1]);
step(F*M,T)
legend('Controlador_Filtro','Modelo')
```




```
modF=stepinfo(F*M);
```

Especificações do modelo com filtro:

```
modF_ts = modF.SettlingTime %0.6870
```

```
modF_ts = 0.6870
```

```
modF_overshoot =modF.Overshoot % 4.2106
```

```
modF_overshoot = 4.2106
```

Adicionando o filtro temos uma resposta igual ao modelo especificado

4) Refaça então o projeto do controlador C(s) usando o modelo de referência $T(s) = \frac{1}{\lambda s + 1}$, para atender a mesma especificação do item 1).

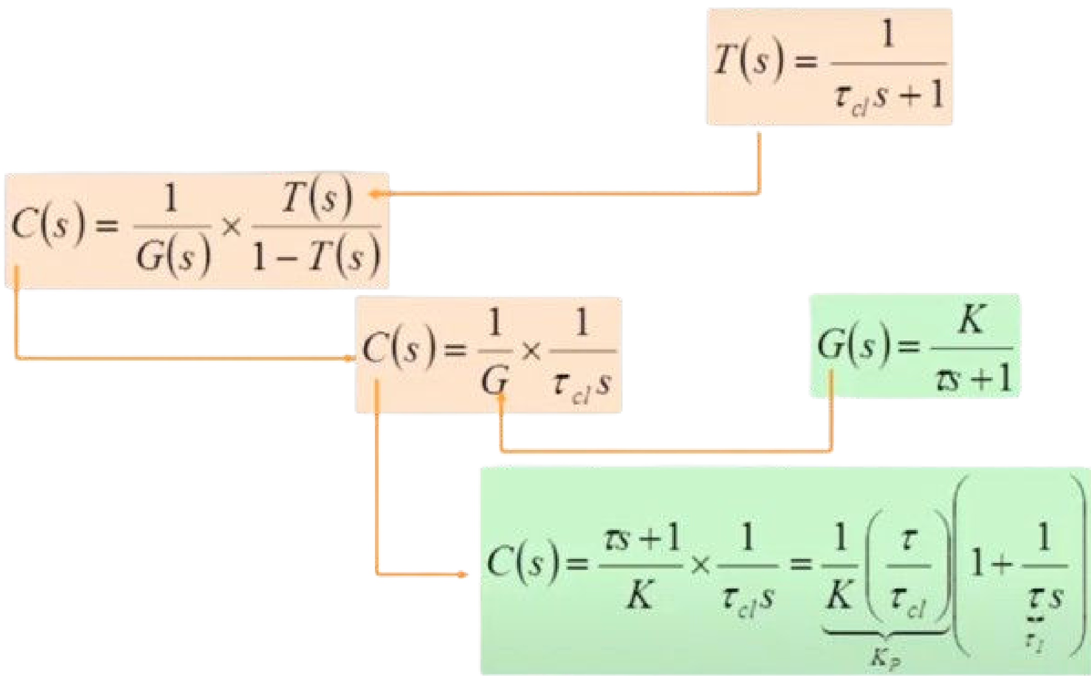
Para um modelo de referência de ordem 1, devemos seguir apenas a especificação de tempo de estabelecimento por não haver sobreelevação.

Para isso, é necessário escolher um valor de λ que atenda os requisitos:

$t_s = 5\lambda$

$\lambda = \frac{0.8}{5} = 0.16$

Logo $\lambda < 0.16$



Em outras palavras, $C(s) = K_p\left(1 + \frac{1}{\tau_i s}\right)$ com $K_p = \frac{\tau}{K\lambda}$ e $\tau_i = \tau$.

```
%parametros
K= 0.2;
lambda = 0.15;
tau = 0.25;
Kp = tau/(K*lambda); %8.3333

%controlador
C4 = Kp*(1 + (1/(tau*s)))
```

```
C4 =
```

```
  2.083 s + 8.333
  ---------------
      0.25 s

Continuous-time transfer function.
Model Properties
```

```
%utilizando as funções pidtuning
[C_foptd, iae]=pidtuning(G,0.15);
[C_order1, iae]=pidtuning(G,'method','lambda','type','PI','param',0.15);

%fechando a malha
M_calc = feedback(C4*G,1);
M_foptd = feedback(C_foptd*G,1);
M_order1 = feedback(C_order1*G,1);

%retirando a constante de tempo (ts)
info_calc = stepinfo(M_calc);
info_foptd = stepinfo(M_foptd);
info_order1 = stepinfo(M_order1);

up_calc = info_calc.Overshoot % 0%
```

Não pode haver dúvidas sobre como fechar as malhas!

