

Sintonia de controladores PID Síntese direta e IMC

Sistemas Realimentados

Projetos baseados em modelo:

Se um modelo dinâmico razoavelmente preciso do processo estiver disponível, é vantajoso basear o projeto do controlador neste modelo. Uma ampla variedade de estratégias de projeto baseadas em modelo está disponível para projetar controladores PID. Consideramos aqui dois importantes métodos de projeto baseados em modelos que são especialmente úteis em controle de processos.

Síntese direta (Direct Synthesis Method)

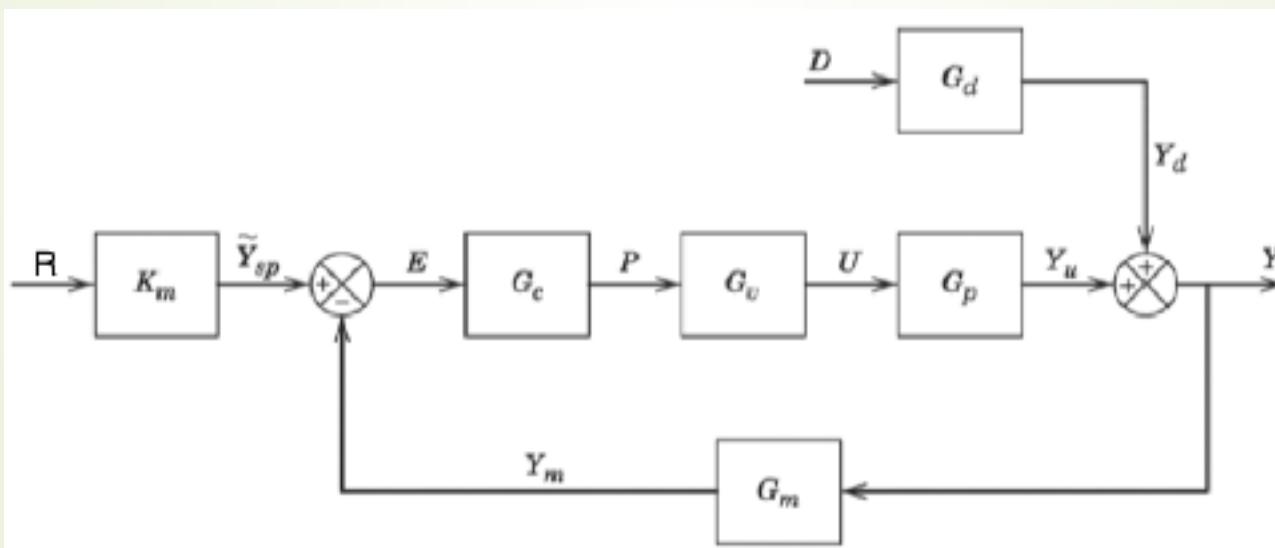
Controle baseado em modelo interno (Internal model control - IMC)

Seborg, Dale E.; Edgar, Thomas F.; Mellicamp, Dukan E ; Doyle III, Francis J. Process Dynamics and Control, John Wiley & Sons, 3a ed. 2011.

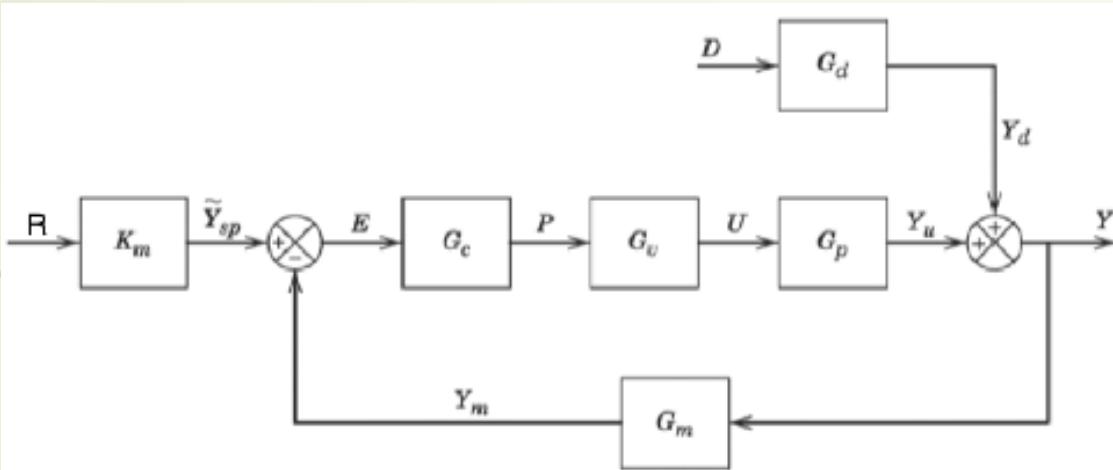
Método da síntese direta

O projeto do controlador é baseado em um modelo de processo e em uma função de transferência de malha fechada que representa o comportamento desejado.

O modelo normalmente esta relacionado a resposta desejada para uma entrada degrau.
O controlador resultante pode ser diferente de um PID.



Método da síntese direta



No sistema mostrado, a resposta para uma entrada degrau é dada por

$$\frac{Y}{R} = \frac{K_m G_c G_v G_p}{1 + G_c G_v G_p G_m} \quad (12-1)$$

Fazendo $G = G_v G_p$ e $G_m = 1$

$$\frac{Y}{R} = \frac{G_c G}{1 + G_c G} \quad (12-2)$$

Método da síntese direta

Rearranjando e resolvendo para G_c , e fazendo $T = \frac{G_c G}{1 + G_c G}$, resulta o controlador ideal

$$G_c = \frac{1}{G} \frac{T}{1 - T} \quad 12-3a$$

A Equação 12-3a não pode ser usada para projeto de controlador, porque a função de transferência em malha fechada $T=Y/R$ não é conhecida a priori. Além disso, é útil distinguir entre o processo real G e o modelo, \tilde{G} , que fornece uma aproximação do comportamento do processo.

Uma equação de projeto prática pode ser derivada substituindo a incógnita G por \tilde{G} , e Y/R pela resposta em malha fechada desejada.

$$G_c = \frac{1}{\tilde{G}} \frac{T}{1 - T} \quad 12-3b$$

Método da síntese direta

$$G_c = \frac{1}{\tilde{G}} \frac{T}{1 - T}$$

A escolha de $T=Y/R$ é um ponto chave do projeto. Trata-se do modelo de referência já discutido, que define a resposta desejada uma uma entrada degrau.

Observa-se também que a função de transferência do controlador contém o termo $1/\tilde{G}$, ou seja, a inversa da FT de malha aberta.

Método da síntese direta

$$G_c = \frac{1}{\tilde{G}} \frac{T}{1 - T} \quad 12-3b$$

O desempenho do controlador dado pela Eq. 12-3b depende fortemente da especificação da função de transferência em malha fechada desejada, $T = T_d$.

No controlador ideal $T_d = 1$, mas isto não é fisicamente possível.

Para sistemas sem tempo morto, uma boa escolha é

$$T_d = \frac{1}{\lambda s + 1} \quad 12-4$$

Com esta escolha, o sistema tem erro nulo ao degrau em malha fechada e constante de tempo λ .

O tempo de estabelecimento será da ordem de 5λ .

Método da síntese direta

Substituindo (12-4) em (12-3b) e resolvendo para G_c vem

$$G_c = \frac{1}{\tilde{G}} \frac{1}{\lambda s} \quad 12.5$$

O termo $\frac{1}{\lambda s}$ é uma ação integral que elimina o erro em regime.

O parâmetro de projeto λ fornece um parâmetro de ajuste do controlador conveniente que pode ser usado para tornar o controlador mais agressivo (λ pequeno) ou menos agressivo (λ grande).

Método da síntese direta

Seja agora o sistema de primeira ordem sem tempo morto, $\tilde{G}(s) = \frac{K}{\tau s + 1}$

Substituindo $G(s)$ em $G_c = \frac{1}{\tilde{G}} \frac{1}{\lambda s}$

resulta um controlador PI, da forma $G_c(s) = K_p(1 + \frac{1}{\tau_i s})$

$$K_p = \frac{\tau}{K\lambda} \quad \tau_i = \tau$$

Observe que $G_c(s)G(s) = \frac{\tau}{K\lambda} \frac{\tau s + 1}{\tau s} \cdot \frac{K}{\tau s + 1} = \frac{1}{\lambda s}$

Logo, $\frac{G_c(s)G(s)}{1 + G_c(s)G(s)} = \frac{1}{\lambda s + 1}$

A relação do ganho integral K_i e a constante de integração τ_i é $\tau_i = 1/K_i$

Método da síntese direta

Se a função de transferência do processo contém um atraso de tempo conhecido θ , uma escolha razoável para a função de transferência em malha fechada desejada é

$$T_d = \frac{e^{-\theta s}}{\tau_c s + 1} \quad (12-6)$$

O termo de atraso em (12-6) é essencial, porque é fisicamente impossível para a variável controlada responder a uma mudança no ponto de ajuste em $t = 0$, antes de $t = \theta$. Se o atraso θ for desconhecido, pode ser substituído por uma estimativa.

Combinando as equações (12-6) e (12-3b), resulta

$$G_c = \frac{1}{\tilde{G}} \frac{e^{-\theta s}}{\tau_c s + 1 - e^{-\theta s}} \quad (12-7)$$

Método da síntese direta

$$G_c = \frac{1}{\tilde{G}} \frac{e^{-\theta s}}{\lambda s + 1 - e^{-\theta s}} \quad (12-7)$$

Embora este controlador não esteja no formato PID padrão, ele é fisicamente realizável.

Mostraremos agora que a equação (12-7) pode ser usada para projetar controladores PID para processos simples.

Para isto, o termo do atraso no denominador de (12-7) é aproximado pela expansão da série de Taylor truncada,

$$e^{-\theta s} \approx 1 - \theta s \quad (12-8)$$

Método da síntese direta

Substituindo

$$e^{-\theta s} \approx 1 - \theta s$$

12-8

em

$$G_c = \frac{1}{\tilde{G}} \frac{e^{-\theta s}}{\lambda s + 1 - e^{-\theta s}}$$

Resulta

$$G_c = \frac{1}{\tilde{G}} \frac{e^{-\theta s}}{(\lambda s + \theta)s}$$

12-9



Método da síntese direta

As aproximações de atraso de tempo são menos precisas quando o atraso de tempo é relativamente grande comparado à constante de tempo dominante do processo. Observe que não é necessário aproximar o termo de atraso no numerador, pois ele é cancelado pelo termo idêntico em \tilde{G} .

Faremos agora o projeto para duas importantes funções de transferência.
Nos dois casos, consideramos $G = \tilde{G}$.

Método da síntese direta

Sistema de primeira ordem+tempo morto

$$\tilde{G}(s) = \frac{Ke^{-\theta s}}{\tau s + 1} \quad 12-10$$

Substituindo esta equação em

$$G_c = \frac{1}{\tilde{G}} \frac{e^{-\theta s}}{(\lambda s + \theta)s}$$

e rearranjando , resulta o controlador PI

$$G_c = K_p(1 + 1/\tau_i s) \quad (12-11)$$

com $K_p = \frac{1}{K} \frac{\tau}{\theta + \lambda} \quad \tau_i = \tau$

Método da síntese direta

$$K_p = \frac{1}{K} \frac{\tau}{\theta + \lambda} \quad \tau_i = \tau$$

Observe que o ganho K_p é inversamente proporcional ao ganho K da planta.

Se o produto KK_p é constante, as características de estabilidade em malha fechada não mudam.

Também é razoável ter-se $\tau_i = \tau$, pois processos lentos têm grandes valores de τ , e assim τ_i também deve ser grande para uma resposta satisfatória.

À medida que λ diminui, K_p aumenta, porque uma resposta mais rápida requer uma ação de controle mais agressiva e, portanto, um valor maior de K_p .

O ganho K_p torna-se ilimitado quando o tempo morto θ e λ tendem a zero.

Método da síntese direta

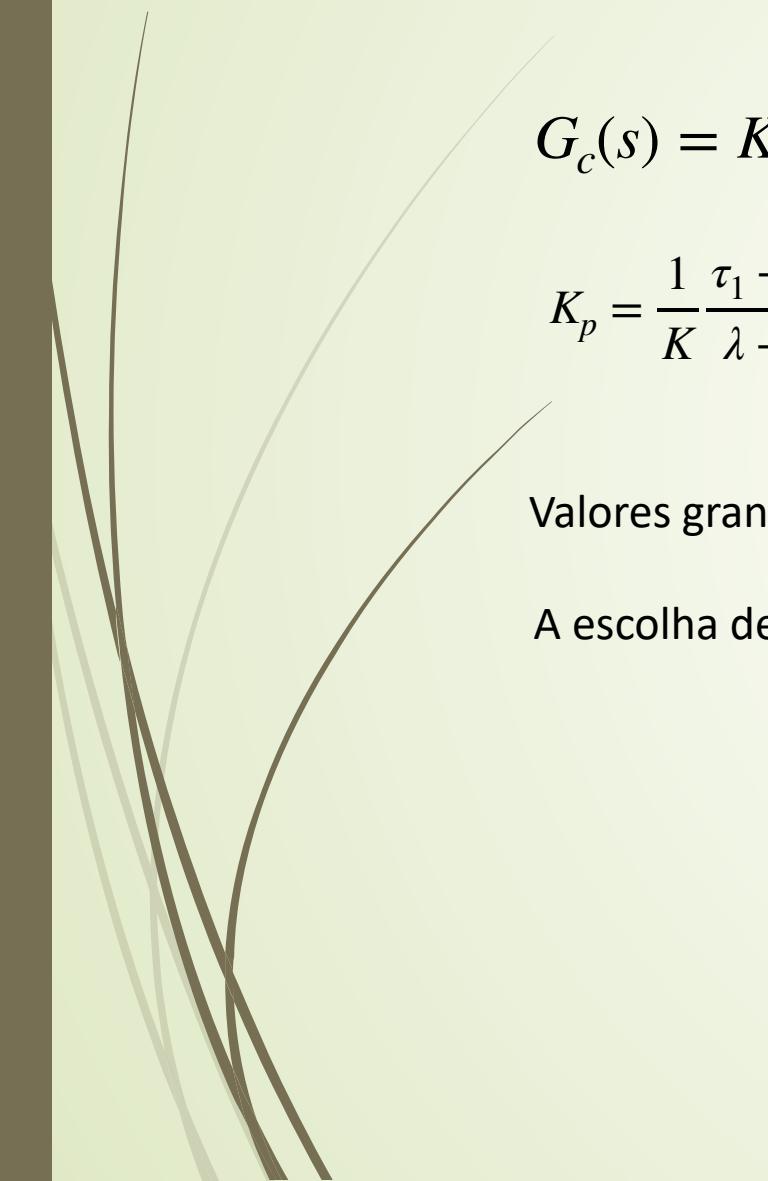
Seja agora o sistema de segunda ordem mais tempo morto

$$\tilde{G}(s) = \frac{Ke^{-\theta s}}{(\tau_1 s + 1)(\tau_2 s + 1)} \quad 12-12$$

Usando o mesmo procedimento, chega-se neste caso ao controlador PID

$$G_c(s) = K_p \left(1 + \frac{1}{\tau_i s} + \tau_d s \right) \quad 12-13 \quad \text{Sendo } \tau_d = K_d$$

$$K_p = \frac{1}{K} \frac{\tau_1 + \tau_2}{\lambda + \theta} \quad \tau_i = \frac{1}{K_i} = \tau_1 + \tau_2 \quad \tau_d = K_d = \frac{\tau_1 \tau_2}{\tau_1 + \tau_2} \quad 12-14$$



Método da síntese direta

$$G_c(s) = K_p \left(1 + \frac{1}{\tau_i s} + \tau_d s \right)$$

$$K_p = \frac{1}{K} \frac{\tau_1 + \tau_2}{\lambda + \theta}$$

Valores grandes de θ reduzem o valor do ganho K_p , mas não afetam τ_i, τ_d .

A escolha de λ afeta diretamente o ganho K_p .

Método da síntese direta

Diferentes modelos de processos podem ser considerados na síntese direta.

O controlador resultante depende deste modelo e do modelo de referência que define o comportamento de malha fechada.

Em cada um dos modelos de processo $G(s)$ apresentado na tabela a seguir, o controlador PID resultante $C(s)$ é tal que

$$\frac{C(s)G(s)}{C(s)G(s)} = \frac{1}{\lambda s + 1}$$

Sendo que λ nesta formulação corresponde ao sistema em malha fechada especificado.

Método da síntese direta para outros modelos

Tabela 6. Sintonia do PID segundo Rivera, Morari e Skogestad

Modelo do Processo	K _P	T _I	T _D
$\frac{K}{\tau s + 1}$	$\frac{\tau}{K \times \lambda}$	τ	-
$\frac{K}{(\tau_1 s + 1)(\tau_2 s + 1)}$	$\frac{(\tau_1 + \tau_2)}{K \times \lambda}$	$(\tau_1 + \tau_2)$	$\frac{\tau_1 \times \tau_2}{(\tau_1 + \tau_2)}$
$\frac{K}{\tau^2 s^2 + 2\xi\tau \times s + 1}$	$\frac{2\xi\tau}{K \times \lambda}$	$2\xi\tau$	$\frac{\tau}{2\xi}$
$\frac{K}{s}$	$\frac{1}{K \times \lambda}$	-	-
$\frac{K}{s(\tau s + 1)}$	$\frac{1}{K \times \lambda}$	-	τ

Nesta tabela usa-se λ equivale ao τ_c .

Método da síntese direta para FOPTD usando outra aproximação

Tabela 7. Sintonia do PID segundo Rivera, Morari e Skogestad para processos com tempo morto

Controlador	K_P	T_I	T_D	Sugestão para o Desempenho
PID	$\frac{2\tau + \theta}{K \times (2\lambda + \theta)}$	$\tau + \left(\frac{\theta}{2}\right)$	$\frac{\tau \times \theta}{(2\tau + \theta)}$	$\frac{\lambda}{\theta} > 0.8$
PI	$\frac{(2\tau + \theta)}{K \times 2\lambda}$	$\tau + \left(\frac{\theta}{2}\right)$	-	$\frac{\lambda}{\theta} > 1.7$

Sintonia PI anterior para FOPTD:

$$K_p = \frac{1}{K} \frac{\tau}{\theta + \lambda} \quad \tau_i = \tau$$

Sugestão: comparar as duas.

Síntese direta via função do Matlab

A função pidtuning executa os cálculos da síntese direta para modelos foptd, ordem 1 e ordem 2 (com polos reais ou complexos).

%	Model G Methods	Types	Parameters
%	foptd lambda	PI, PID	lambda
%	order 1 lambda	PI	lambda
%	order 2 lambda	PID	lambda
%			

O método lambda é o método de síntese direta ou IMC, sendo lambda a cte de MF fornecida

Alguns exemplos.

