Sistemas Realimentados - Turma 2

EP5 - Comparação dos métodos: síntese direta, Ziegler-Nichols e IAE ótimo.

Vinicius Cole de Amorim

Seja a FT $G(s) = \dfrac{0.7e^{-2s}}{(5s+1)}$

1) Projete um controlador PI via primeiro método de Ziegler-Nichols.

$$G(s) = \frac{Ke^{-\theta s}}{(\tau s+1)} = \frac{0.7e^{-2s}}{(5s+1)}$$

$$K = 0.7;\ \theta = 2;\ \tau = 5$$

Pela Regra de sintonia de Ziegler-Nichols:

Ziegler-Nichols tuning

$$\begin{bmatrix} \text{Controller} & \text{Kp} & T_i & T_d \\ P & \dfrac{\tau}{K\theta} & - & - \\ \text{PI} & \dfrac{0.9\tau}{K\theta} & 3.33\theta & - \\ \text{PID} & \dfrac{1.2\tau}{K\theta} & 2\theta & 0.5\theta \end{bmatrix}$$

$$K_p = \frac{0.9*5}{0.7*2} = 3.21$$

$$T_i = 3.33*2 = 6.66$$

$$G_c = K_p\left(1+\frac{1}{T_i s}\right) = 3.21\left(1+\frac{1}{6.66s}\right) = \frac{21.3786s+3.21}{6.66s}$$

2) Projete um controlador PI via método do IAE ótimo.

$$t = \frac{\tau}{\theta}$$

$$\begin{bmatrix} \text{Controller} & K_p & T_i & T_d \\ \text{PI} & (0.7589/K)*(t\hat{}0.861) & \tau/(1.02-0.323/t) & - \\ \text{PID} & (1.086/K)*(t\hat{}0.869) & \tau/(0.74-0.130/t) & 0.348\tau(t\hat{}0.914) \end{bmatrix}$$

$$t = \frac{5}{2} = 2.5$$

$$K_p = \left(\frac{0.7589}{0.7}\right) * (2.5^{0.861}) = 2.38$$

$$T_i = \frac{5}{\left(1.02 - \frac{0.323}{2.5}\right)} = 5.61$$

$$G_c = K_p\left(1+\frac{1}{T_i s}\right) = 2.38\left(1+\frac{1}{5.61s}\right) = \frac{13.3518s+2.38}{5.61s}$$

3) Usando os controladores projetados, compare as resposta em malha fechada á entrada degrau em termos de sobreelevação, tempo de estabelecimento e IAE

```
s = tf('s')
gs = exp(-2*s)*(0.7/(5*s+1))

controladorZN = 3.21*(1 + 1/(6.66*s))
```

```
controladorZN =

  21.38 s + 3.21
  --------------
      6.66 s

Continuous-time transfer function.
Model Properties
```

```
controladorIAE = 2.38*(1 + 1/(5.61*s))
```

```
controladorIAE =

  13.35 s + 2.38
  --------------
      5.61 s
```

```
Continuous-time transfer function.
Model Properties
```

```
FTZN = (gs * controladorZN)/(1 + controladorZN*gs)
FTIAE = (gs * controladorIAE)/(1 + controladorIAE*gs)
infoZN = stepinfo(FTZN)
```

```
infoZN = struct with fields:
         RiseTime: 1.8777
    TransientTime: 20.9664
     SettlingTime: 20.9664
      SettlingMin: 0.8508
      SettlingMax: 1.2789
        Overshoot: 27.8683
       Undershoot: 0
             Peak: 1.2789
         PeakTime: 6.1561
```

```
infoIAE = stepinfo(FTIAE)
```

```
infoIAE = struct with fields:
         RiseTime: 2.6173
    TransientTime: 15.1515
     SettlingTime: 15.1515
      SettlingMin: 0.9139
      SettlingMax: 1.1259
        Overshoot: 12.5270
       Undershoot: 0
             Peak: 1.1259
         PeakTime: 7.2446
```

```
stabilization_time = max([infoZN.SettlingTime, infoIAE.SettlingTime])
time = 0:0.01:stabilization_time

[y1, t] = step(FTZN, time)
[y2, ~] = step(FTIAE, time)

figure
plot(t, y1, 'DisplayName', "ZN", "Color", "blue")
hold on
plot(t, y2, 'DisplayName', "IAE", "Color", "red")
line([min(time), max(time)], [1, 1], 'Color', 'black', 'LineStyle', '--',
'DisplayName', '1');
hold off

disp(['ZN: ', num2str(infoZN.Overshoot), '%'])
```

```
ZN: 27.8683%
```

```
disp(['IAE: ', num2str(infoIAE.Overshoot), '%'])
```

```
IAE: 12.527%
```

