Sistemas Realimentados

Ep5 - Comparação dos métodos: Síntese Direta, Ziegler-Nichols e IAE Ótimo

Alunos: André Thomaz Fabris e Cayque Pires Monteiro

Seja a FT: $G(s) = \dfrac{4e^{-5s}}{25s+1}$

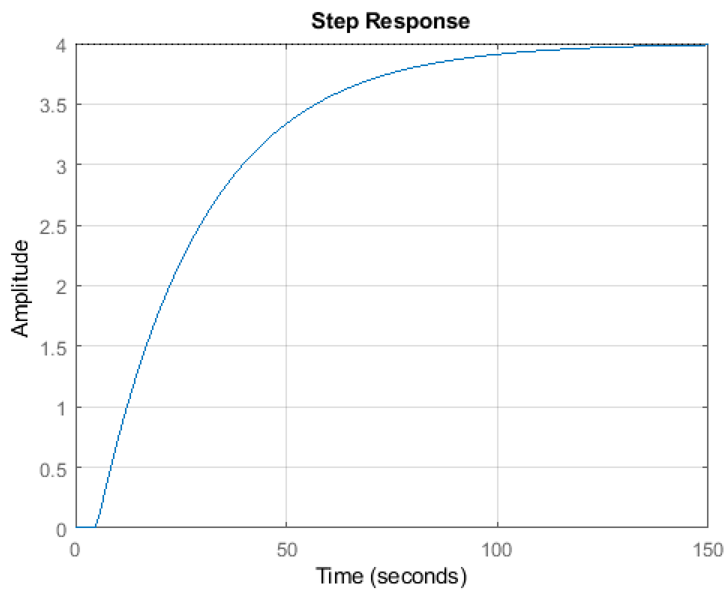
Analisando a Função de Transferência:

```
warning off
G = tf(4,[25 1]);
G.InputDelay = 5
```

```
G =
 
                 4
  exp(-5*s) * --------
              25 s + 1
 
Continuous-time transfer function.
```

A função de transferência G nos fornece constante de tempo τ = 25 s e tempo morto $\theta =$ 5 s. Além disso, é possível observar um ganho K = 4.

```
figure
step(G)
grid on
```



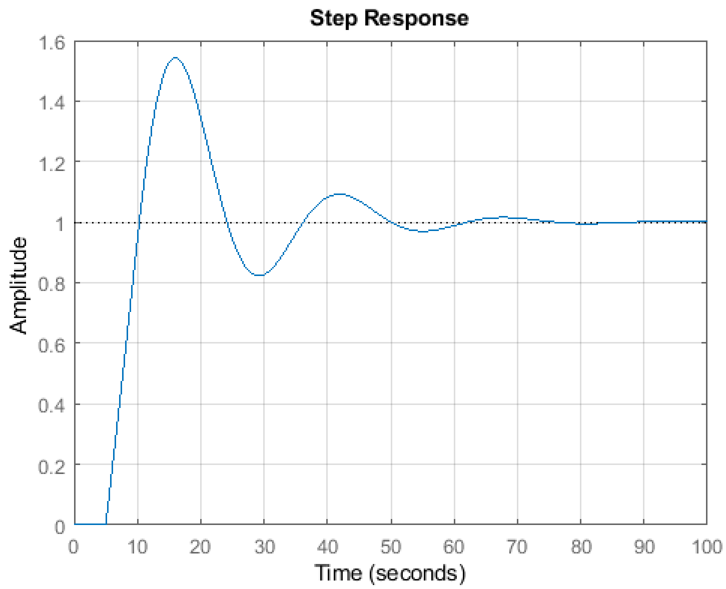

```
L = stepinfo(G);
Tipo = {'FT'};
UP = L.Overshoot;
ts = L.SettlingTime;
tr = L.RiseTime;
Tabela_FT = table (Tipo,UP,ts,tr)
```

Tabela_FT = 1×4 table

	Tipo	UP	ts	tr
1	'FT'	0	102.8061	54.9304

1) Projete o controlador PI via método de Ziegler-Nichols

```
cpi_zie = pidtuning(G,'method','zie','type','PI');
m1 = feedback(cpi_zie*G,1);
step(m1)
grid on
```