Projeto de um controlador PID para sistema de ordem 2 (g_2) com $\lambda = 2$

[C, iae]=pidtuning(g_2 , 'method', 'lambda', 'type', 'PID', 'param', 2)

Projeto de um controlador PI para FOPTD (g_1) com $\lambda = 0.5$

[C, iae]=pidtuning(g_1 , 0.5)

Síntese direta usando modelo de referência de ordem 2

Nos projetos usando o modelo de referência de ordem um, normalmente a escolha da constante de tempo de malha fechada λ afeta apenas o ganho proporcional.

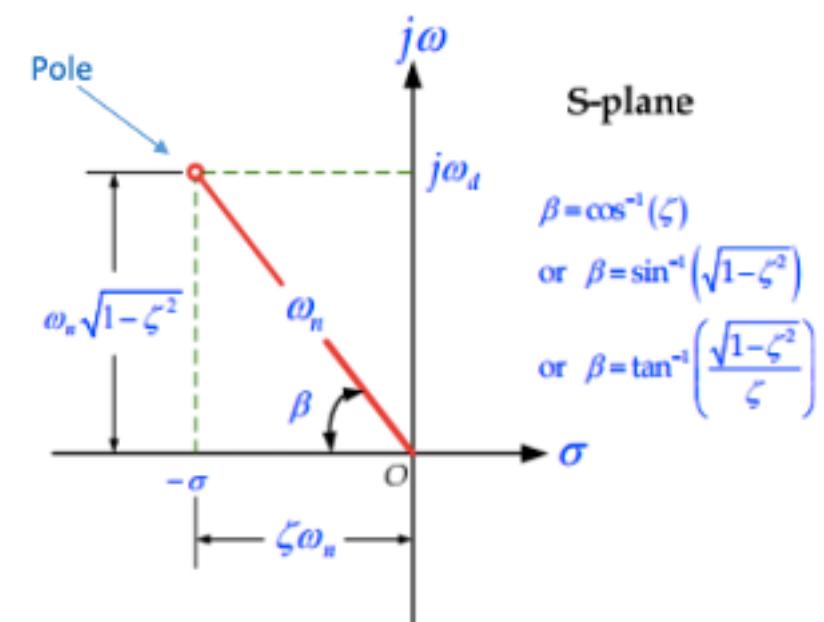
Uma forma de tornar o projeto mais flexível, aumentando os graus deliberado, é escolher um modelo de referência de ordem 2:

$$M(s) \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

$$P_{1,2} = \sigma \pm j\omega_d = -\zeta\omega_n \pm j\omega_n\sqrt{1 - \zeta^2}$$

$$t_s = \frac{4}{\zeta\omega_n}$$

$$UP = 100e^{\frac{-\pi\zeta}{\sqrt{1-\zeta^2}}}$$



Síntese direta usando modelo de referência de ordem 2

Considere $P(s) = \frac{K}{\tau s + 1}$ e $C(s) = K_p(1 + \frac{1}{T_i s})$

A FT de malha aberta é $C(s)P(s) = \frac{KK_p(1 + T_i s)}{sT_i(\tau s + 1)}$

Os polos de MF são as raízes de $s^2 + \frac{1 + KK_p}{\tau} s + \frac{KK_p}{T_i \tau} = 0$

Comparando com $s^2 + 2\zeta\omega_n s + \omega_n^2 = 0$

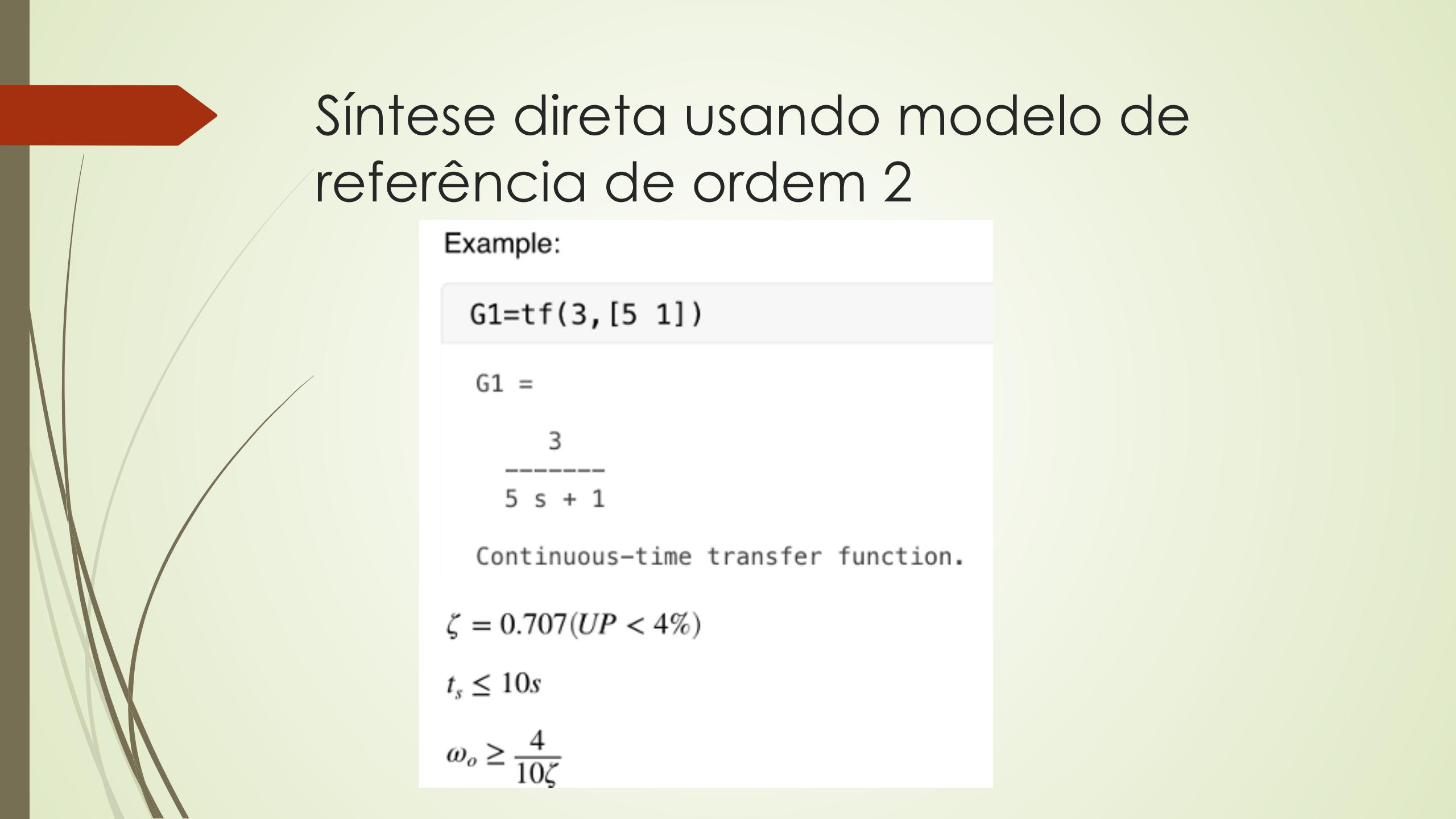
$$K_p = \frac{2\zeta\omega_n\tau - 1}{K}$$
$$T_i = \frac{2\zeta\omega_n\tau - 1}{\omega_n^2\tau}$$



Síntese direta usando modelo de referência de ordem 2

Sistema de malha fechada será:

$$M(s) = \frac{KK_p(1 + T_i s)}{s^2 + \frac{1 + KK_p}{\tau} s + \frac{KK_p}{T_i \tau}} = 0 = \frac{KK_p(1 + T_i s)}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$



Síntese direta usando modelo de referência de ordem 2

Example:

```
G1=tf(3,[5 1])
```

G1 =

$$\frac{3}{5s + 1}$$

Continuous-time transfer function.

$$\zeta = 0.707 (UP < 4\%)$$

$$t_s \leq 10s$$

$$\omega_o \geq \frac{4}{10\zeta}$$

Síntese direta usando modelo de referência de ordem 2

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

```
C =
```

$$\frac{1.767 \ s + 1}{1.767 \ s}$$

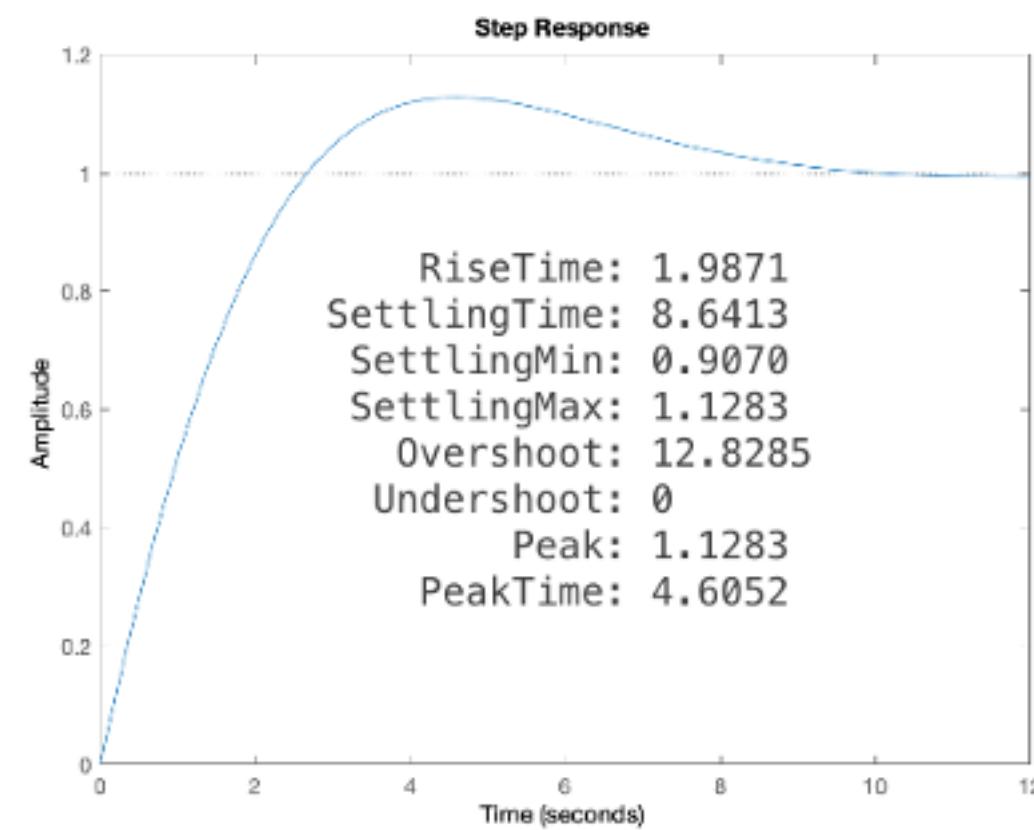
```
Continuous-time transfer function.
```

```
M=feedback(C*G1,1)
```

```
M =
```

$$\frac{5.302 \ s + 3}{8.837 \ s^2 + 7.07 \ s + 3}$$

Síntese direta usando modelo de referência de ordem 2

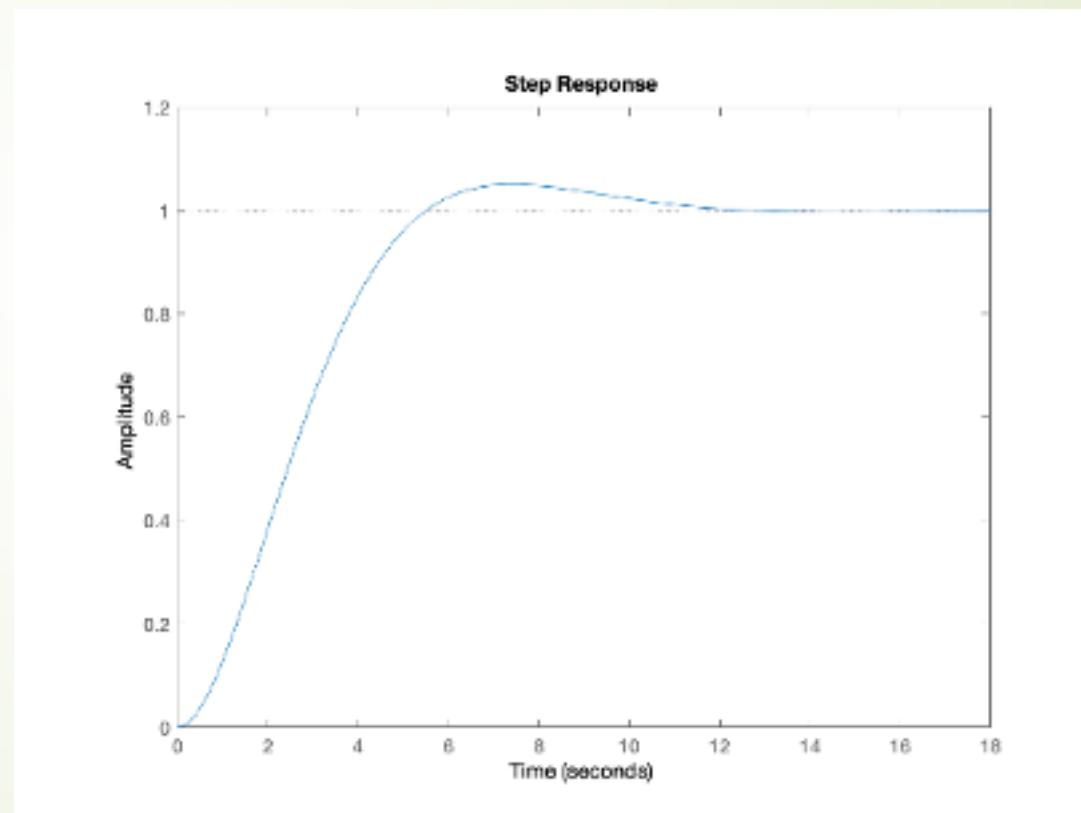


Observe que a sobreelevação não foi atendida.

Síntese direta usando modelo de referência de ordem 2

- ▶ $F = tf(1, [Ti \ 1]);$
- ▶ $step(F * M);$

Um pré-compensador $F(s)$ é utilizado para cancelar o zero introduzido pelo controlador PI



Síntese direta usando modelo de referência de ordem 2

Seja agora o processo com um modelo de ordem 2 dado por $P(s) = \frac{K}{(\tau_1 s + 1)(\tau_2 s + 1)}$

Usando um controlador PI

$$s^3 + \left(\frac{1}{\tau_1} + \frac{1}{\tau_2} \right) s^2 + \frac{1 + KK_p}{\tau_1 \tau_2} s + \frac{KK_p}{T_I \tau_1 \tau_2} = 0$$

Síntese direta usando modelo de referência de ordem 2

Seja agora o processo com um modelo de ordem 2 dado por $P(s) = \frac{K}{(\tau_1 s + 1)(\tau_2 s + 1)}$

Usando um controlador PI

$$s^3 + \left(\frac{1}{\tau_1} + \frac{1}{\tau_2}\right)s^2 + \frac{1+KK_p}{\tau_1\tau_2}s + \frac{KK_p}{T_I\tau_1\tau_2} = 0$$

$$(s + \alpha\omega_0)(s^2 + 2\zeta\omega_0 s + \omega_0^2) = 0$$

$$\omega_0 = \omega_n$$

$$\omega_0 = \frac{\tau_1 + \tau_2}{(\alpha + 2\zeta)\tau_1\tau_2}$$

$$K_p = \frac{(1 + 2\alpha\zeta)\omega_0^2\tau_1\tau_2 - 1}{K}$$

$$T_I = \frac{KK_p}{\alpha\omega_0^3\tau_1\tau_2}$$

Exemplo de projeto usando modelo de ordem 2

$$G(s) = \frac{1}{(s + 1)(10s + 1)}$$

```
[c1,iae]=pidtuning(g,3);
```

Kp=1.83, Ki=0.167, Kd=1.67

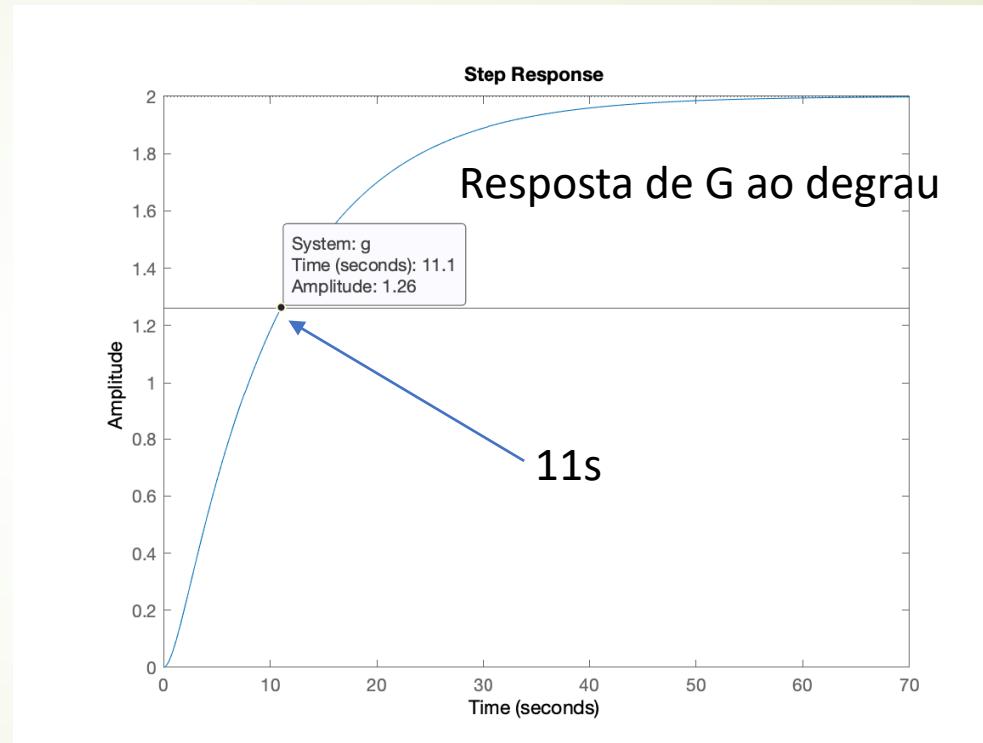
IAE=2.72

UP=5, ts=8

```
[c2,iae]=pidtuning(g,'method','polealloc','type','PI','param',[UP ts])
```

Kp=2.64, Ki=0.205

IAE=2.34



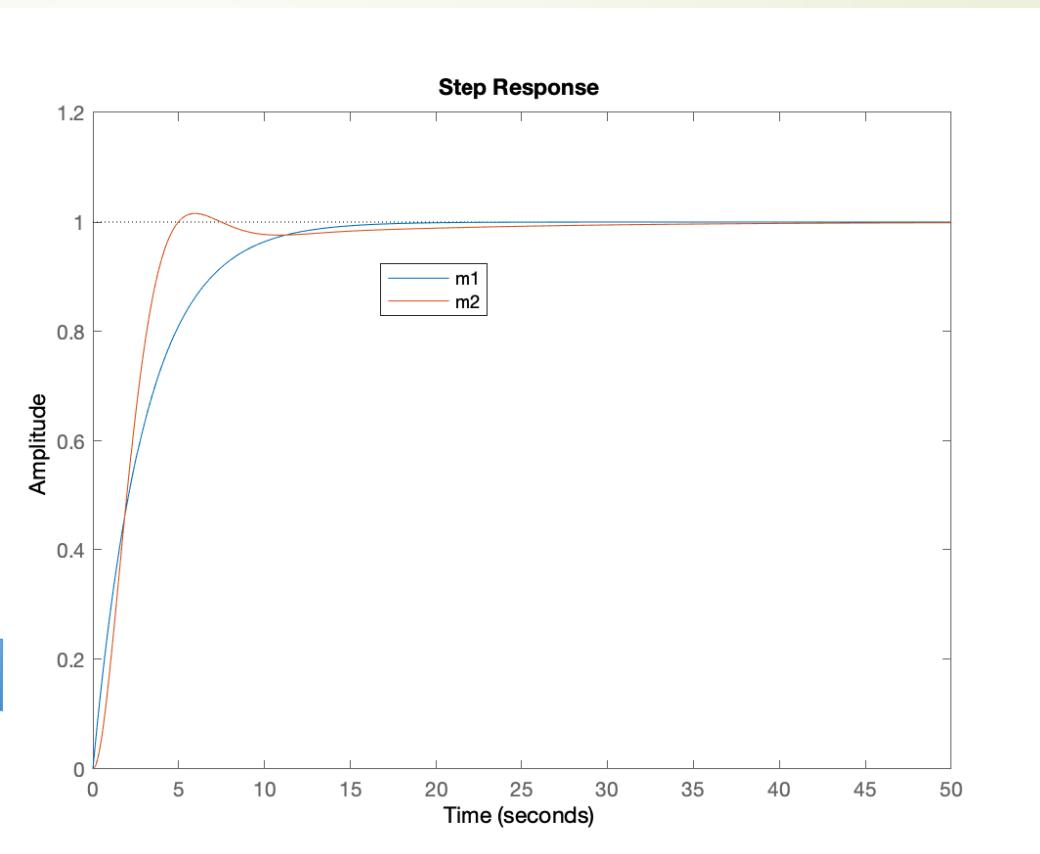
Exemplo de projeto usando modelo de ordem 2

```
>> m1=feedback(c1*g,1);  
>> m2=feedback(c2*g,1);  
>>step(m1,m2);
```

C1 é um PID
C2 é um PI

Resultado melhor com o controlador PI

Pode-se escolher UP e ts!