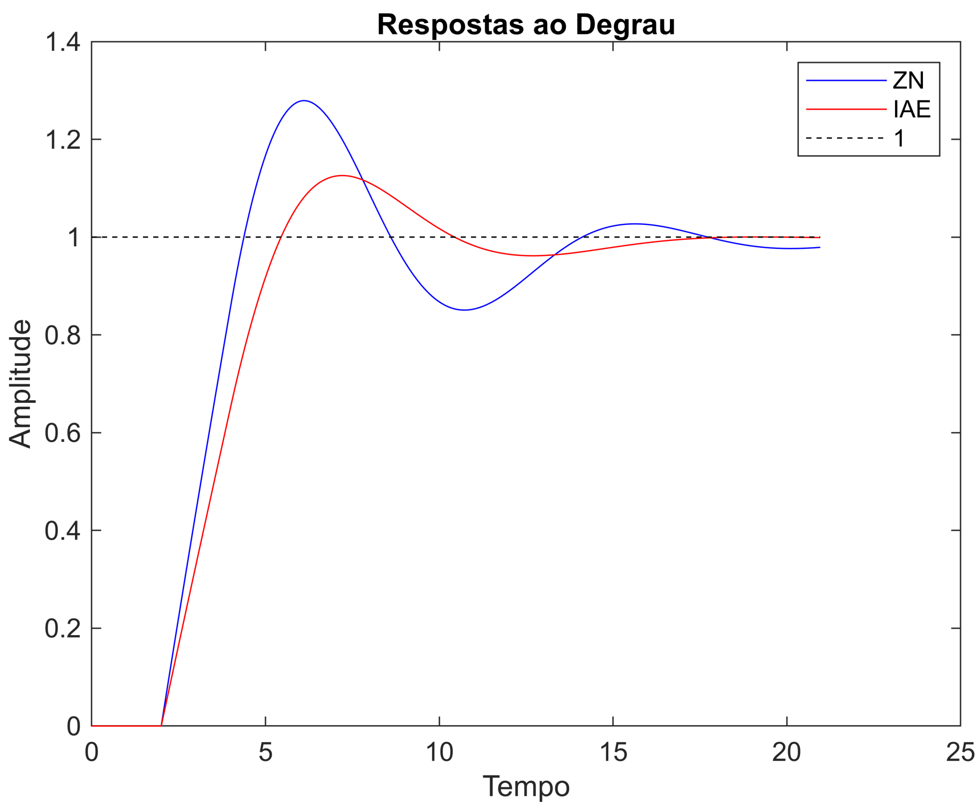
```
disp(['ZN: ', num2str(infoZN.SettlingTime), 's'])
```

```
ZN: 20.9664s
```

```
disp(['IAE: ', num2str(infoIAE.SettlingTime), 's'])
```

```
IAE: 15.1515s
```

```
% Configurações do gráfico
xlabel('Tempo');
ylabel('Amplitude');
title('Respostas ao Degrau');
legend('show');
```



Podemos confirmar que as curvas obtidas estão de fato corretas ao compará-las com o método pidtuning:

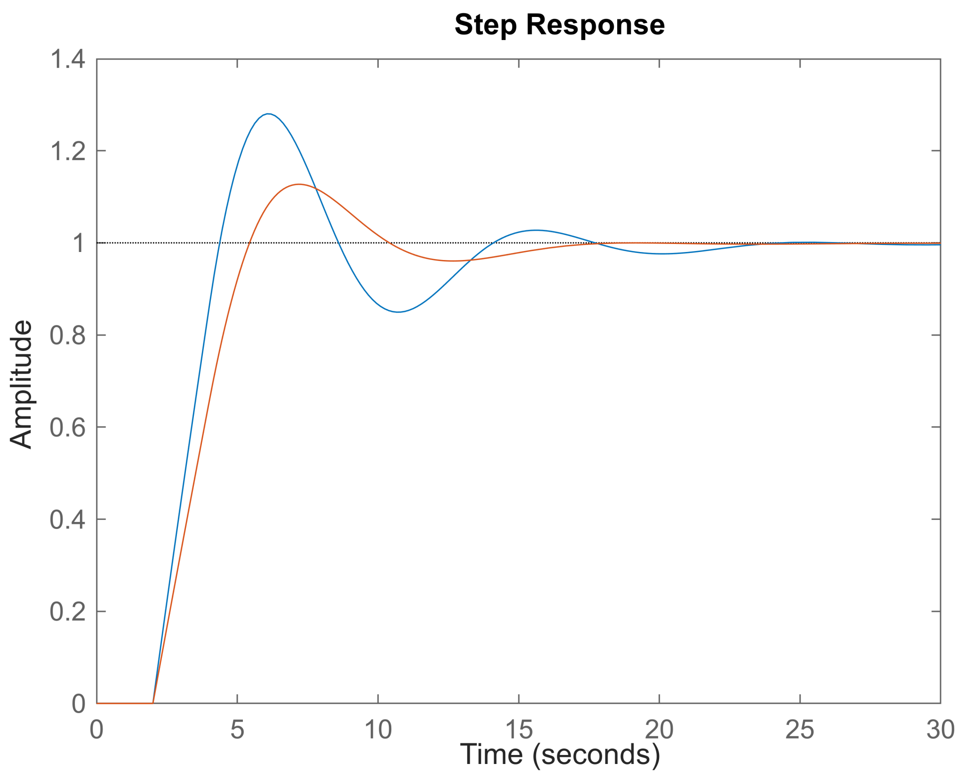
```
g = tf(0.7, [5 1], 'InputDelay' ,2)
[cpizn, iaeZN] = pidtuning(g, 'method', 'zie', 'type', 'PI')
display(iaeZN)
```

```
iaeZN = 4.9302
```

```
m1 = feedback(cpizn*g, 1)
[cpiiae, iaeOtimo] = pidtuning(g, 'method', 'iaeot', 'type', 'PI')
display(iaeOtimo)
```

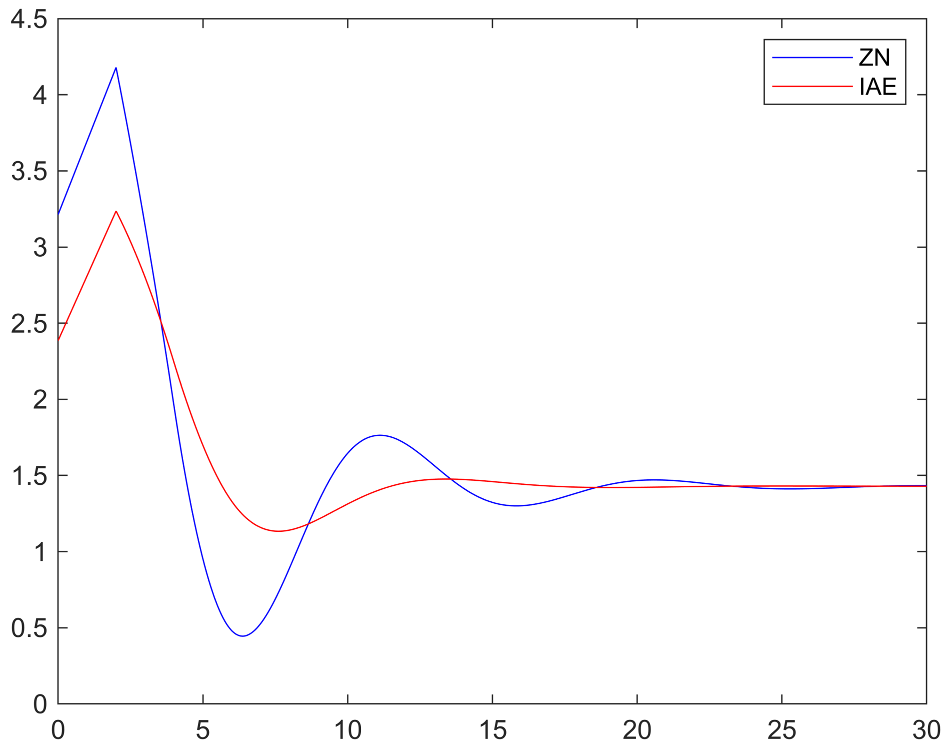
```
iaeOtimo = 4.3153
```

```
m2 = feedback(cpiiae*g, 1)
step(m1, m2)
```