O controlador solicitado PI segue a seguinte estrutura:

$$C = \mathrm{Kp} + \frac{\mathrm{Ki}}{s}$$

Aplicando o método de Ziegler-Nichols para a sintonia do controlador PI, temos que:

$$\mathrm{Kp} = \frac{0,9\tau}{K\theta} = \frac{0,9*25}{4*5} = \frac{22,5}{20} = 1,125$$

Para acharmos Ki, primeiramente encontramos o valor de Ti, já que $\mathrm{Ki} = \frac{1}{\mathrm{Ti}}$, assim:

$$\mathrm{Ti} = 3,33\theta = 3,33*5 \cong 16,65$$

$$\mathrm{Ki} = \frac{1}{\mathrm{Ti}} = \frac{1}{16,65} \cong 0,061$$

Logo, com os valores de Kp e Ki da resposta ao degrau plotada, é possível confirmar os cálculos. Sendo os valores da figura mostrados abaixo:

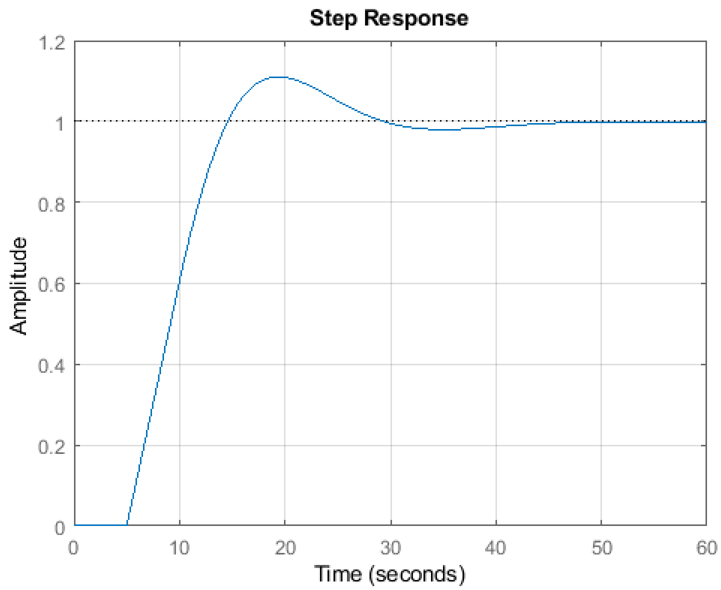
Kp = 1,13 e Ki = 0,0676

Este método funciona melhor para sistemas de primeira ordem e pouco oscilatórios.

2) Projete o controlador PI via método do IAE ótimo

```
cpi_iae = pidtuning(G,'method','iaeot','type','PI');
m2 = feedback(cpi_iae*G,1);
step(m2)
```

```
grid on
```

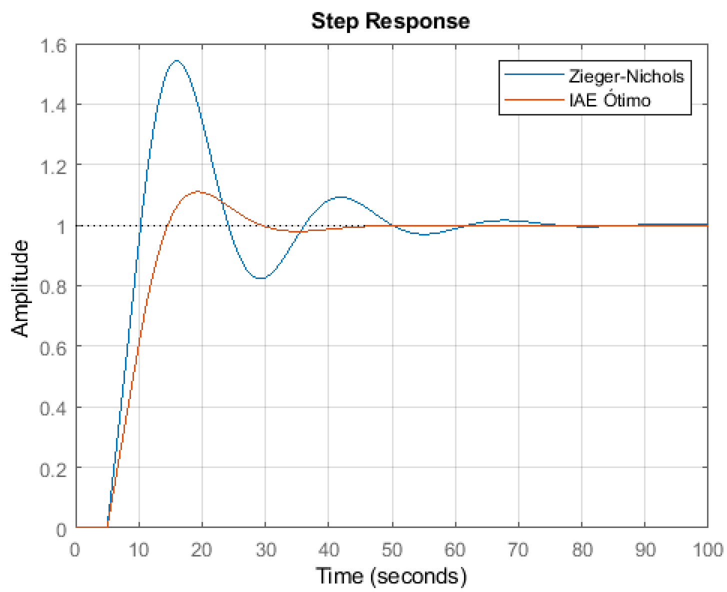


O método do IAE ótimo tenta reduzir ao máximo o tempo de estabelecimento e sobreelevação, visto que esses dois parâmetros são os que mais influenciam no cálculo do IAE. Os ganhos obtidos para esse controlador são mostrados abaixo:

Kp =0,758 e Ki = 0,029

3) Compare as duas respostas em termos de sobreelevação, tempo de estabelecimento e IAE.

```
step(m1,m2)
legend('Zieger-Nichols','IAE Ótimo')
grid on
```




```
%Criando tabela para mostrar os valores de sobreelevação, tempo de estabelecimento e tempo de subida aplicando o método de Ziegler-Nichols
L = stepinfo(m1);
Tipo = {'Ziegler-Nichols'};
UP = L.Overshoot;
ts = L.SettlingTime;
tr = L.RiseTime;
Tabela_ZIE = table (Tipo,UP,ts,tr);

%Criando tabela para mostrar os valores de sobreelevação, tempo de estabelecimento e tempo de subida aplicando o método do IAE ótimo
P = stepinfo(m2);
Tipo = {'IAE Otimo'};
UP = P.Overshoot;
ts = P.SettlingTime;
tr = P.RiseTime;
Tabela_IAE = table (Tipo,UP,ts,tr);

%Juntando as duas tabelas e comparando os resultados
Tabela_Unica = vertcat(Tabela_ZIE,Tabela_IAE)
```

Tabela_Unica = 2×4 table

	Tipo	UP	ts	tr
1	'Ziegler-Nichols'	54.2160	58.4493	4.2321
2	'IAE Otimo'	11.0052	35.7253	7.1955

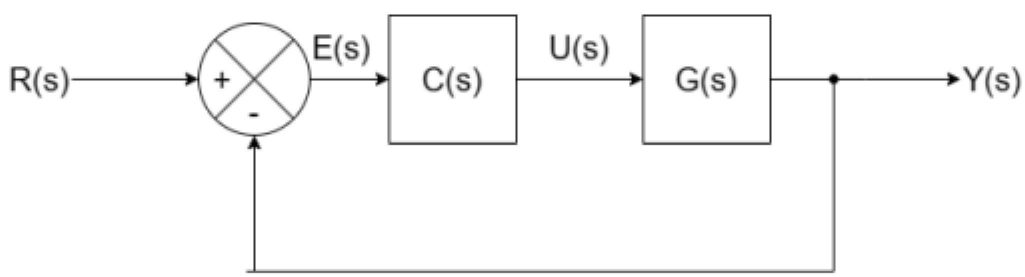
```
%Valor de IAE para o método de Ziegler-Nichols
[y,t]=step(m1);
t=linspace(0,max(t),200);
y=step(m1,t);
iae_zieger=trapz(t,abs(1-y));

%Valor de IAE para o método do IAE ótimo
[y,t]=step(m2);
t=linspace(0,max(t),200);
y=step(m2,t);
iae_iaeotimo=trapz(t,abs(1-y));

%Tabela comparando os dois resultados
Tabela_IAE = table(iae_zieger,iae_iaeotimo)
```

Tabela_IAE = 1×2 table

	iae_zieger	iae_iaeotimo
1	14.8642	10.4697

4) Compare o sinal de controle aplicado por cada um dos controladores para chegar à resposta desejada e os ganhos dos controladores.

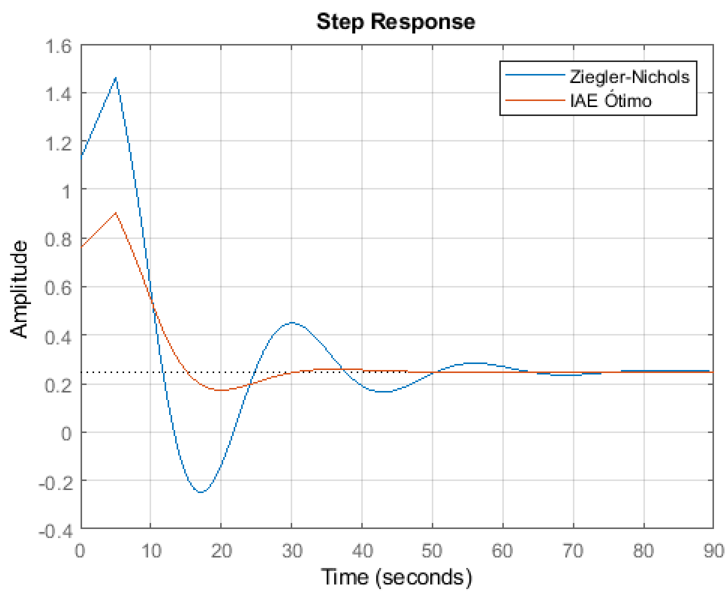
Diagrama de Blocos do sistema controlado em malha fechada