Exemplo de projeto usando modelo de ordem 2

```
% | order 2 | lambda  
% | order 2 | polealloc  
% |-----| PID | lambda  
% |-----| PI | UP,ts
```

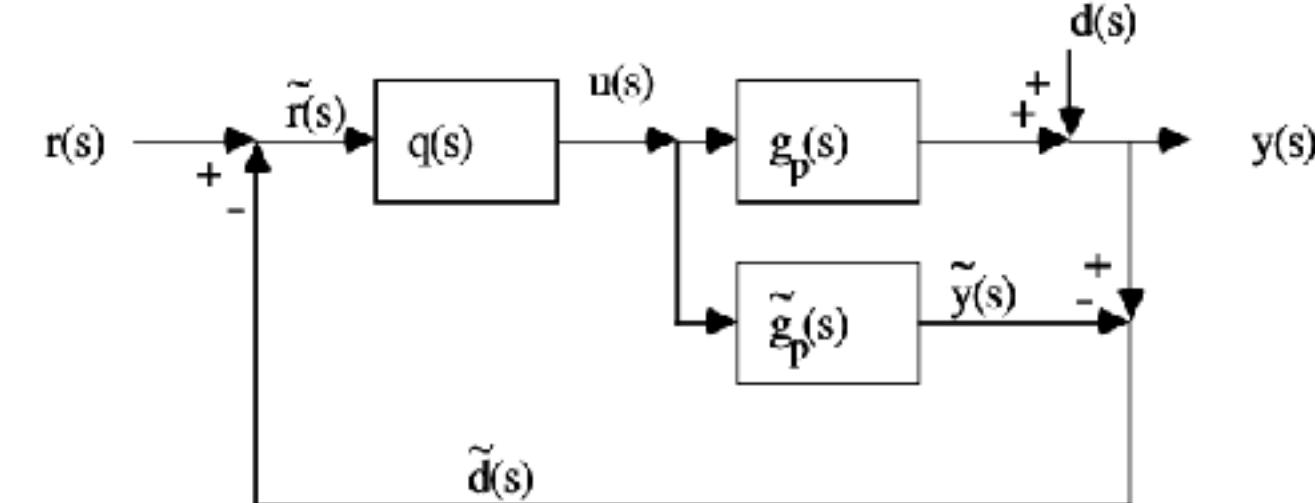
O método lambda é o método de síntese direta ou IMC, sendo lambda a cte de MF fornecida. Ele retorna um PID.

O método polealloc é o método de síntese direta no qual se fornece UP e ts. Ele retorna um PI.

Projeto via IMC

No projeto IMC, o modelo de processo é explicitamente usado no procedimento de projeto do sistema de controle.

A estrutura de feedback padrão utiliza o modelo de processo de forma implícita, ou seja, os parâmetros de ajuste do PID são “ajustados” em um modelo de função de transferência, mas nem sempre é claro como o modelo de processo afeta a decisão de ajuste.





Projeto via IMC

Na formulação IMC, o controlador, $q(s)$, é baseado diretamente na parte “boa” da função de transferência do processo. A formulação IMC geralmente resulta em apenas um parâmetro de sintonia, a constante de tempo de malha fechada (λ , o fator de filtro IMC). Ou seja, o modelo de referência de ordem 1 já discutido.

Os parâmetros de sintonia do PID são então uma função desta constante de tempo de malha fechada.

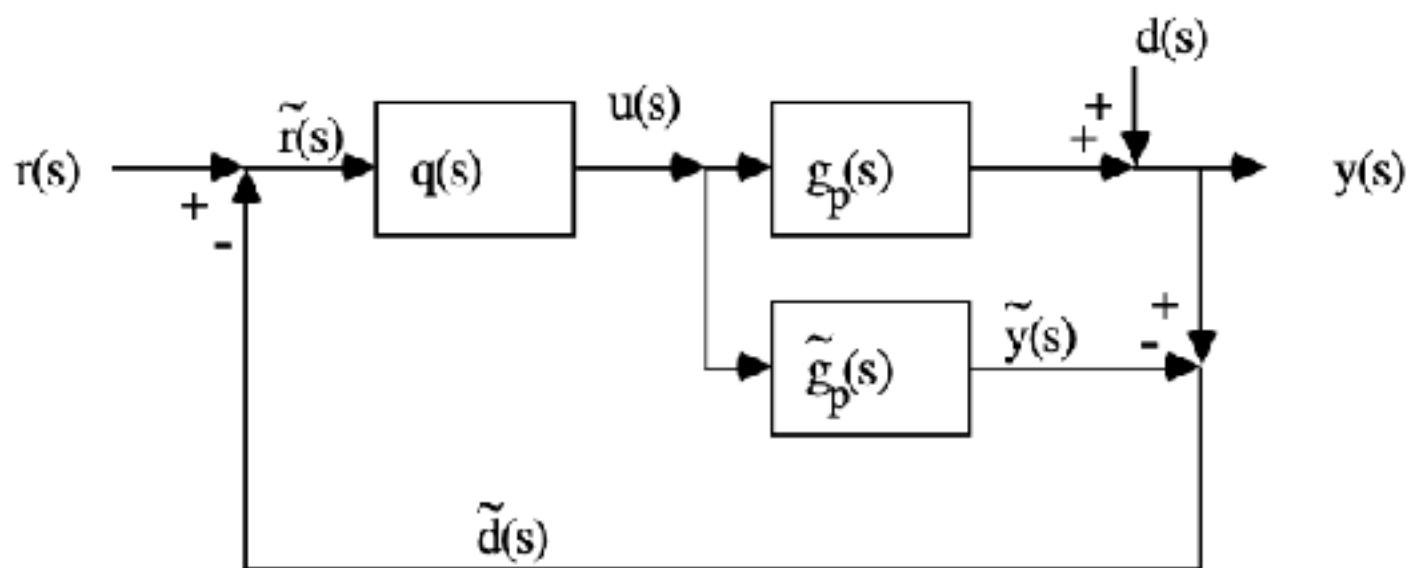
A seleção da constante de tempo em malha fechada está diretamente relacionada à robustez (sensibilidade ao erro do modelo) do sistema em malha fechada.



Projeto via IMC

O controlador PID baseado em IMC aqui discutido não dará os mesmos resultados que a estratégia IMC quando houver atrasos de tempo de processo, porque o procedimento PID baseado em IMC usa uma aproximação de Padé para tempo morto, enquanto a estratégia IMC usa a representação exata para tempo morto.

Estrutura IMC

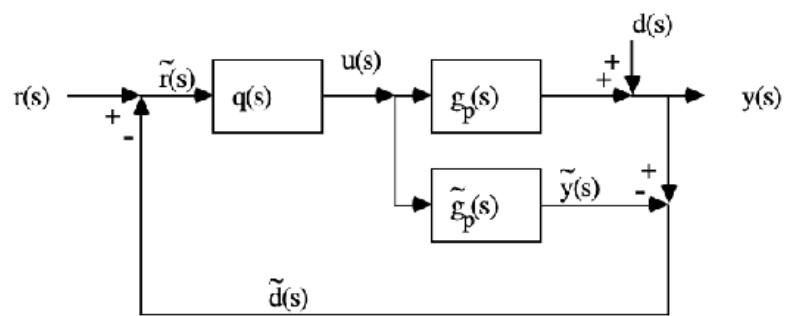


$g_p(s)$ = FT do processo a controlar

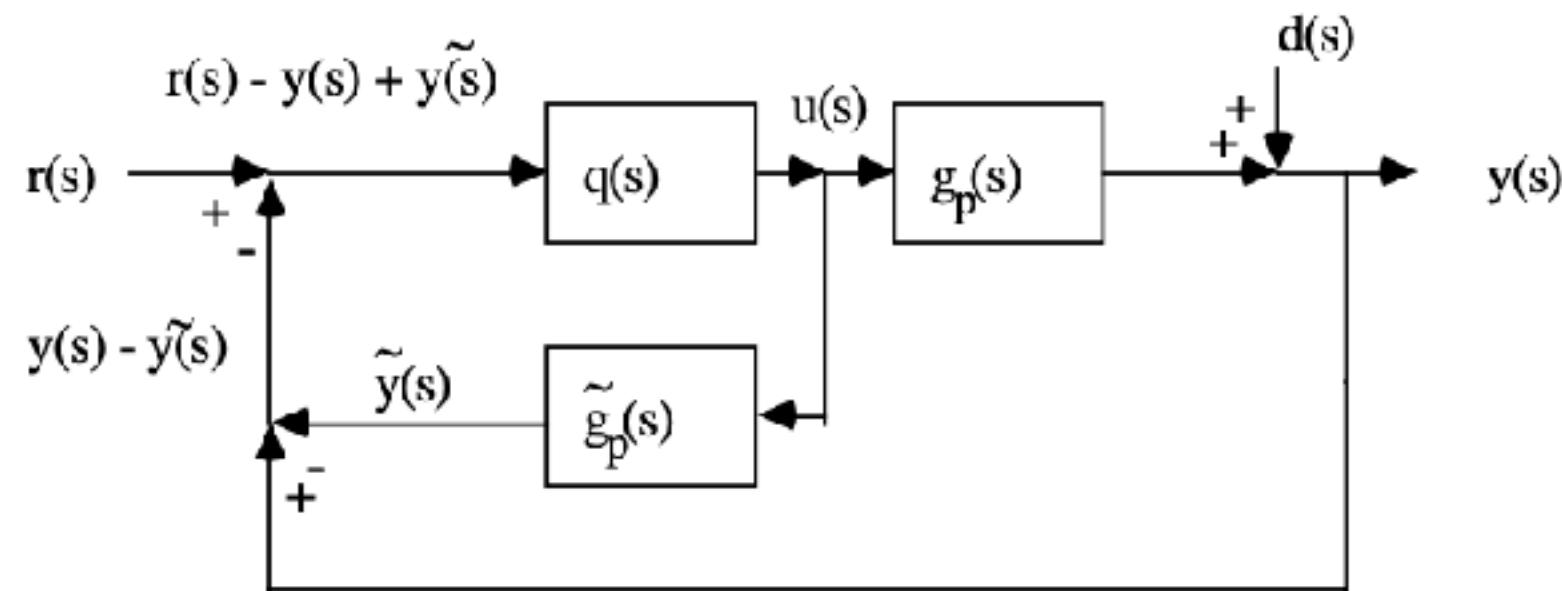
$\tilde{g}_p(s)$ = a parte boa de $g_p(s)$

$q(s)$ = filtro

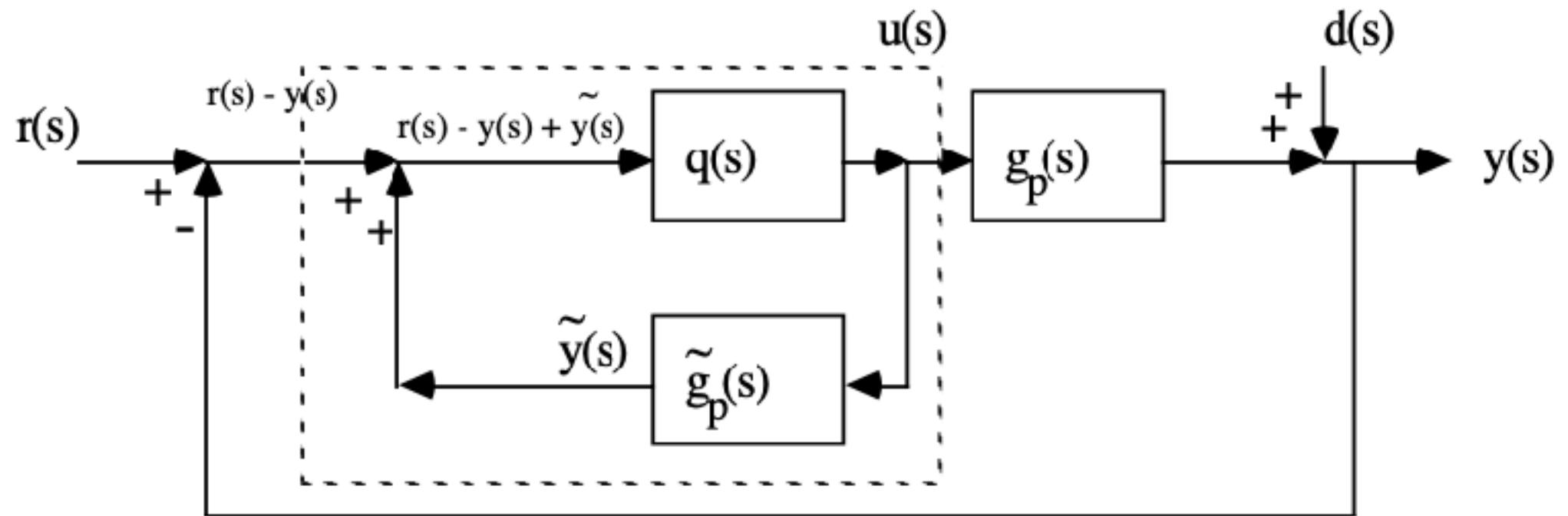
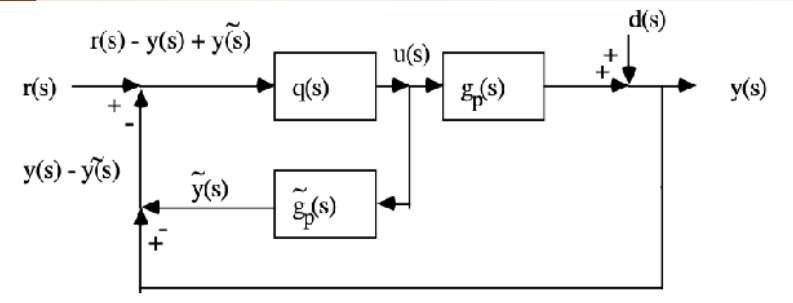
Estrutura IMC