```
up_calc = 0
```

```
ts_calc = info_calc.SettlingTime %0.5868
```

```
ts_calc = 0.5868
```

```
ts_foptd = info_foptd.SettlingTime %0.5868
```

```
ts_foptd = 0.5868
```

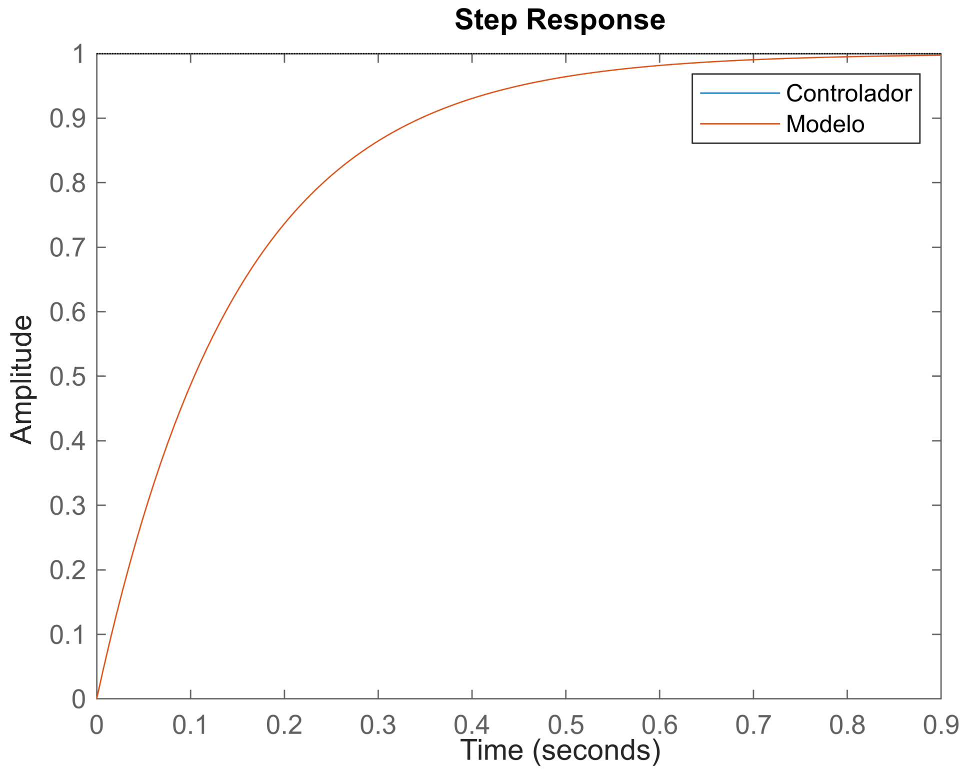
```
ts_order1 = info_order1.SettlingTime %0.5868
```

```
ts_order1 = 0.5868
```

Com as informações apresentadas, é possível notar que ao utilizar a função "pidtuning.m", tanto o método com o foptd quanto o de ordem 1 resultaram no mesmo controlador que o calculado, com o tempo de estabelecimento de 0.5868 segundos.

5) Plote a resposta ao degrau em malha fechada e compare com T(s), comentando as diferenças, caso houver.

```
%T(s)=1/(λs+1)
s = tf('s');
lambda = 0.15;
Ts=(1/(lambda*s+1));
step(M_calc,Ts);
legend('Controlador','Modelo')
```




```
info_model = stepinfo(Ts);
ts_model = info_model.SettlingTime %0.5868
```

```
ts_model = 0.5868
```

Como podemos observar, temos uma resposta bem próxima ao modelo especificado, tempo de estabelecimento de 0.5868 segundos.

Importante ressaltar que no segundo projeto zero do controlador cancelou o polo da função de transferência, resultando em um modelo em malha fechada de ordem 1.

No primeiro projeto o modelo em malha fechada é de ordem 2, e a resposta somente ficou igual à especificada após cancelar o zero com o pre-compensador F(s).