4) Compare o sinal de controle aplicado $u(t)$ por cada um dos controladores para chegar à resposta desejada e os ganhos dos controladores, sabendo que $\frac{U(s)}{R(s)} = \frac{G_c(s)}{1 + G_c(s)G(s)}$

```
uZN = (cpizn)/(1+cpizn*g)
uIAE = (cpiiae)/(1+cpiiae*g)
time = 0:0.01:30
[y1, t] = step(uZN, time)
[y2, ~] = step(uIAE, time)
figure
plot(t, y1, 'DisplayName', "ZN", "Color", "blue")
hold on
plot(t, y2, 'DisplayName', "IAE", "Color", "red")
hold off

legend('show');
```




```
[~, ~, k]=tf2zp(cell2mat(controladorZN.Numerator),
cell2mat(controladorZN.Denominator))
[~, ~, k2]=tf2zp(cell2mat(controladorIAE.Numerator),
cell2mat(controladorIAE.Denominator))
disp(['Ganho do ZN: ', num2str(k)])
```

```
Ganho do ZN: 3.21
```

```
disp(['Ganho do IAE ótimo: ', num2str(k2)])
```

```
Ganho do IAE ótimo: 2.38
```

5) Projete um controlador PI via método de síntese direta usando o modelo de referência $T(s) = \frac{1}{\lambda s + 1}$, escolhendo λ para resultar em um controlador PI com IAE similar ao obtido via IAE ótimo. Explique então como obter o IAE mínimo variando λ

Com o atraso, temos:

$$T(s) = \frac{e^{-2s}}{\lambda s + 1}$$

Pelo método da síntese direta, podemos obter o controlador $G_c(s) = K_c\left(1 + \frac{1}{\tau_i * s}\right)$, com $K_c = \frac{1}{K}\left(\frac{\tau}{\theta + \lambda}\right)$ e

$\tau_i = \tau$

Sabemos que $K = 0.7;\ \theta = 2;\ \tau = 5$

Logo,

$$K_c = \frac{1}{0.7}\left(\frac{5}{2+\lambda}\right) = \frac{5}{1.4+0.7\lambda}$$

$$G_c(s) = K_c\left(1+\frac{1}{\tau_i * s}\right) = \frac{5}{1.4+0.7\lambda} + \frac{5}{5s*(1.4+0.7\lambda)} = \frac{25s+5}{7s+3.5\lambda s}$$

Tendo em vista que o controlador IAE ótimo que obtivemos é:

`13.35 s + 2.38`

`--------------`

`5.61 s`

`Então, o lambda que queremos escolher para aproximar os dois controladores é tal que`

$$\frac{25s+5}{3.5s*(2+\lambda)} = \frac{13.35s+2.38}{5.61s}$$

Um valor aproximado de lambda que faz com que os controladores sem aproximem é 1.04