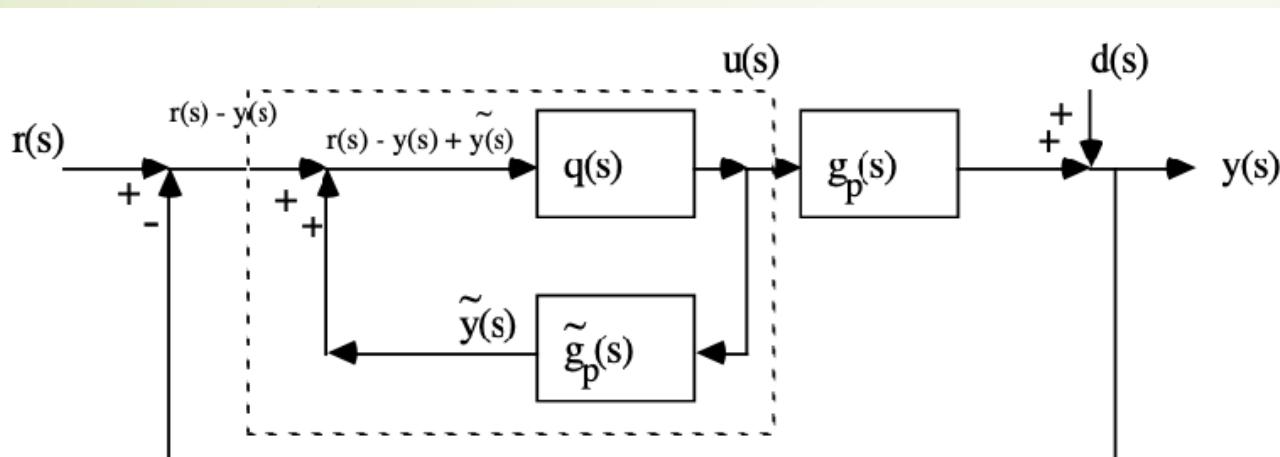
Mudança cosmética



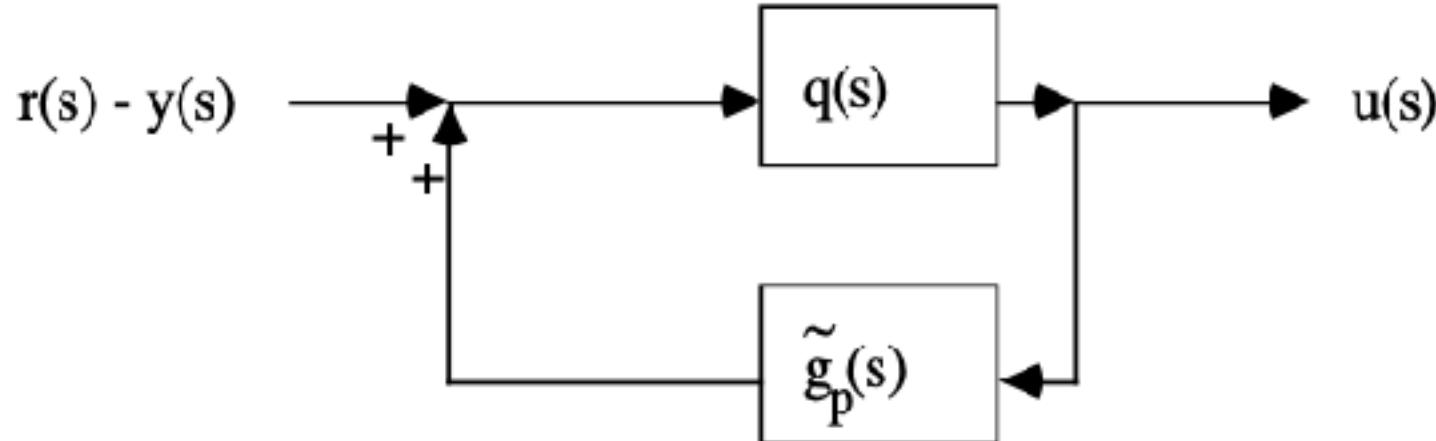
Estrutura IMC



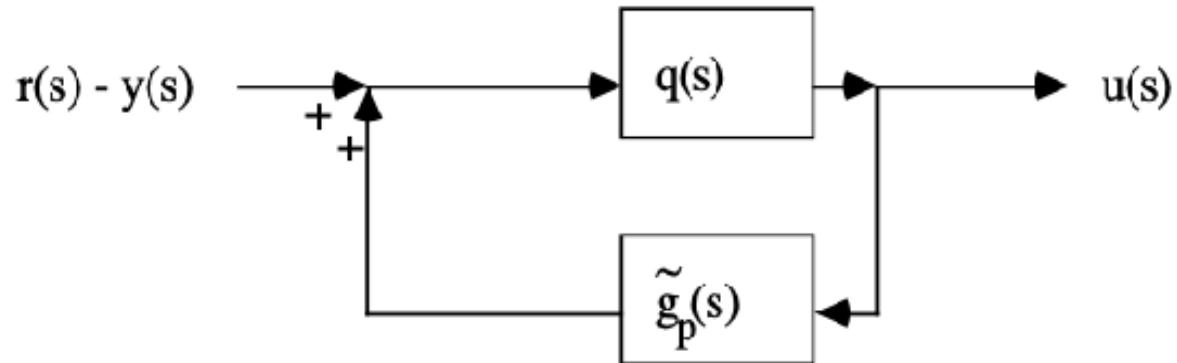
Estrutura IMC



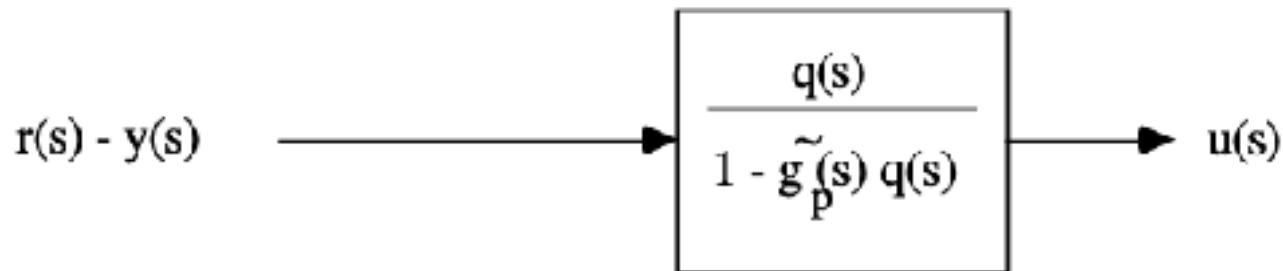
Malha interna



Estrutura IMC



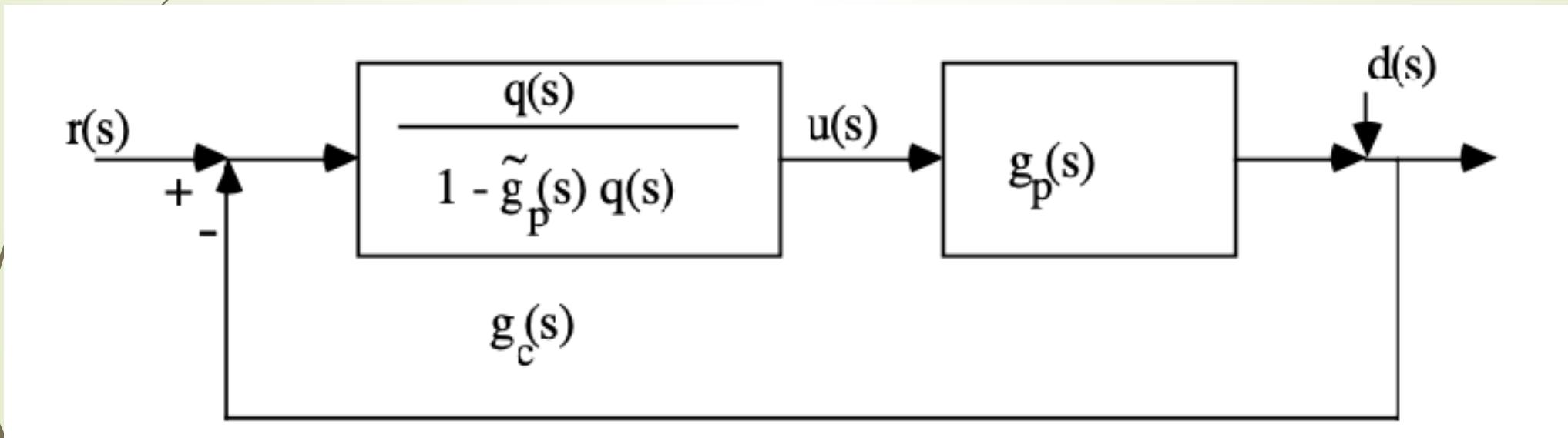
Malha interna



Observe que $r(s) - y(s)$ é simplesmente o termo do erro usado por um controlador de realimentação padrão.

Estrutura IMC

A estrutura IMC pode ser reorganizada para a estrutura de controle de feedback (FBC), conforme mostrado abaixo. Esta reformulação é vantajosa porque descobriremos que muitas vezes resulta um controlador PID quando o procedimento de projeto IMC é usado. Além disso, o diagrama de blocos IMC padrão não pode ser usado para sistemas instáveis, portanto este formato de feedback deve ser usado para esses casos.



Estrutura IMC

Podemos usar o procedimento de projeto IMC para nos ajudar a projetar um controlador de realimentação padrão. O controlador de feedback padrão é uma função do modelo interno $\tilde{g}_p(s)$

$$g_c(s) = \frac{q(s)}{1 - \tilde{g}_p(s)q(s)}$$

$g_c(s)$ é frequentemente um controlador PID.

Uma grande diferença é que o procedimento baseado em IMC, muitas vezes, não exigirá que o controlador seja causal. Além disso, o tempo morto do processo será aproximado utilizando o procedimento de Padé, para se chegar a um controlador tipo PID equivalente. Devido à aproximação de Padé para o tempo morto, o controlador PID baseado em IMC não terá um desempenho tão bom quanto o IMC para processos com atrasos de tempo.

Passos para o projeto baseado no IMC

Passo 1) Encontre a função de transferência do controlador IMC, $q(s)$, que inclui um filtro, $f(s)$, para tornar $q(s)$ semi-próprio ou para dar-lhe ação derivativa (a ordem do numerador de $q(s)$ é uma ordem maior que o denominador de $q(s)$).

Observe que esta é uma grande diferença em relação ao procedimento IMC. Aqui, no procedimento baseado em IMC, podemos permitir que $q(s)$ seja impróprio, a fim de encontrar um controlador PID equivalente. A má notícia é que você deve saber a resposta que está procurando antes de decidir se deseja tornar $q(s)$ própria ou imprópria neste procedimento.

Steps for IMC-based step control

Passo 2) Encontre o controlador por realimentação padrão equivalente ao IMC usando a transformação

$$g_c(s) = \frac{q(s)}{1 - \tilde{g}_p(s)q(s)}$$

e escreva na forma de razão entre dois polinômios.

Steps for IMC-based step control

Passo 3) Mostre os resultados na forma de um controlador PID para obter k_c , τ_I , τ_D . Obtenha a forma padrão equivalente usando a transformação.

Às vezes este procedimento resulta em um controlador PID em série com um termo de atraso (τ_F):

$$g_c(s) = k_c \left[\frac{\tau_I \tau_D s^2 + \tau_I s + 1}{\tau_I s} \right] \left[\frac{1}{\tau_F s + 1} \right]$$

Passo 4) Faça simulações em malha fechada para os casos de modelo perfeito e modelos um pouco diferentes. Escolha o valor desejado de λ como um trade-off entre desempenho e robustez.

Projeto de PID baseado em IMC: Processo de ordem 1

Obtenha o PID-equivalente IMC para o processo de primeira ordem:

$$\tilde{g}_p(s) = \frac{k_p}{\tau_p s + 1}$$

Passo 1. Obtenha a FT do controlador IMC, $q(s)$, o qual inclui um filtro para fazer $q(s)$ semi- própria

$$q(s) = \tilde{q}(s)f(s) = \tilde{g}_{p-}^{-1}(s)f(s) = \frac{\tau_p s + 1}{k_p} \frac{1}{\lambda s + 1}$$

$$q(s) = \frac{1}{k_p} \frac{\tau_p s + 1}{\lambda s + 1} \quad (7-3)$$

Projeto de PID baseado em IMC: Processo de ordem 1

Passo 2. Obtenha o controlador por realimentação padrão usando a transformação

$$g_c(s) = \frac{q(s)}{1 - \tilde{g}_p(s)q(s)} = \frac{\frac{\tau_p s + 1}{k_p(\lambda s + 1)}}{1 - \frac{k_p}{\tau_p s + 1} \frac{\tau_p s + 1}{k_p(\lambda s + 1)}} = \frac{\tau_p s + 1}{k_p \lambda s} \quad (7-4)$$

Lembrando que a função de transferência de PI é

$$g_c(s) = k_c \frac{\tau_I s + 1}{\tau_I s} \quad (7-5)$$

Projeto de PID baseado em IMC: Processo de ordem 1

Passo 3: Rearrange (7.4) para se ajustar a forma de (7.5). Multiplicando (7.4) por τ_p/τ_p

$$g_c(s) = \frac{\tau_p}{k_p\lambda} \frac{\tau_p s + 1}{\tau_p s} \quad (7-6)$$

Comparando com o controlador PI

$$k_c = \frac{\tau_p}{k_p\lambda} \quad (7-7)$$

$$\tau_I = \tau_p$$

$$g_c(s) = k_c \frac{\tau_I s + 1}{\tau_I s}$$

Projeto de PID baseado em IMC: Processo de ordem 1

$$k_c = \frac{\tau_p}{k_p \lambda}$$

$$\tau_I = \tau_p$$

Observações importantes:

- O projeto resultou em um controlador PI
- Não há mais 2 graus de liberdade (K_p, K_I), mas apenas λ
- No projeto apenas o ganho proporcional precisa ser ajustado.
- O ganho integral é feito igual à constante de tempo do processo.
- Os ganhos são os mesmos que os obtidos pelo método de síntese direta.

No Matlab:

```
[C, iae]=pidtuning(G,λ)
```

Projeto de PID baseado em IMC: Processo de ordem 1 mais atraso(FOPTD)

Obtenha o PID-equivalente IMC para o modelo

$$\tilde{g}_p(s) = \frac{k_p e^{-\theta s}}{\tau_p s + 1}$$

Passo 1. Use uma aproximação de primeira ordem de Padé para o tempo morto:

$$e^{-\theta s} \approx \frac{-0.5\theta s + 1}{0.5\theta s + 1}$$

$$\tilde{g}_p(s) = \frac{k_p(-0.5\theta s + 1)}{(\tau_p s + 1)(0.5\theta s + 1)} \quad (7.14))$$

Projeto de PID baseado em IMC: Processo de ordem 1 mais atraso

Obtenha o PID-equivalente IMC para o modelo

Passo 2. Separe a parte "boa" da parte "ruim" da função de transferência

$$\tilde{g_{p-}}(s) = \frac{k_p}{(\tau_p s + 1)(0.5\theta s + 1)} \quad (7.15)$$

$$\tilde{g_{p+}}(s) = -0.5\theta s + 1$$

Projeto de PID baseado em IMC: Processo de ordem 1 mais atraso

Passo 3. Obtenha o controlador ideal

$$\tilde{q}(s) = \frac{(\tau_p s + 1)(0.5\theta s + 1)}{k_p} \quad (7.16)$$

Projeto de PID baseado em IMC: Processo de ordem 1 mais atraso

Passo 4. Adicione o filtro, que esta vez não tornará $q(s)$ própria, mas permitirá obter um controlador PID (que também não é próprio)

$$q(s) = \tilde{q}(s)f(s) = \tilde{g}_{p-1}^{-1}f(s) = \frac{(\tau_p s + 1)(0.5\theta s + 1)}{k_p} \frac{1}{\lambda s + 1} \quad (7.17)$$

Projeto de PID baseado em IMC: Processo de ordem 1 mais atraso

Agora, obtenha o PID equivalente. Lembre que

$$g_c(s) = \frac{q(s)}{1 - \tilde{g}_p(s)q(s)} = \frac{\tilde{q}(s)f(s)}{1 - \tilde{g}_p(s)\tilde{q}(s)f(s)}$$

$$g_c(s) = \frac{\tilde{q}(s)f(s)}{1 - \tilde{g}_{p-}(s)\tilde{g}_{p+}(s)\tilde{g}_{p-}^{-1}(s)f(s)} = \frac{\tilde{q}(s)f(s)}{1 - \tilde{g}_{p+}(s)f(s)} = \left(\frac{1}{k_p}\right) \frac{(\tau_p s + 1)(0.5\theta s + 1)}{(\lambda + 0.5\theta)s}$$

(7.18)

Projeto de PID baseado em IMC: Processo de ordem 1 mais atraso

Podemos expandir o termo do numerador para obter

$$g_c(s) = \left(\frac{1}{k_p}\right) \frac{0.5\tau_p\theta s^2 + (\tau_p + 0.5\theta)s + 1}{(\lambda + 0.5\theta)s} \quad (7.19)$$

Podemos multiplicar (7.19) por $\tau_p + 0.5\theta/\tau_p + 0.5\theta$ para obter os parâmetros do PID

$$k_c = \frac{(\tau_p + 0.5\theta)}{k_p(\lambda + 0.5\theta)}$$

$$\tau_I = \tau_p + 0.5\theta$$

$$\tau_D = \frac{\tau_p\theta}{2\tau_p + \theta}$$

$$g_c(s) = k_c \left[\frac{\tau_I \tau_D s^2 + \tau_I s + 1}{\tau_I s} \right]$$

No Matlab:

```
[C, iae]=pidtuning(G,'method','lambda','type','PID','param',λ)
```



Projeto de PID baseado em IMC: Processo de ordem 1 mais atraso

O procedimento de projeto resultou em um controlador PID, quando o processo é de primeira ordem mais tempo morto.

Lembre-se que uma aproximação de Padé para tempo morto foi usada neste desenvolvimento, o que significa que o fator de filtro não pode ser arbitrariamente pequeno, portanto haverá limitações de desempenho para a estratégia PID baseada em IMC que não ocorrem na estratégia IMC. Rivera et al. (1986) recomendou que $\lambda > 0.8\theta$ devido à incerteza no modelo gerada pela aproximação de Padé.

Projeto de PID baseado em IMC: Processo de ordem 1 mais atraso.

Projeto 2

Neste caso, desconsideramos o tempo morto, mas aumentamos a constante de tempo para ajudar a aproximar o efeito do tempo morto.

$$\tilde{g}_p(s) = \frac{k_p e^{-\theta s}}{\tau_p s + 1} \approx \frac{k_p}{(\tau_p + 0.5\theta)s + 1}$$

Antes fizemos assim:

$$\tilde{g}_p(s) = \frac{k_p e^{-\theta s}}{\tau_p s + 1} \approx \frac{k_p (-0.5\theta s + 1)}{(\tau_p s + 1)(0.5\theta s + 1)}$$

Neste caso, voltamos aos parâmetros de sintonia baseados em IMC para um processo de primeira ordem que já foi derivado.



Projeto de PID baseado em IMC: Processo de ordem 1 mais atraso.

Projeto 2

$$k_c = \frac{\tau_p + 0.5\theta}{k_p \lambda}$$
$$\tau_I = \tau_p + 0.5\theta$$

No Matlab:

```
[C, iae]=pidtuning(G,'method','lambda','type','PI','param',λ)
```

ou

```
[C, iae]=pidtuning(G,λ)
```

Como o tempo morto foi aproximado, Morari and Zafiriou (1989) recomendam $\lambda > 1.7\theta$ para garantir robustez e desempenho.

Projeto de PID baseado em IMC: Modelo de ordem 2

Obtenha o PID-equivalente IMC para o modelo de ordem 2.

$$\tilde{g}_p(s) = \frac{k_p}{(\tau_1 s + 1)(\tau_2 s + 1)}$$

Passo 1. Obtenha a FT do controlador baseado em IMC, $q(s)$, permitindo que $q(s)$ seja imprópria para obter o PID.

$$q(s) = \tilde{q}(s)f(s) = \tilde{g}_p^{-1}f(s) = \frac{(\tau_1 s + 1)(\tau_2 s + 1)}{k_p} \frac{1}{\lambda s + 1} \quad (7-8)$$

Projeto de PID baseado em IMC: Modelo de ordem 2

Passo 2: Obtenha o controlador padrão por realimentação usando a transformação

$$g_c(s) = \frac{q(s)}{1 - \tilde{g}_p(s)q(s)} = \frac{\frac{(\tau_1 s + 1)(\tau_2 s + 1)}{k_p(\lambda s + 1)}}{1 - \frac{k_p}{(\tau_1 s + 1)(\tau_2 s + 1)} \frac{(\tau_1 s + 1)(\tau_2 s + 1)}{k_p(\lambda s + 1)}} = \frac{\tau_1 \tau_2 s^2 + (\tau_1 + \tau_2)s + 1}{k_p \lambda s} \quad (7-9)$$

Compare com a FT do PID

$$g_c(s) = k_c \left[\frac{\tau_I \tau_D s^2 + \tau_I s + 1}{\tau_I s} \right] \quad (7-10)$$

Projeto de PID baseado em IMC: Modelo de ordem 2

Passo 3. Rearrange (7.9) para ficar no formato (7.10). Multiplique (7.9) por $(\tau_1 + \tau_2)/(\tau_1 + \tau_2)$

$$g_c(s) = \left(\frac{\tau_1 + \tau_2}{k_p \lambda} \right) \frac{\tau_1 \tau_2 s^2 + (\tau_1 + \tau_2)s + 1}{(\tau_1 + \tau_2)s} \quad (7-11)$$

Equacionando os termos em (7.10) e (7.11), obtém-se as relações desejadas

$$k_c = \frac{(\tau_1 + \tau_2)}{k_p \lambda}$$

$$\tau_I = \tau_1 + \tau_2$$

$$\tau_D = \frac{\tau_1 \tau_2}{\tau_1 + \tau_2}$$

No Matlab:

```
[C, iae]=pidtuning(G,'method','lambda','type','PID','param',λ)
```



Projeto do PID via IMC: comentários finais

O projeto pode ser feito para modelos de diferentes ordens com ou sem tempo morto.