Sinal de Entrada: $R(s) = \frac{1}{s}$

Sinal de controle: $\frac{U(s)}{R(s)} = \frac{C(s)}{1 + C(s)G(s)}$

```
%Para obter a função de transferência e a resposta do sinal de controle dos dois métodos, utilizamos:
h1 = feedback(cpi_zie,G);
h2 = feedback(cpi_iae,G);
step(h1,h2)
legend('Ziegler-Nichols','IAE Ótimo')
grid on
```



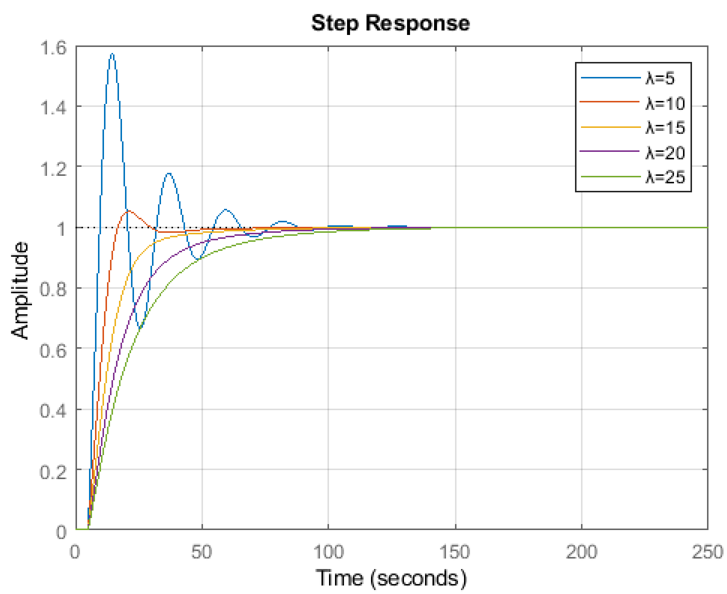
Ambos os controladores começam em seu valor de Kp. Após o tempo morto, atingem o seu máximo e, em seguida, decrescem até se estabilizarem em 0,25, ou seja, 1/K. Isso se deve ao fato da saída ao degrau ser 1, logo, o controlador deve estabilizar em 0,25 para que o sistema com ganho 4 chegue a 1 na resposta ao degrau.

Assim, pelo método de Ziegler-Nichols, temos uma amplitude e oscilação maiores do sinal de controle devido ao maior valor de Kp do controlador projetado. Já pelo método do IAE ótimo, temos uma amplitude e oscilação menores do sinal de controle devido ao menor valor de Kp.

Portanto, podemos notar que o controlador PI via método IAE Ótimo obteve um desempenho melhor.

5) Projete um controlador PI via método de síntese direta usando o modelo de referência $T(s) = \dfrac{1}{\lambda s + 1}$, 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.

```
%Projetando um controlador PI utilizando diferentes valores de
lambda(λ=5,10,15,20,25)
cpi_lambda = pidtuning(G,5);
m3 = feedback(cpi_lambda*G,1);
cpi_lambda = pidtuning(G,10);
m4 = feedback(cpi_lambda*G,1);
cpi_lambda = pidtuning(G,15);
m5 = feedback(cpi_lambda*G,1);
cpi_lambda = pidtuning(G,20);
m6 = feedback(cpi_lambda*G,1);
cpi_lambda = pidtuning(G,25);
m7 = feedback(cpi_lambda*G,1);
step(m3,m4,m5,m6,m7)
legend('λ=5','λ=10','λ=15','λ=20','λ=25')
grid on
```