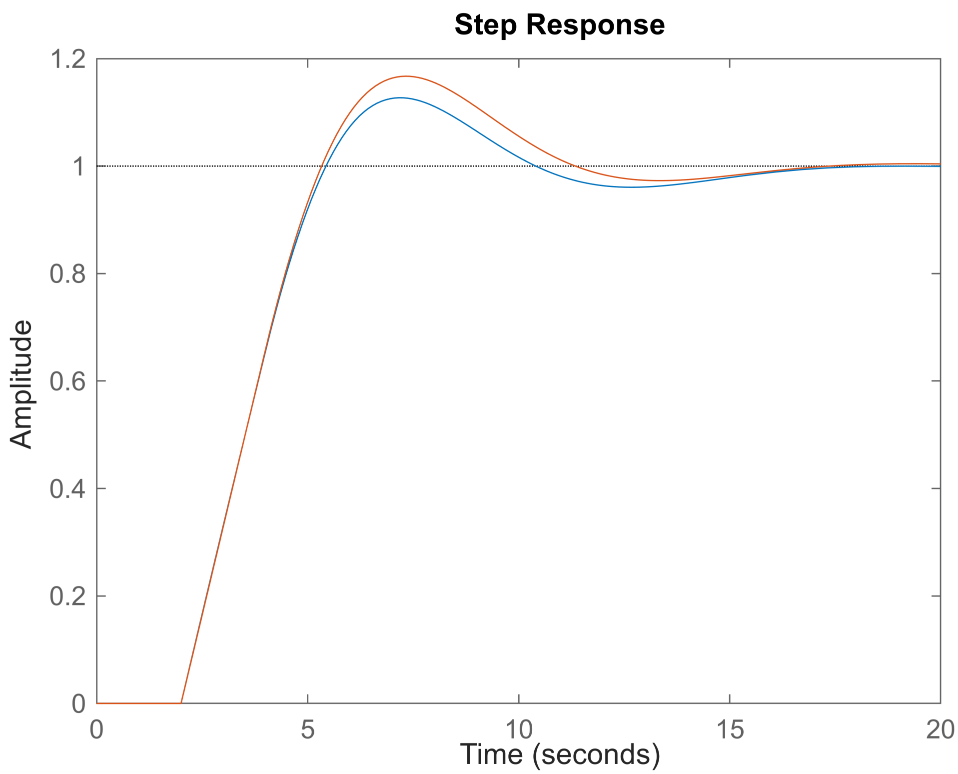
```
lambda = 1.04
Kc = 5/(0.7*(2+lambda))
Gc=Kc*(1+1/(5*s))
```

```
Gc =

  11.75 s + 2.35
  --------------
       5 s

Continuous-time transfer function.
Model Properties
```

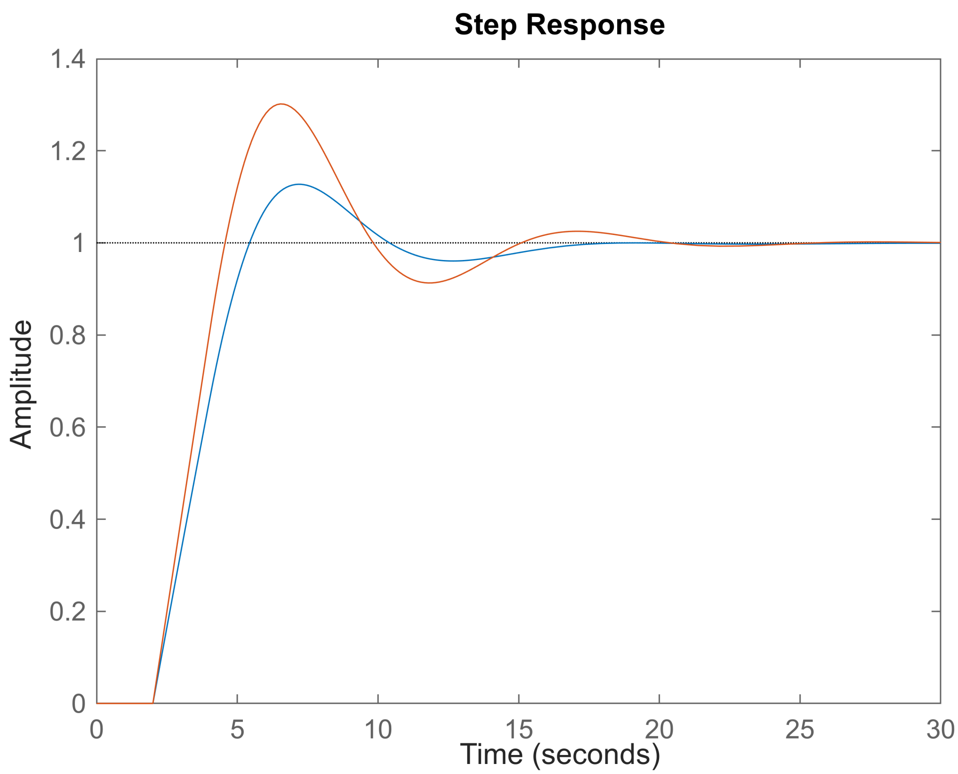
```
controladorIAE
```

```
controladorIAE =

  13.35 s + 2.38
  --------------
      5.61 s

Continuous-time transfer function.
Model Properties
```