Quando os modelos vêm de dados, pode-se selecionar a estrutura do modelo que afetará o projeto.

O controlador projetado pode ser PI, PID or PID com filtro de atraso. O modelo usado e o modelo de referência afetam o controlador resultante,

A ação derivativa requer atenção com sinais muito ruidosos.

Método de Ziegler-Nichols

- Publicado em 1942
- Propõe uma sintonia de controlador P, PI, ou PID baseado na resposta ao degrau de um sistema de primeira ordem com tempo morto
- Não permite escolher a resposta desejada (modelo de referência e λ)

Tabela 1. Sintonia via segundo método de Ziegler Nichols

Controlador	K_p	T_I	T_D
P	$\frac{\tau}{K\theta}$	-	-
PI	$\frac{0.9\tau}{K\theta}$	3.33θ	
PID	$\frac{1.2\tau}{K\theta}$	2θ	0.5θ

$$C(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s \right)$$

Projeto de controlador para minimizar o IAE

O IAE é afetado pela sobreelevação e pelo tempo de resposta.

Métodos de otimização podem ser empregados para obter os ganhos dos controladores que minimizam o IAE.

Os controladores possíveis são o PI e o PID.

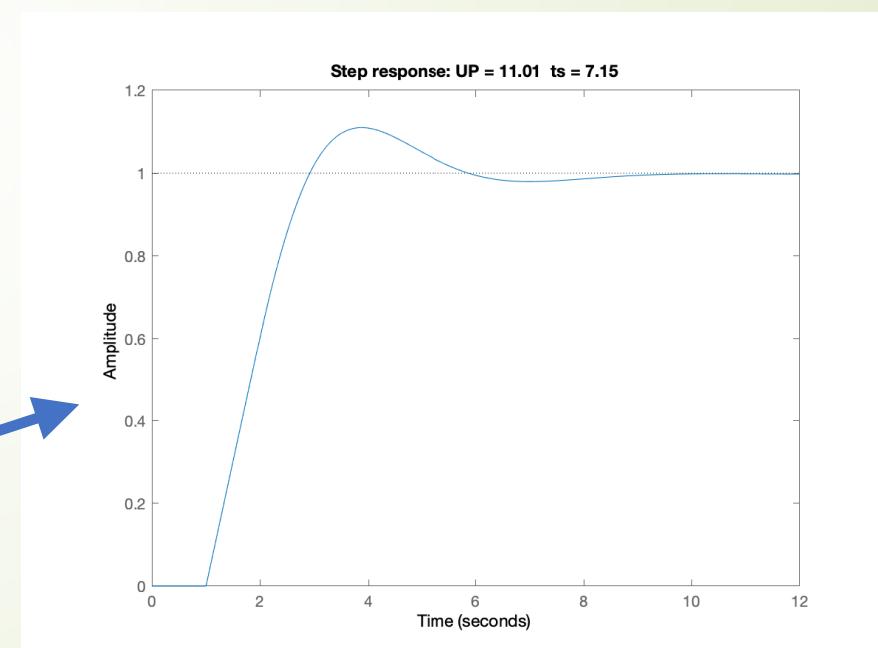
```
[C, iae]=pidtuning(g,'method','iaeot','type','PI')
```

```
[C, iae]=pidtuning(g,'method','iaeot','type','PID')
```

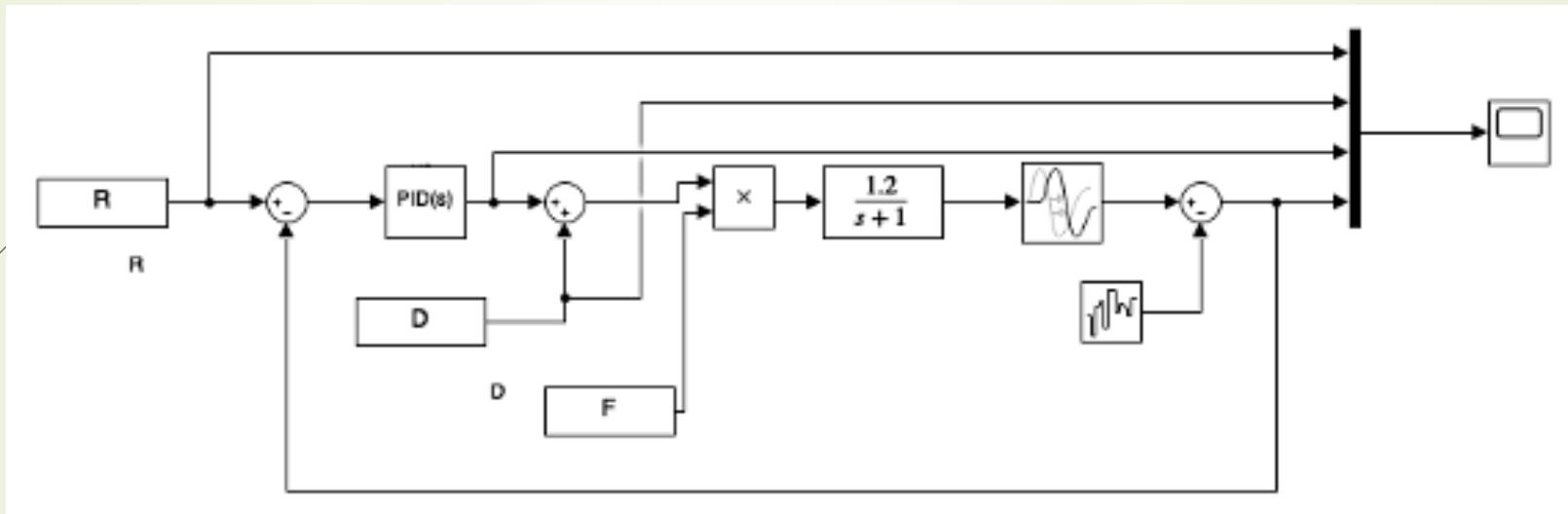
Exemplo:

```
g=tf(1,[5 1],'InputDelay',1)
```

```
pidtuning(g,'method','iaeot','type','PI')
```



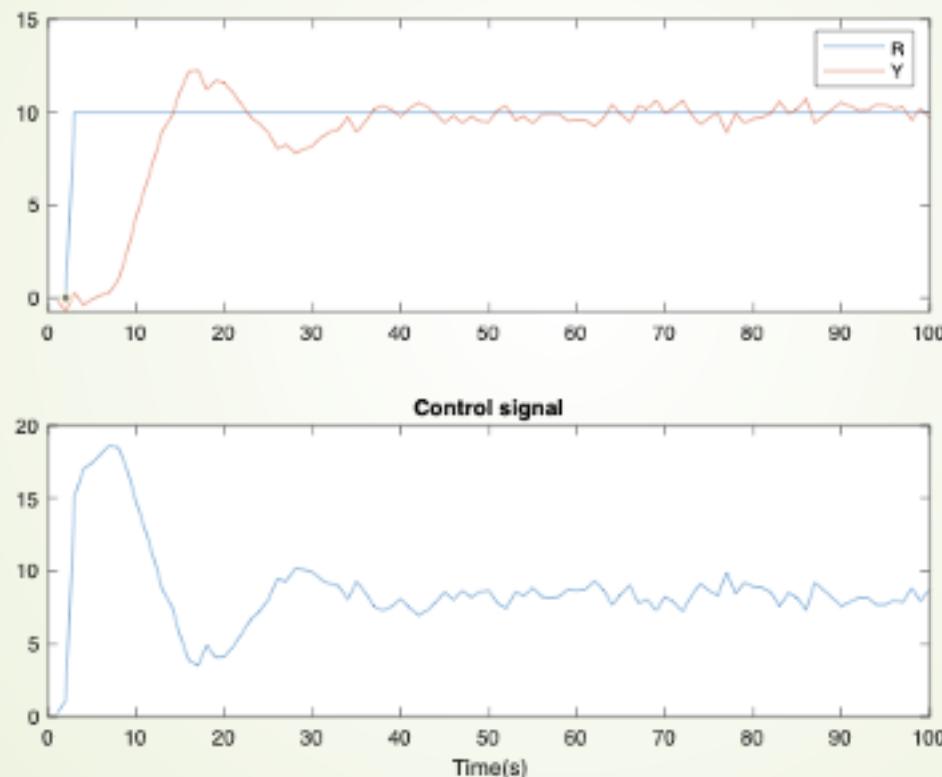
Exemplo de projeto: escolha do modelo de referência



Adiciona-se ruído na saída, para avaliar seu efeito na sintonia.

$$G(s) = \frac{1.2e^{-0.5s}}{s + 1}$$

Exemplo de projeto: escolha do modelo de referência



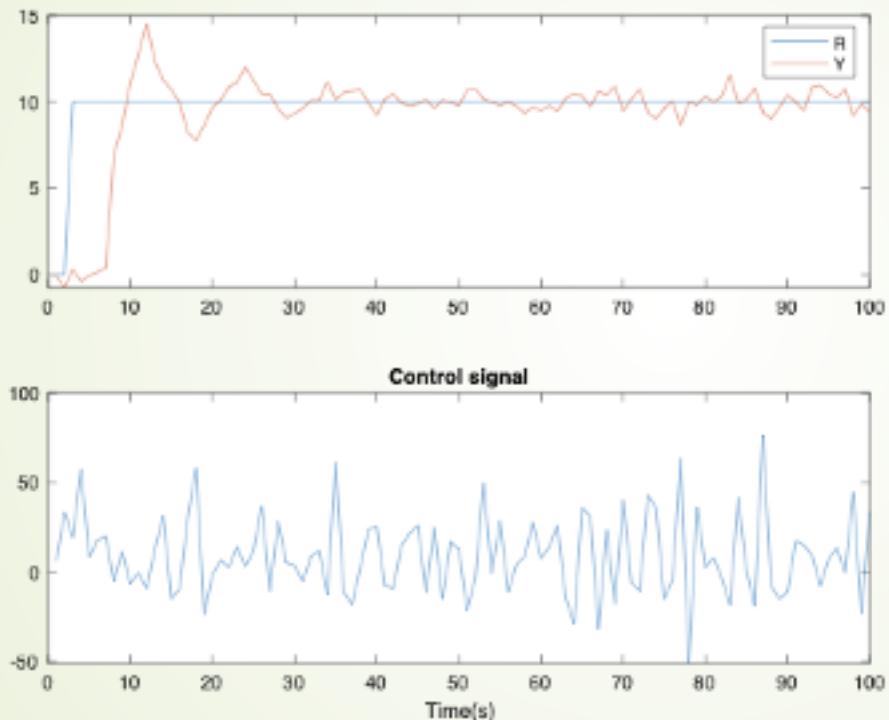
Sintonia PI via Ziegler Nichols

Efeito do ruído na saída e no sinal de controle

No Matlab:

```
[C, iae]=pidtuning(G,'method','zie','type','PI')
```

Exemplo de projeto: escolha do modelo de referência



Sintonia PID via Ziegler Nichols

Efeito do ruído na saída e no sinal de controle

No Matlab:

```
[C, iae]=pidtuning(G,'method','zie','type','PID')
```

Exemplo de projeto: escolha do modelo de referência

Esforço do controlador:

$$G(s) = \frac{1.2e^{-0.2s}}{s + 1}$$

$$C(s) = K_p \left(1 + \frac{1}{T_i s}\right)$$

$$K_p = \frac{2\tau + \theta}{2K\lambda}$$

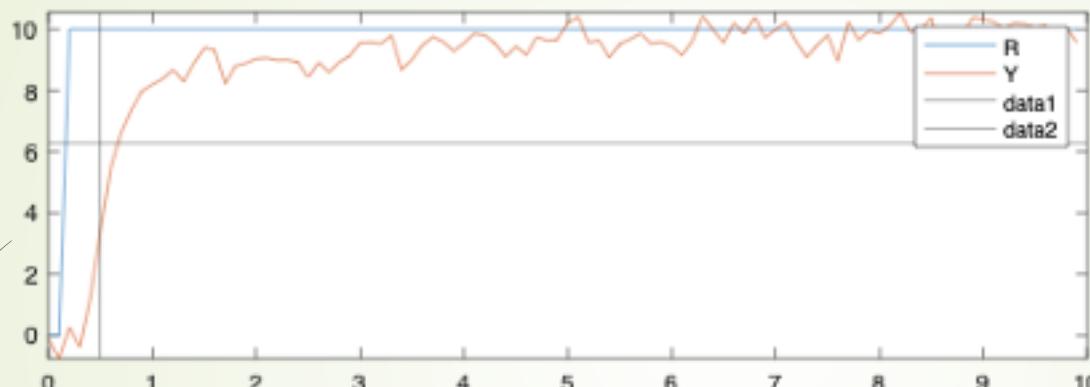
$$T_i = \tau + \frac{\theta}{2}$$

Método IMC ou síntese direta

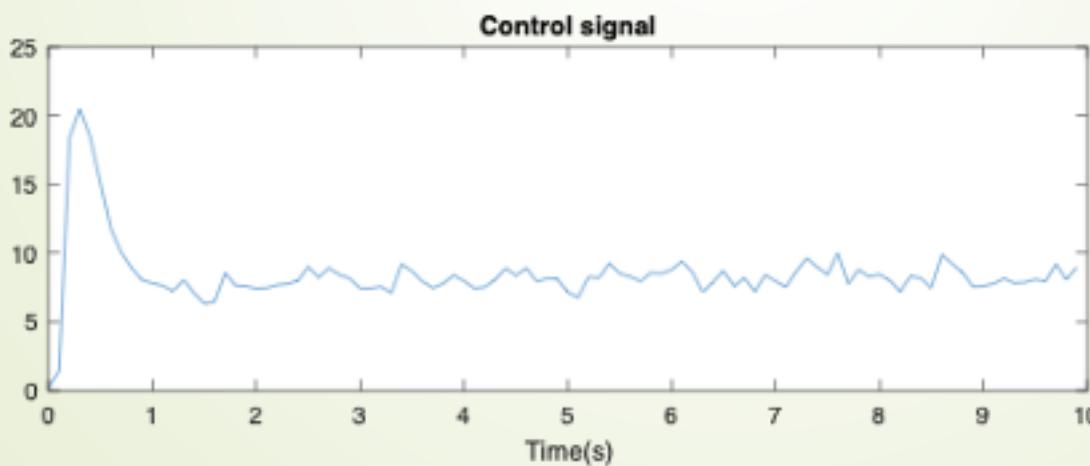
No Matlab:

[C, iae]=pidtuning(G,λ)

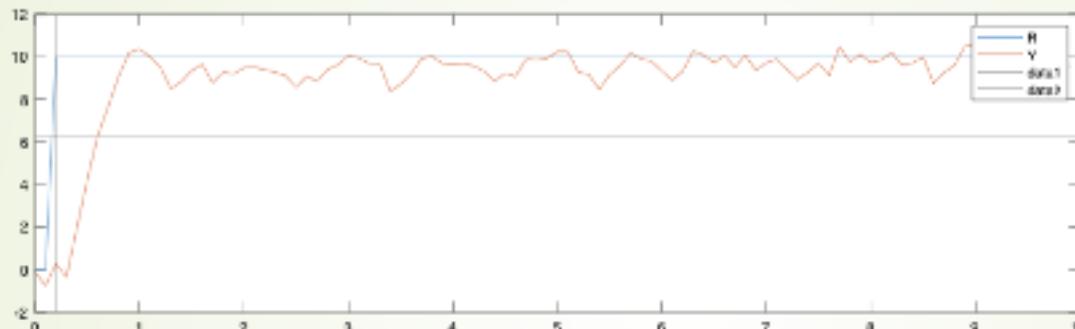
Exemplo de projeto: escolha do modelo de referência



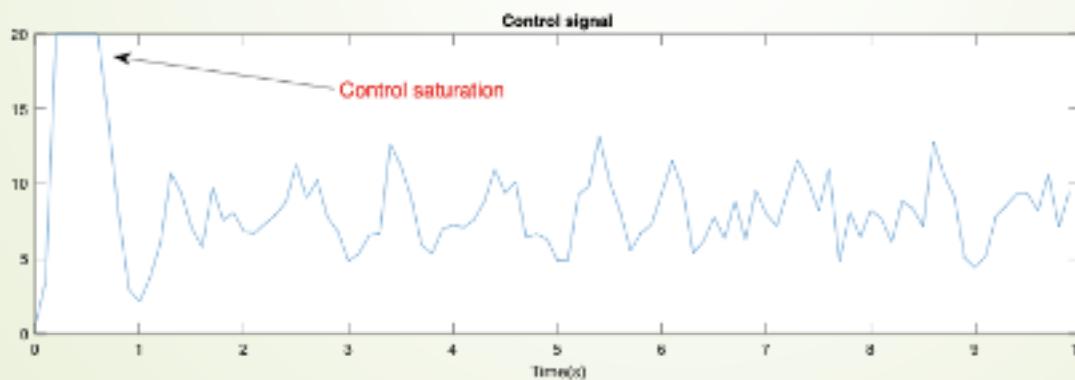
Usando $\lambda = 0.5$, a constante de tempo de malha fechada é próxima de 0.5



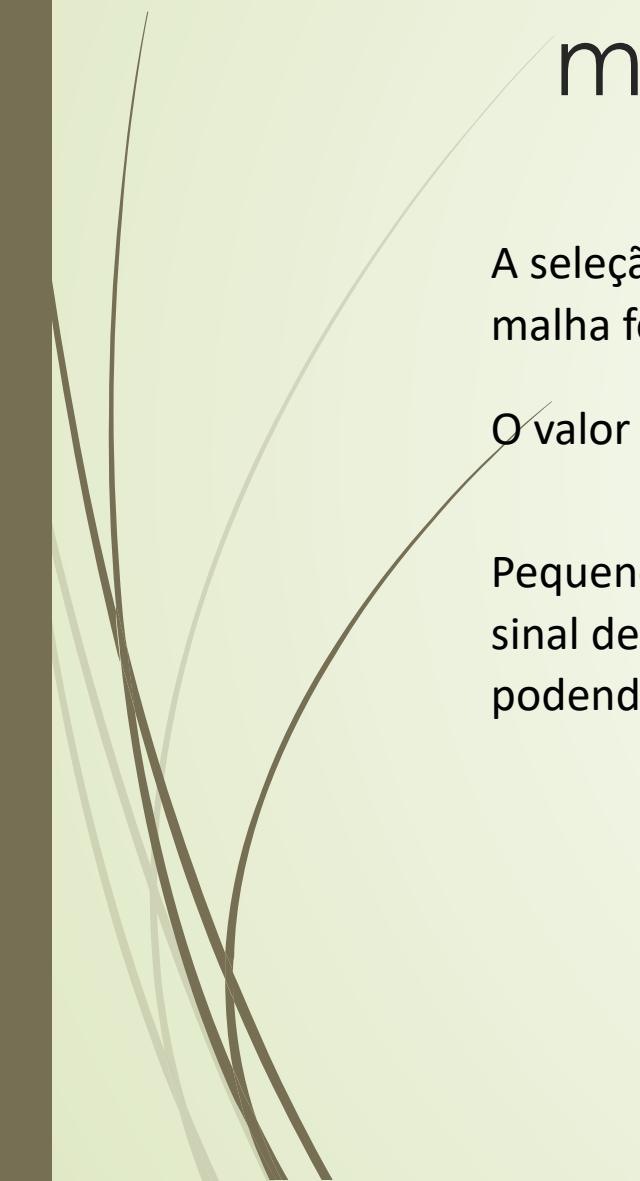
Exemplo de projeto: escolha do modelo de referência



Usando $\lambda = 0.2$, a constante de tempo de malha fechada não é alcançada.