Pelo gráfico, é visível que quanto menor o valor de lambda, mais oscilatória fica a resposta com uma alta sobreelevação. Já, quanto maior o valor de lambda, mais lenta fica a resposta com um alto tempo de estabelecimento. Nesses dois casos o valor de IAE é alto, tornando o contralador pouco satisfatório.

```
%Calculando os valores de IAE para esses diferentes valores de lambda
[y,t]=step(m3);
t=linspace(0,max(t),200);
y=step(m3,t);
iae_lambda5=trapz(t,abs(1-y));
[y,t]=step(m4);
t=linspace(0,max(t),200);
```

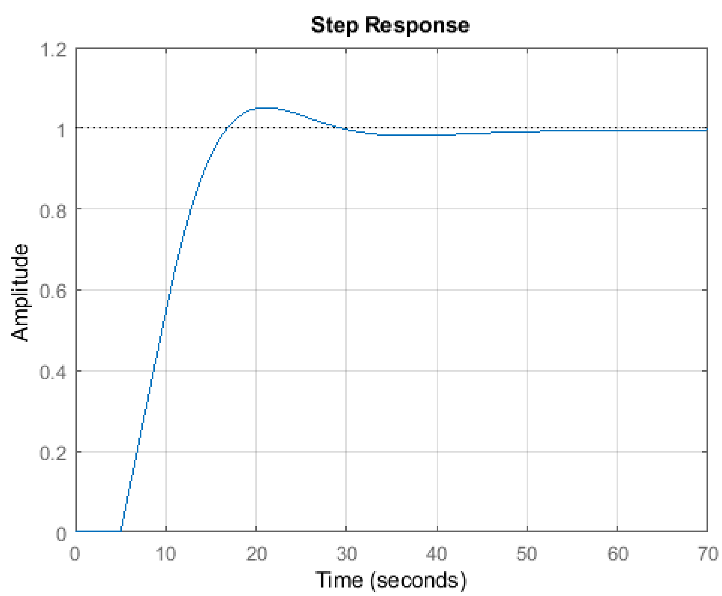
```
y=step(m4,t);
iae_lambda10=trapz(t,abs(1-y));
[y,t]=step(m5);
t=linspace(0,max(t),200);
y=step(m5,t);
iae_lambda15=trapz(t,abs(1-y));
[y,t]=step(m6);
t=linspace(0,max(t),200);
y=step(m6,t);
iae_lambda20=trapz(t,abs(1-y));
[y,t]=step(m7);
t=linspace(0,max(t),200);
y=step(m7,t);
iae_lambda25=trapz(t,abs(1-y));
%Tabela com os diferentes resultados
Tabela = table(iae_lambda5,iae_lambda10,iae_lambda15,iae_lambda20,iae_lambda25)
```

Tabela = 1×5 table

	iae_lambda5	iae_lambda10	iae_lambda15	iae_lambda20	...
1	16.5840	10.6411	14.9380	19.9938	

Comparando os valores de IAE com a variação do lambda, obtivemos com $\lambda = 10$ o menor valor de IAE. Assim, temos as especificações:

```
step(m4)
grid on
```



```
L = stepinfo(m4);
Tipo = {'Lambda'};
UP = L.Overshoot;
ts = L.SettlingTime;
tr = L.RiseTime;
Tabela_IAE = table (Tipo,UP,ts,tr)
```

Tabela_IAE = 1×4 table

	Tipo	UP	ts	tr
1	'Lambda'	5.0806	26.4881	8.4471

A escolha do lambda utilizando a função pidtuning deve ser feita seguindo o critério de ser maior que o tempo morto (lambda>5) e menor que a constante de tempo de malha aberta G (25s). A ideia é ter uma resposta mais rápida e com erro em regime próximo de zero. De fato, a resposta obtida pelo controlador foi aceitável, pois, mesmo gerando uma pequena sobreelevação (5,08%), o IAE obtido de 10,6411 se aproximou do IAE no método do IAE Ótimo (10,4697). O lambda foi testado algumas vezes dentro range estipulado (5 < lambda < 25) e os valores mais satisfatórios foram próximos ao tempo morto, portanto escolhemos 10.