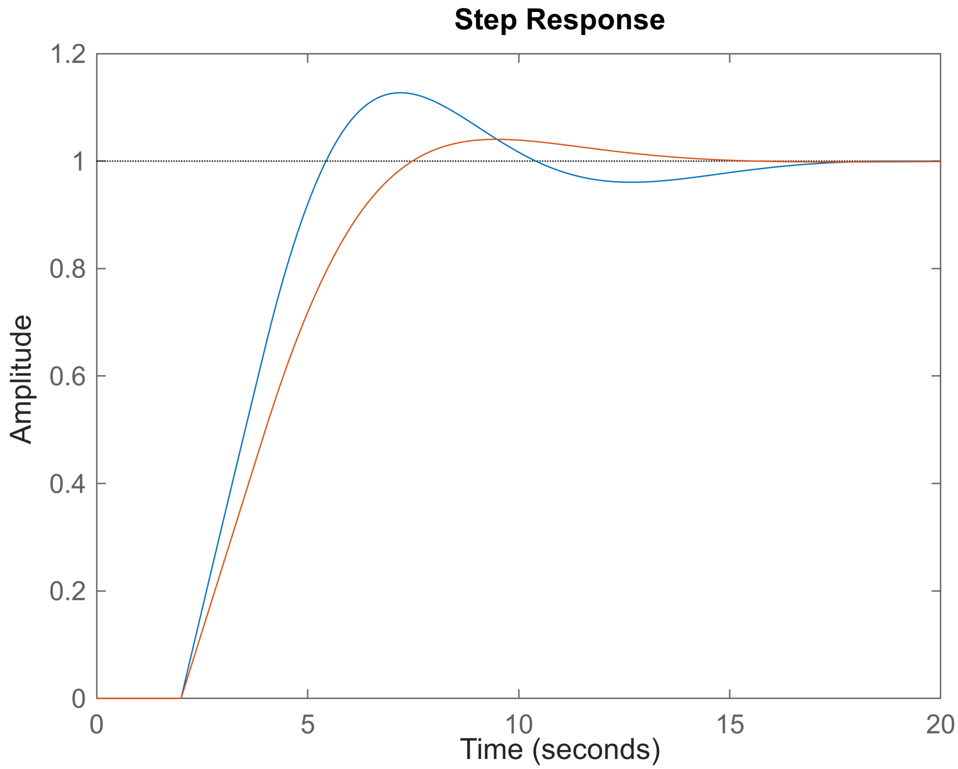
```
Csintese = feedback(Gc*g, 1)

step(m2, Csintese)
```




```
lambda = 0.5
Kc = 5/(0.7*(2+lambda))
Gc=Kc*(1+1/(5*s))
Csintese = feedback(Gc*g, 1)
%lambda=0.5
step(m2, Csintese)
```




```
lambda = 2
Kc = 5/(0.7*(2+lambda))
Gc=Kc*(1+1/(5*s))
Csintese = feedback(Gc*g, 1)
%lambda=2
step(m2, Csintese)
```


Step Response
Amplitude
Time (seconds)
1.2
1
0.8
0.6
0.4
0.2
0
0
5
10
15
20