O sinal de controle é muito agressivo neste caso.



Exemplo de projeto: escolha do modelo de referência

A seleção de λ permite escolher a constante de tempo de malha fechada desejada.

O valor de λ deve ser maior que o tempo morto.

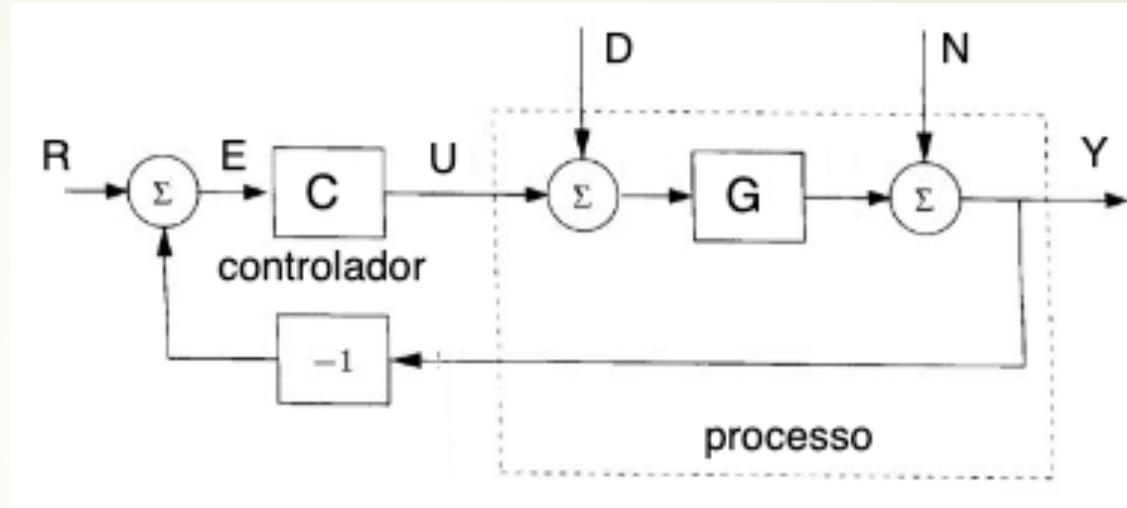
Pequenos valores de λ resultam em ganhos maiores e um sinal de controle mais agressivo (maior intensidade), podendo haver saturação.

Passos para o projeto baseado em modelo

O métodos baseados síntese direta baseiam-se nos seguintes passos:

1. Dada uma resposta ao degrau, identificar a dinâmica: primeira ou segunda ordem, com ou sem tempo morto.
2. Identificar o modelo associado a esta resposta e seus parâmetros, lembrando que diferentes modelos podem ser associados a uma resposta, com aproximações melhores ou piores
3. Escolher o modelo de referência: $M(s) = T(s) = \frac{1}{\lambda s + 1}$ ou $M(s) = \frac{e^{-\theta s}}{\lambda s + 1}$ ou $M(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$
4. Calcular os ganhos do controlador PI ou PID correspondentes a $G(s)$ e $M(s)$ escolhidos.
5. Fechar a malha com “cautela”. Se for um PID, o polo do filtro pode ser necessário para torná-lo causal. Em sistemas com atraso e ganho derivativo, o atraso deve ser aproximado antes de fechar a malha.
6. Simular para entrada degrau e distúrbio, mostrando a saída e o sinal de controle.

Simulação dos sistemas em malha fechada



Ao projetar o controlador C, estamos interessados na análise das relações:

$$\frac{Y}{R} = \frac{CG}{1 + CG} \quad \frac{E}{R} = \frac{1}{1 + CG} \quad \frac{U}{R} = \frac{C}{1 + CG}$$

Essas várias FTs podem ser usadas para fazer o gráfico dos sinais de interesse com um único comando step.

$$\frac{Y}{D} = \frac{G}{1 + CG} \quad \frac{Y}{N} = \frac{1}{1 + CG}$$

Exemplo: m1=feedback(C*G,1)
m2=feedback(1,C*G)
step(m1,m2)

Controladores com dois graus de liberdade

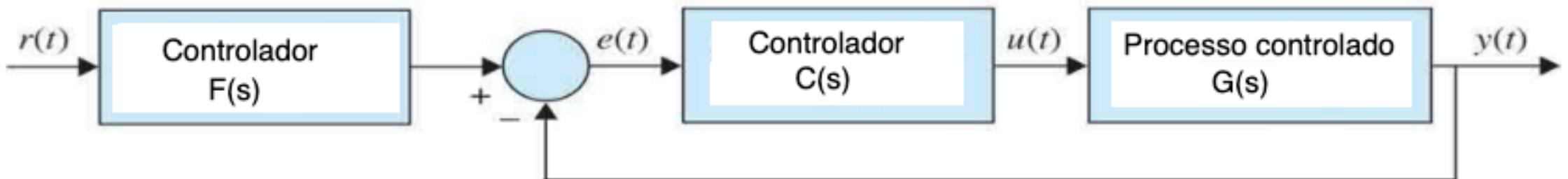
Referências para este assunto:

- Kuo, Benjamin C. , Automatic Control Systems, Prentice-Hall International, 10 th edition, McGraw-Hill (2017). Cap 11.1 e 11.7
- Ogata, K. Engenharia de Controle Moderno, 4a edição. Prentice Hall, São Paulo, 2003. Cap 8.6
- Seborg, Dale E.; Edgar, Thomas F.; Mellicamp, Dukan E ; Doyle III, Francis J. Process Dynamics and Control, John Wiley & Sons, 3a ed. 2011. Cap 12.4
- Astrom, Karl J.; Hagglund, Tore. Advanced PID Control. ISA, 2006. Cap 3.4 e Cap 5

Controladores com dois graus de liberdade

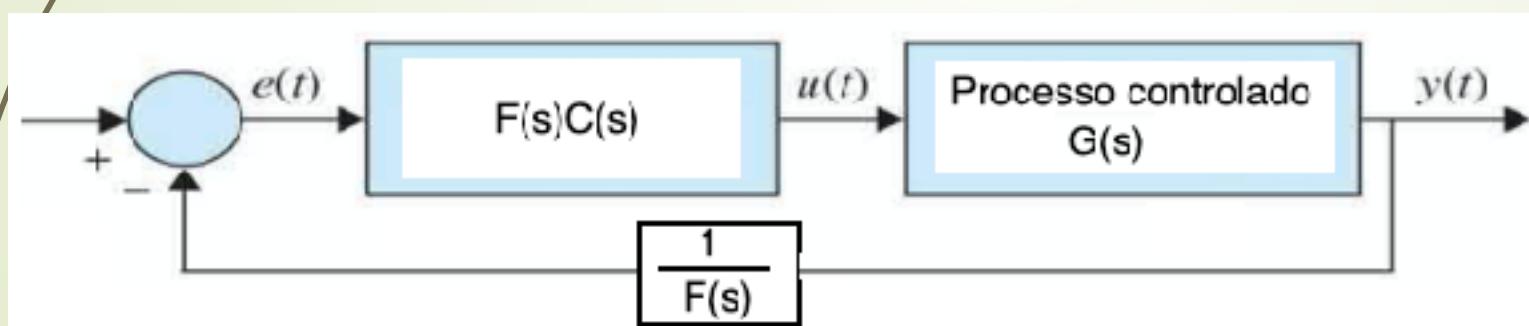
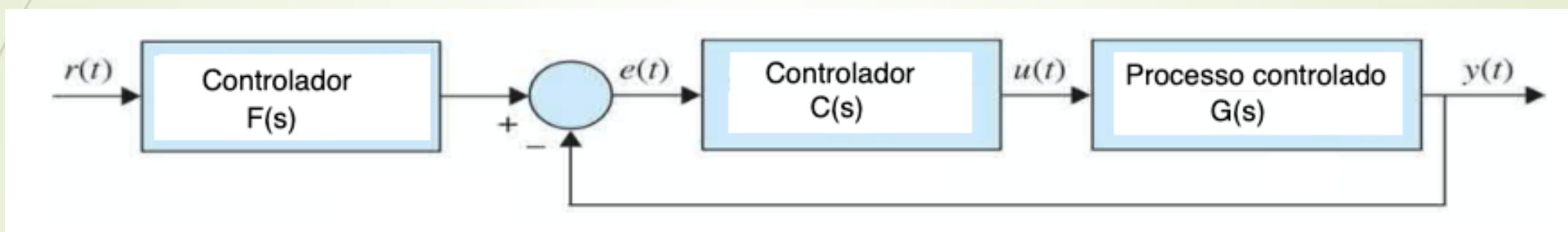
Nas situações discutidas até agora apenas um controlador é usado. Nesse caso, afirma-se que há apenas 1 grau de liberdade

Seja o sistema de controle mostrado na figura abaixo.



Controladores com dois graus de liberdade

Colocando $F(s)$ dentro da malha fechada,

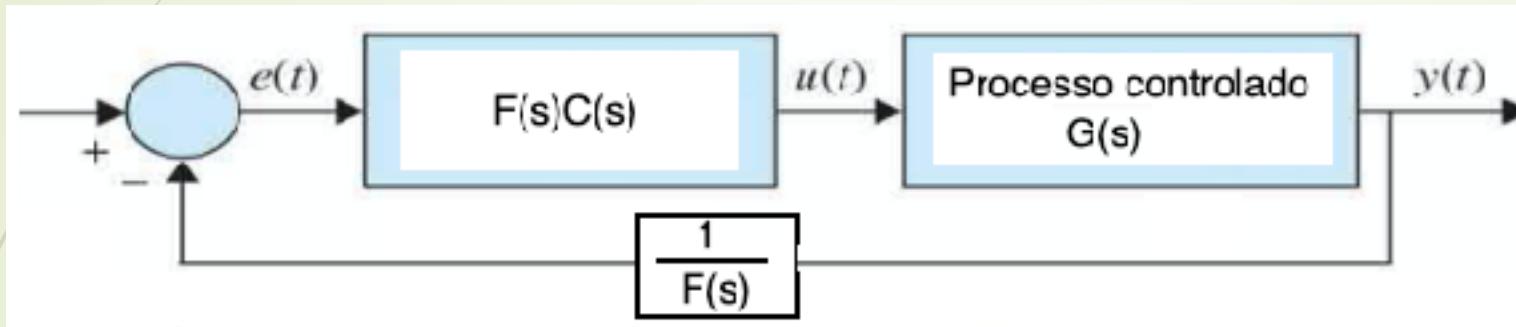


$$\frac{Y(s)}{R(s)} = \frac{F(s)C(s)G(s)}{1 + C(s)G(s)}$$

$$E(s) = R(s) - F(s)C(s)G(s)F(s)^{-1}$$

$$\frac{E(s)}{R(s)} = \frac{1}{1 + C(s)G(s)}$$

Controladores com dois graus de liberdade

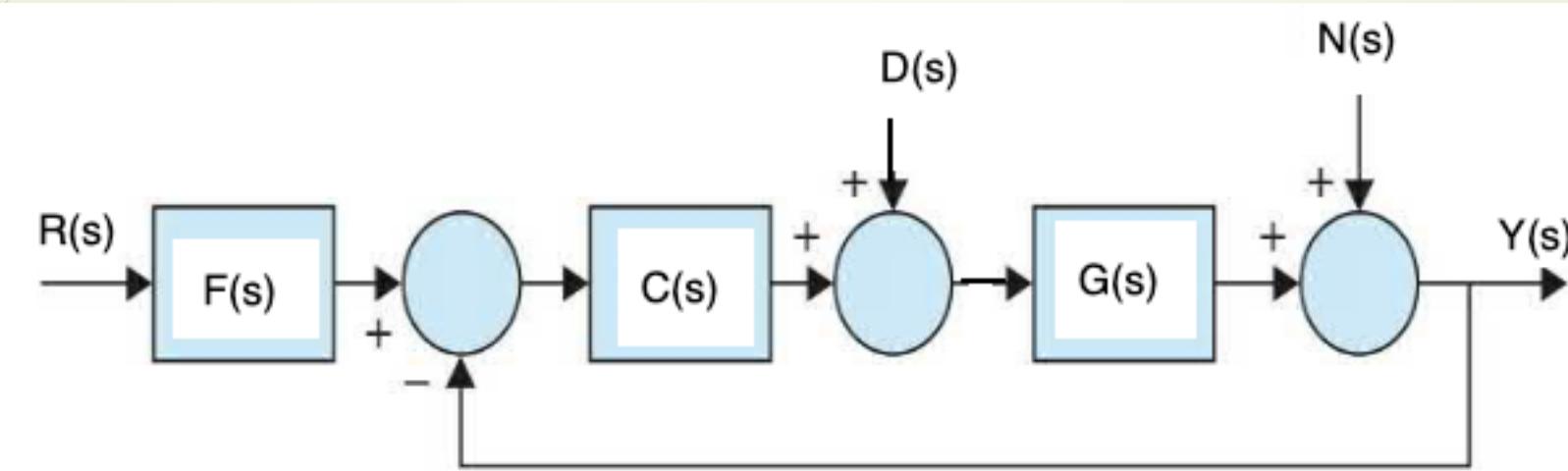


$$\frac{Y(s)}{R(s)} = \frac{F(s)C(s)G(s)}{1 + C(s)G(s)} \quad \frac{E(s)}{R(s)} = \frac{1}{1 + C(s)G(s)}$$

Portanto, controlador $C(s)$ pode ser projetado de modo que o erro tenha características desejadas.

O controlador $F(s)$ pode ser projetado para que a relação entre $R(s)$ e $Y(s)$ atenda especificações adicionais.

Controladores com dois graus de liberdade



$$\frac{Y(s)}{R(s)} = \frac{F(s)C(s)G(s)}{1 + C(s)G(s)}$$

$$\frac{Y(s)}{D(s)} = \frac{G(s)}{1 + C(s)G(s)}$$

$$\frac{Y(s)}{N(s)} = \frac{1}{1 + C(s)G(s)}$$

O controlador $F(s)$ aparece apenas na relação entre R e Y .



Controladores com dois graus de liberdade

Outra maneira de descrever a flexibilidade de um projeto de dois graus de liberdade é que o controlador $C(s)$ é geralmente projetado para fornecer um certo grau de estabilidade e desempenho do sistema.

Mas como os zeros de $C(s)$ sempre se tornam os zeros da função de transferência de malha fechada, a menos que alguns dos zeros sejam cancelados pelos polos da função de transferência do processo, $G(s)$, esses zeros podem causar uma grande sobreelevação na saída do sistema, mesmo quando há um amortecimento relativo definido pela equação característica.

Controladores com dois graus de liberdade

Neste caso e por outras razões, a função de transferência $F(s)$ pode ser utilizada para o controle ou cancelamento dos zeros indesejáveis da função de transferência em malha fechada, mantendo a equação característica obtida via $C(s)$.

$$\zeta = 0.7$$

$$UP=4.6\%$$

$$\omega_n = 1$$

$$t_s = \frac{4}{\zeta \omega_n} = \frac{4}{0.7 * 1} = 5.7$$

$$M(s) = \frac{s + 1}{s^2 + 1.4s + 1}$$

$$M_1(s) = \frac{1}{s^2 + 1.4s + 1}$$

$M_1(s)$ dá a resposta com UP e t_s desejados.

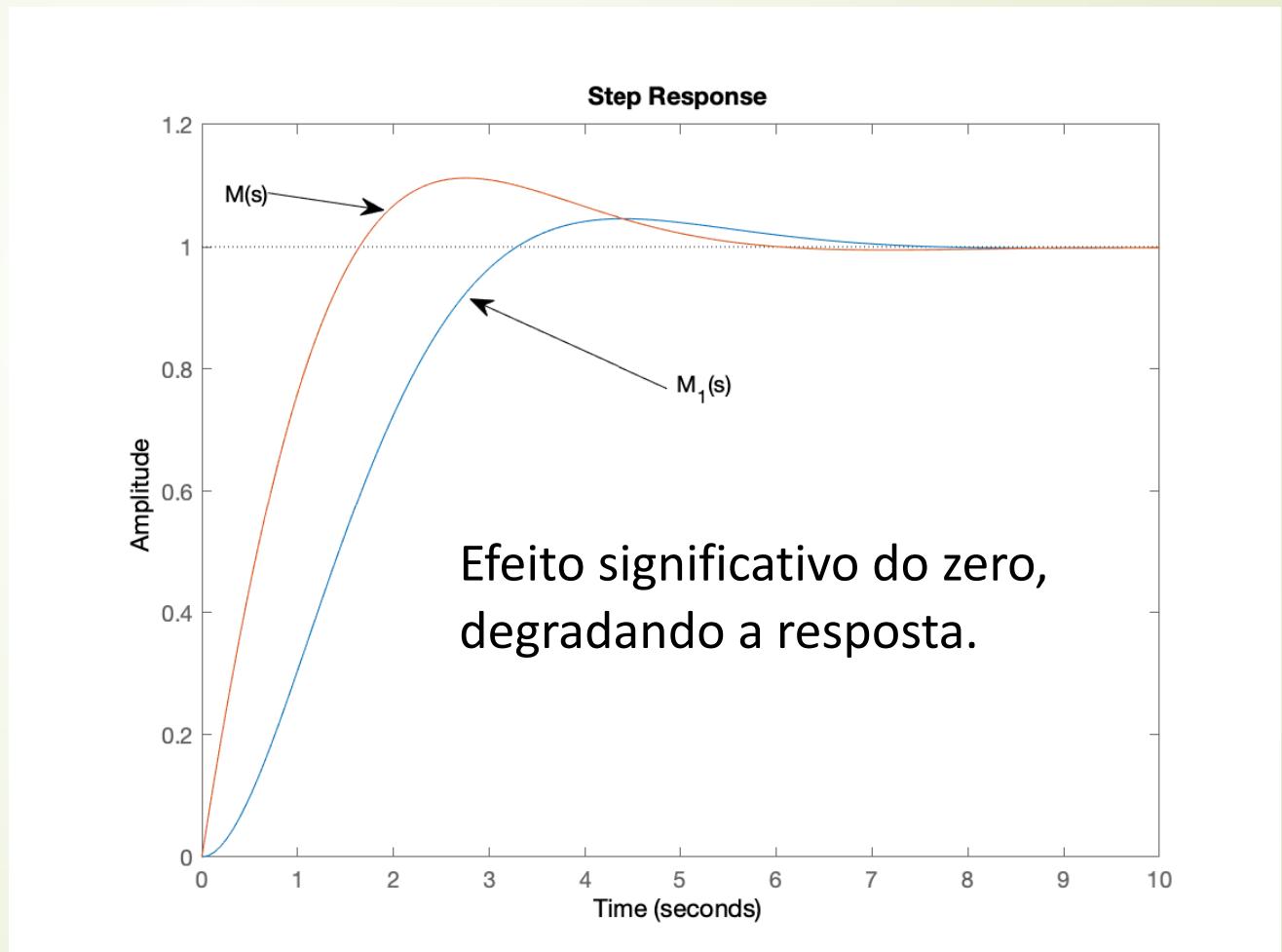
Controladores com dois graus de liberdade

$$\zeta = 0.7 \quad UP=4.6\%$$

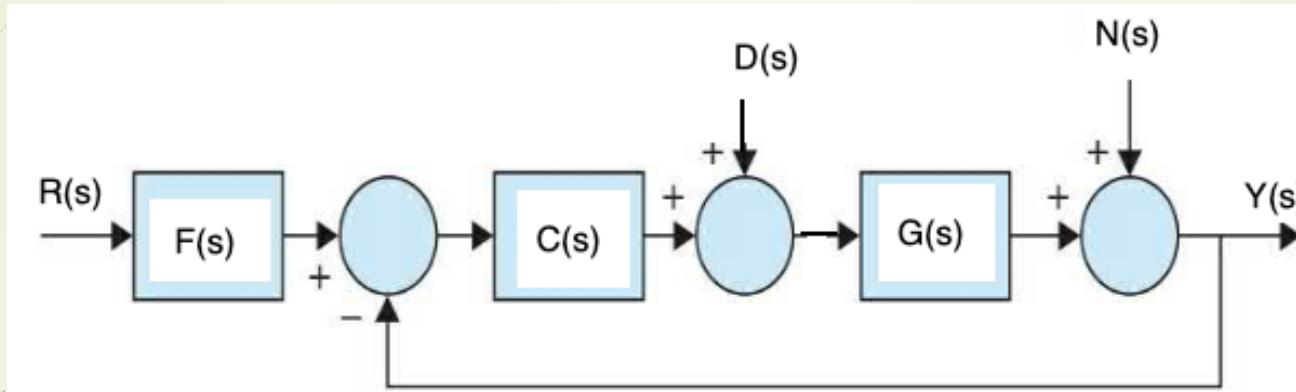
$$\omega_n = 1 \quad t_s = \frac{4}{0.7 * 1} = 5.7$$

$$M(s) = \frac{s + 1}{s^2 + 1.4s + 1}$$

$$M_1(s) = \frac{1}{s^2 + 1.4s + 1}$$



Controladores com dois graus de liberdade

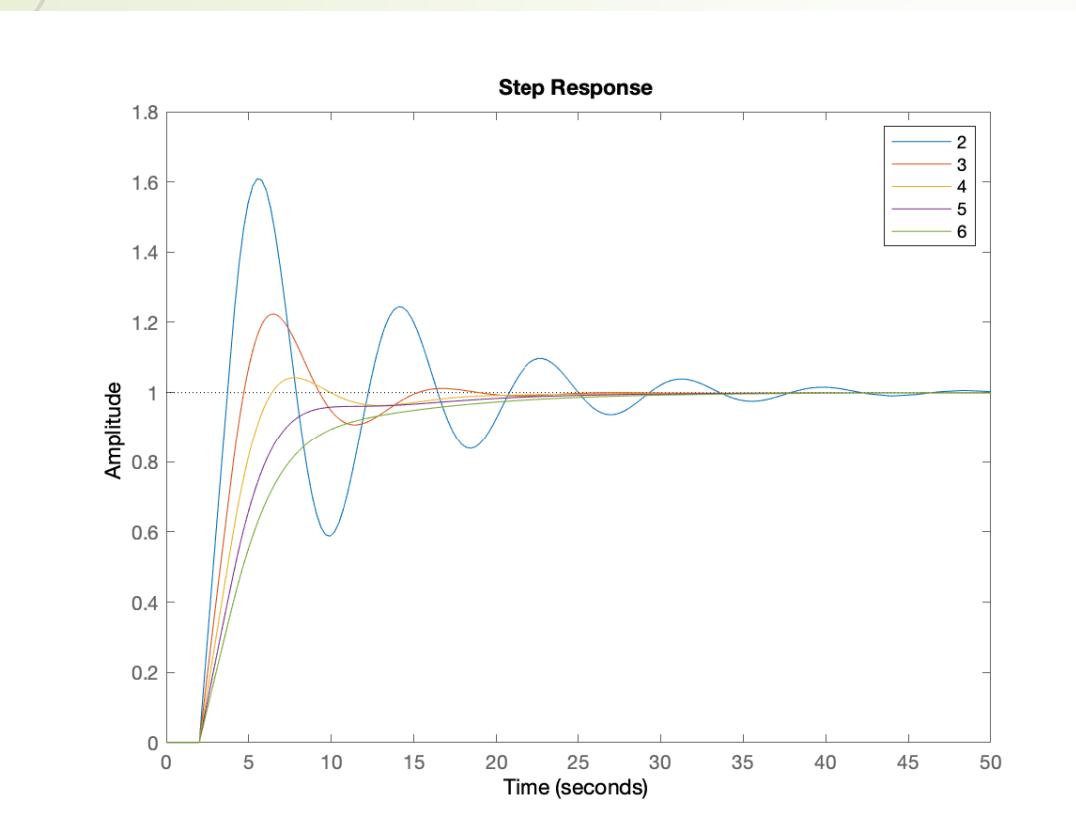
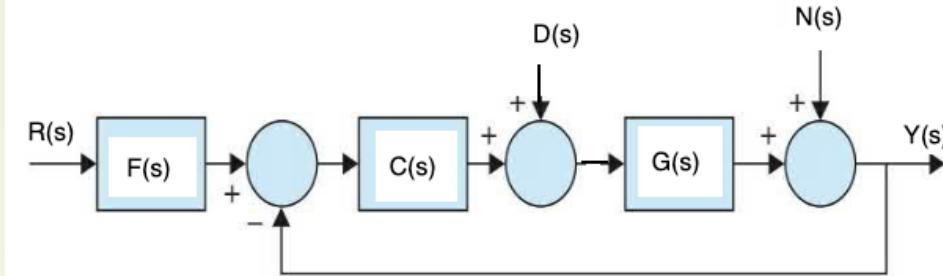


Exemplo 1:

Seja $G(s) = \frac{2e^{-2s}}{5s + 1}$ e o projeto via síntese direta do PI:

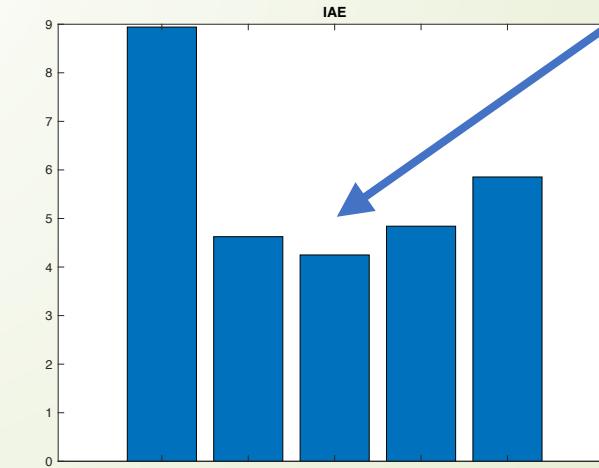
$$K_p = \frac{1}{K} \frac{\tau}{\theta + \tau_c} \quad \tau_i = \tau$$

Controladores com dois graus de liberdade

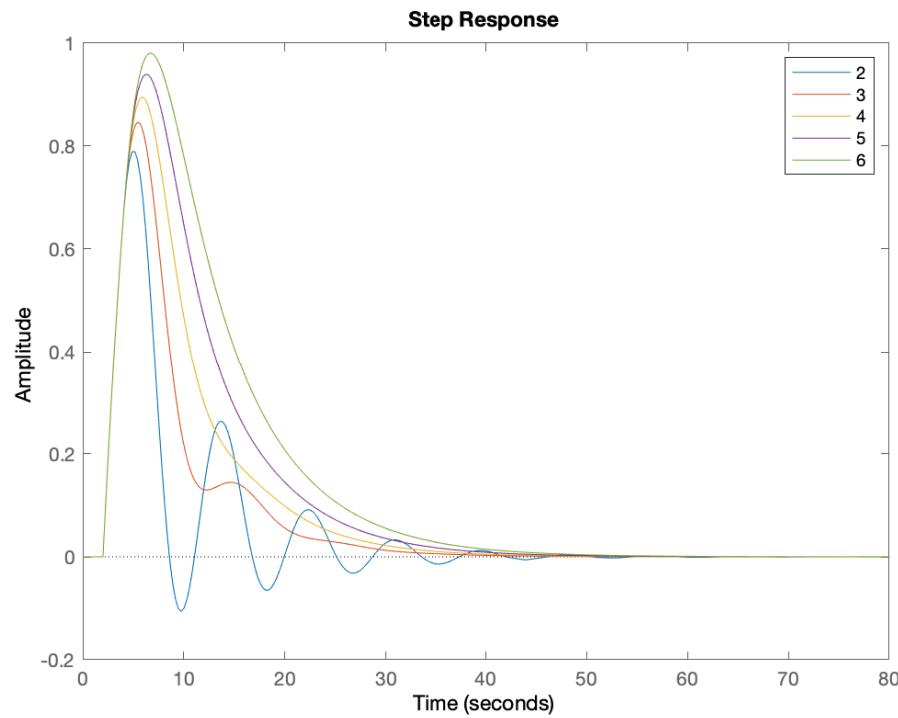
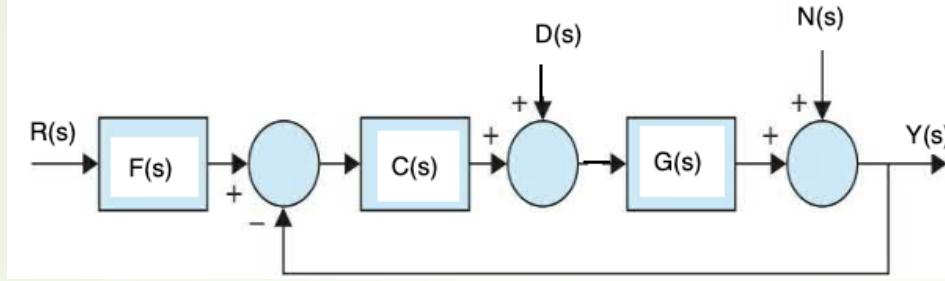


$$G(s) = \frac{2e^{-2s}}{5s + 1}$$
$$K_p = \frac{1}{K} \frac{\tau}{\theta + \tau_c} \quad \tau_i = \tau$$

Resposta para $\lambda = \{2,3,4,5,6\}$

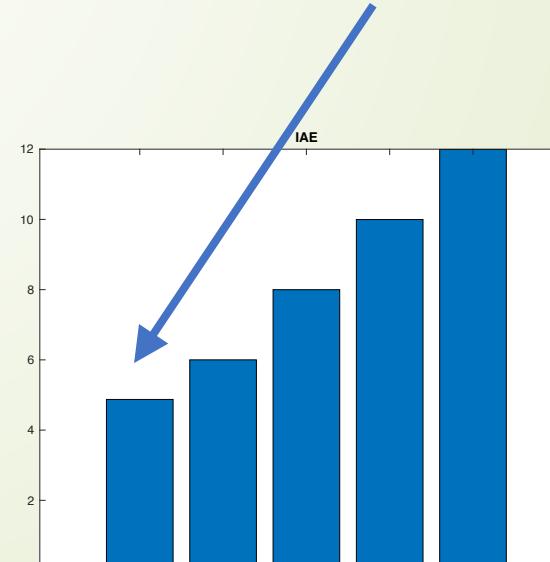


Controladores com dois graus de liberdade

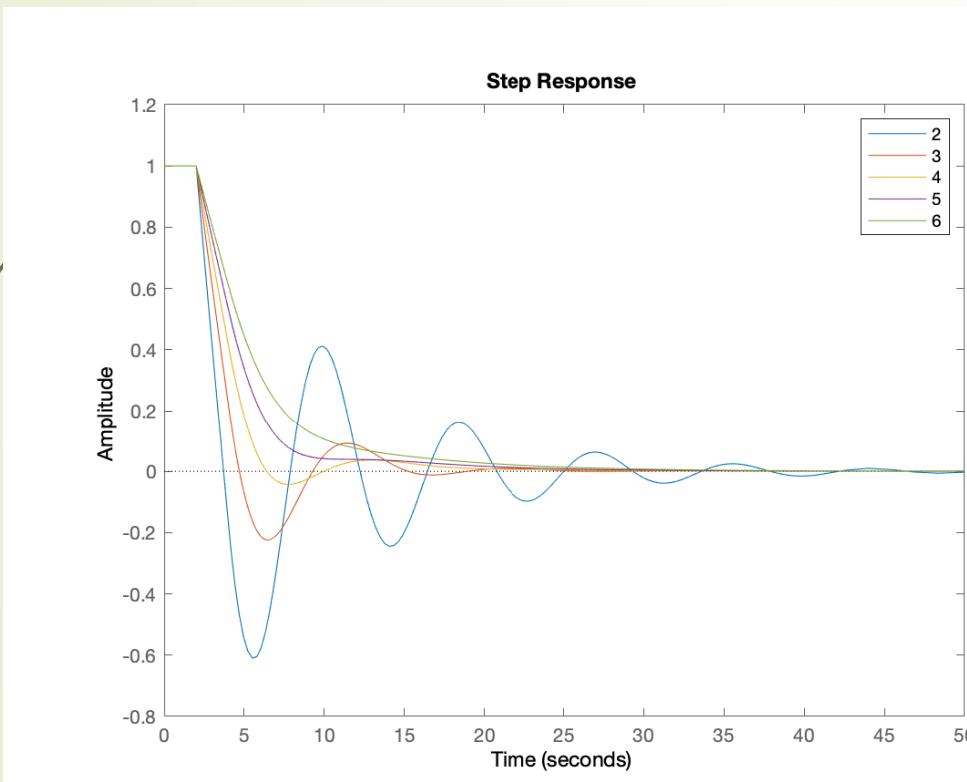
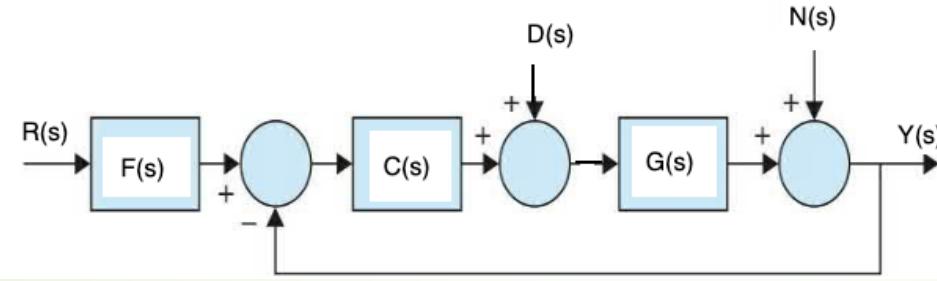


$$G(s) = \frac{2e^{-2s}}{5s + 1}$$
$$K_p = \frac{1}{K} \frac{\tau}{\theta + \tau_c} \quad \tau_i = \tau$$

Resposta ao distúrbio D
para $\lambda = \{2,3,4,5,6\}$

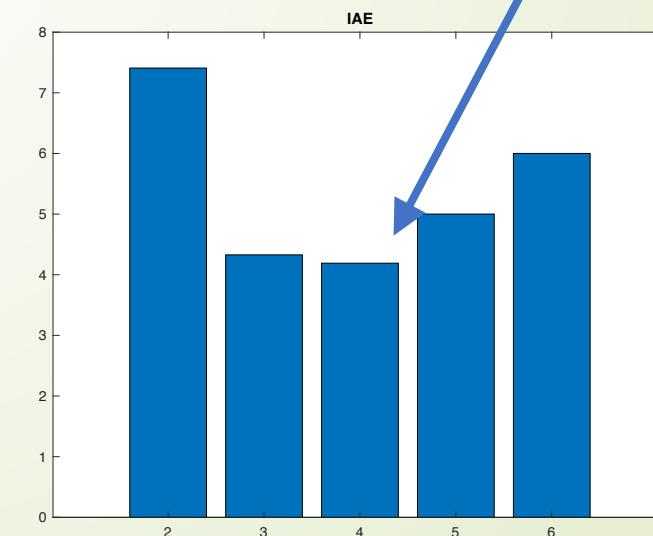


Controladores com dois graus de liberdade

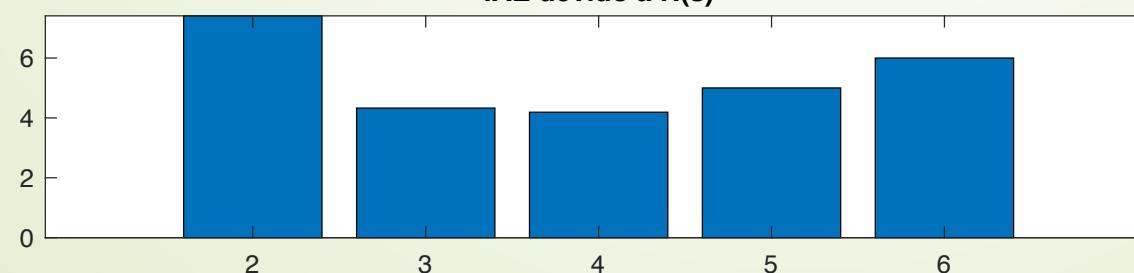
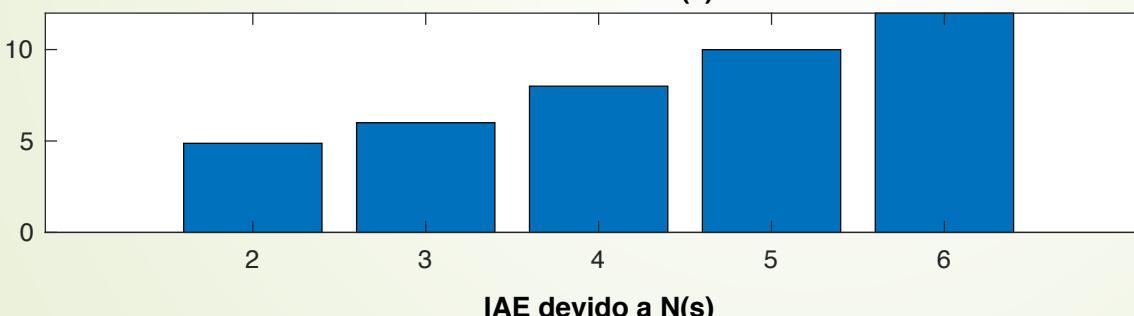
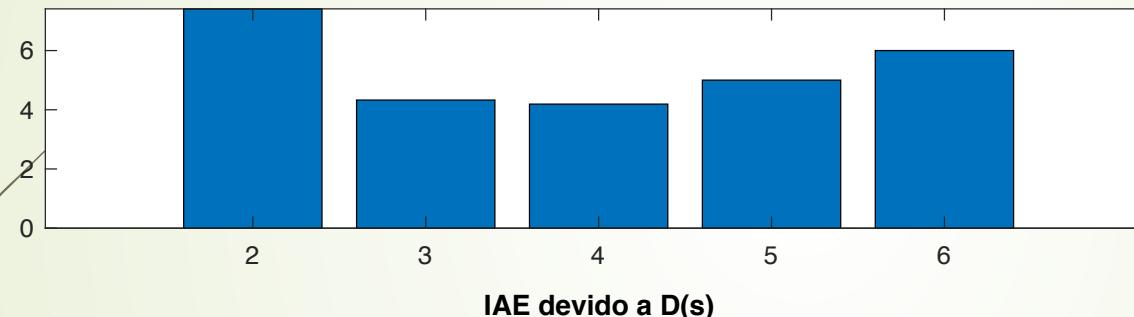
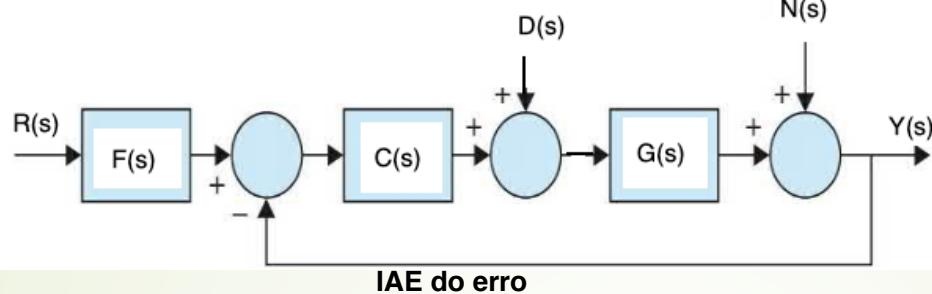


$$G(s) = \frac{2e^{-2s}}{5s + 1}$$
$$K_p = \frac{1}{K} \frac{\tau}{\theta + \tau_c} \quad \tau_i = \tau$$

Resposta ao distúrbio N
para $\lambda = \{2,3,4,5,6\}$



Controladores com dois graus de liberdade



Aumentar o ganho melhora a resposta ao distúrbio D e piora a resposta à entrada R e distúrbio em N .

Controladores com dois graus de liberdade

Como $F(s)$ afeta apenas Y/R , usaremos este compensador para melhorar a resposta. Assim podemos manter o controlador que deu uma boa resposta ao distúrbio.

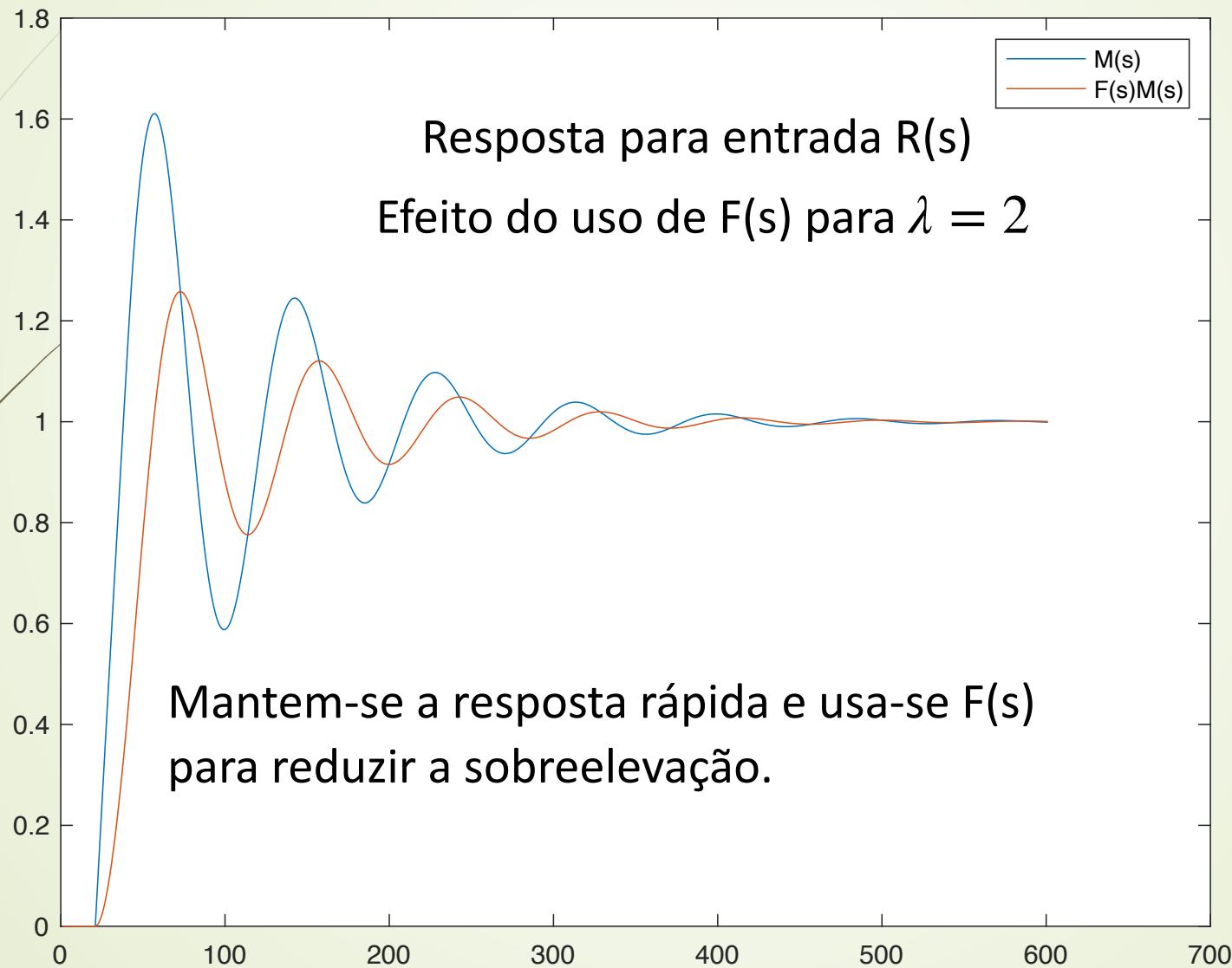
$$\frac{Y(s)}{R(s)} = \frac{F(s)C(s)G(s)}{1 + C(s)G(s)} \quad \frac{Y(s)}{D(s)} = \frac{G(s)}{1 + C(s)G(s)} \quad \frac{Y(s)}{N(s)} = \frac{1}{1 + C(s)G(s)}$$

$$\frac{Y(s)}{R(s)} = M(s) = \frac{C(s)G(s)}{1 + C(s)G(s)} = \frac{-3s^2 + 2.5s + 0.5}{5s^3s^2 + 3.5s + 0.5}$$

Os zeros de $M(s)$ são 1 e -0.167. O zero no SPD não pode ser cancelado.

Usaremos um zero próximo a -0.167, com $F(s) = \frac{1}{2s + 1}$

Controladores com dois graus de liberdade



Controladores com dois graus de liberdade

Exemplo 2:

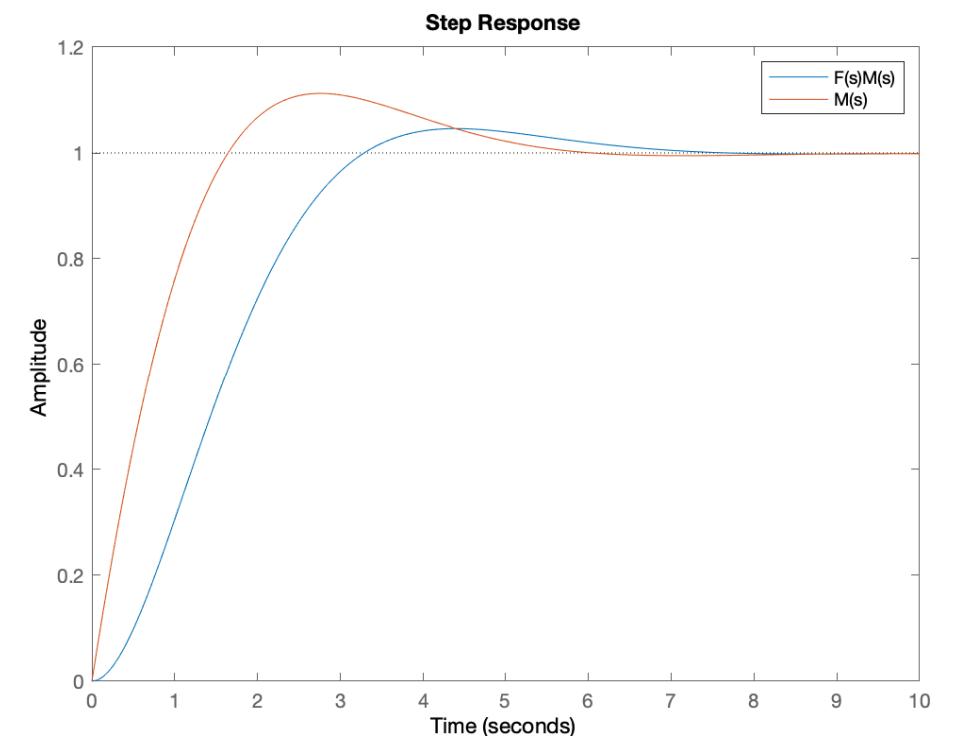
Suponha que a FT de malha fechada resultante do projeto de $C(s)$ seja

$$M(s) = \frac{C(s)G(s)}{1 + C(s)G(s)} = \frac{s + 1}{s^2 + 1.4s + 1}$$

Veja o efeito de um pré compensador

$$F(s) = \frac{1}{s + 1}$$

na resposta:



Controladores com dois graus de liberdade

O compensador $F(s)$ também pode ser usado para cancelar polos de malha fechada indesejados.

Embora este método possa parecer poderoso por cancelar polos e zeros de malha fechada, seu uso é limitado em situações reais.

Há sempre incerteza nos parâmetros dos modelos, o que torna a localização de polos e zeros incerta. Portanto, seu cancelamento pode ser apenas parcial neste casos.

No exemplo 1, a FT de malha fechada é $M(s)$, e o zero em 1 não pode ser cancelado por estar no SPD

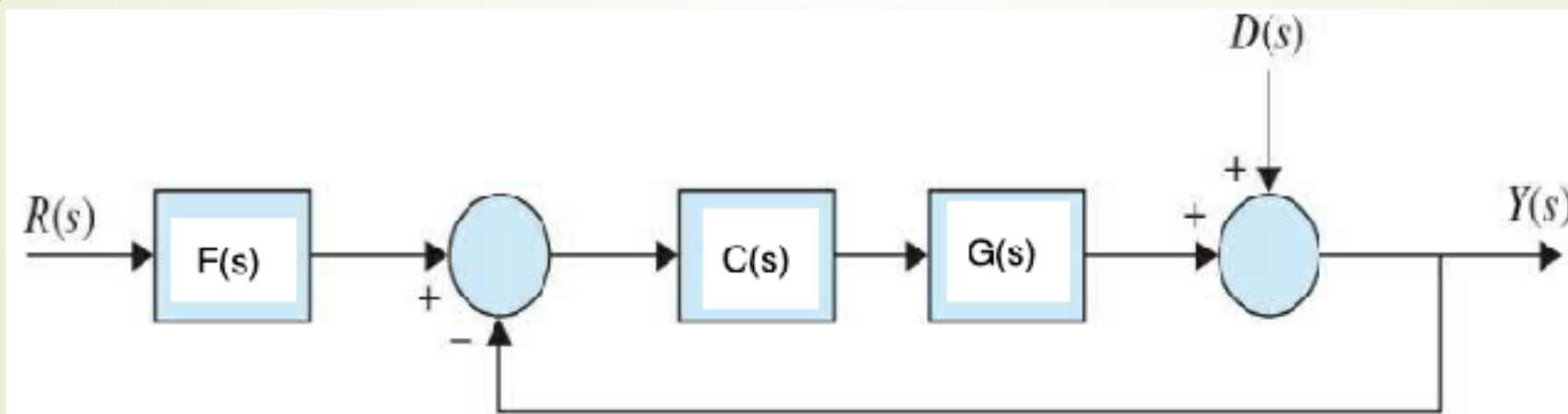
$$M(s) = \frac{-3s^2 + 2.5s + 0.5}{5s^3s^2 + 3.5s + 0.5}$$

Raízes de $-3s^2 + 2.5s + 0.5$ são $\{1, -0.167\}$

Controladores com dois graus de liberdade

O compensador $F(s)$ pode conseguir um bom desempenho (resposta a $R(s)$) e uma boa robustez (resposta do distúrbio $D(s)$).

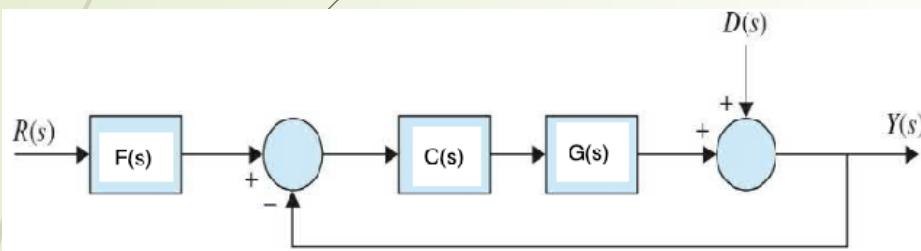
Exemplo 3: Seja $G(s) = \frac{1}{(s + 1)^4}$



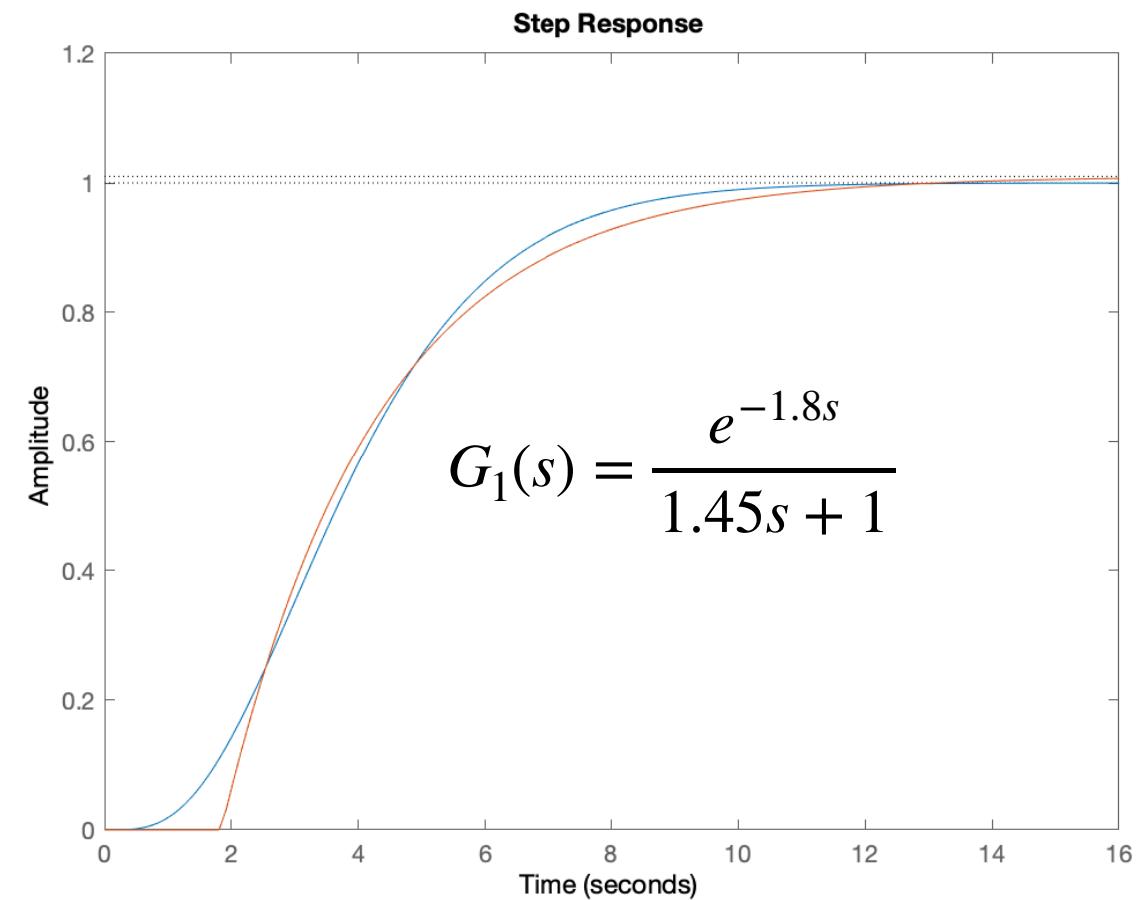
Controladores com dois graus de liberdade

O compensador $F(s)$ pode conseguir um bom desempenho (resposta a $R(s)$) e uma boa robustez (resposta do distúrbio $D(s)$).

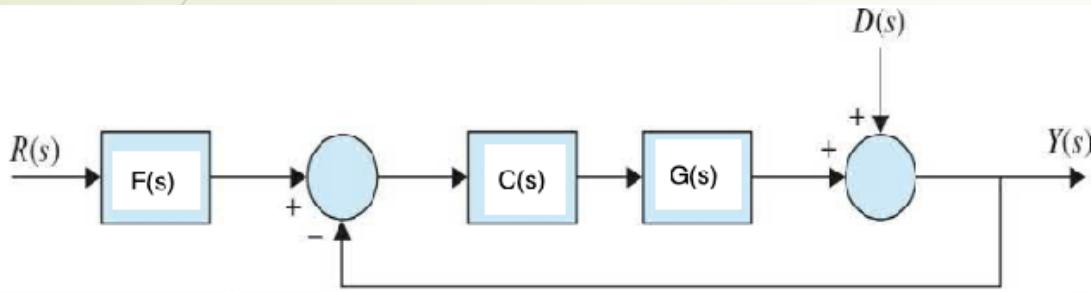
$$\text{Seja } G(s) = \frac{1}{(s + 1)^4}$$



Projetar um controlador usando um modelo aproximado FOPTD.



Controladores com dois graus de liberdade



Projetar um controlador usando um modelo FOPTD.

$$\frac{Y(s)}{R(s)} = \frac{F(s)C(s)G(s)}{1 + C(s)G(s)}$$

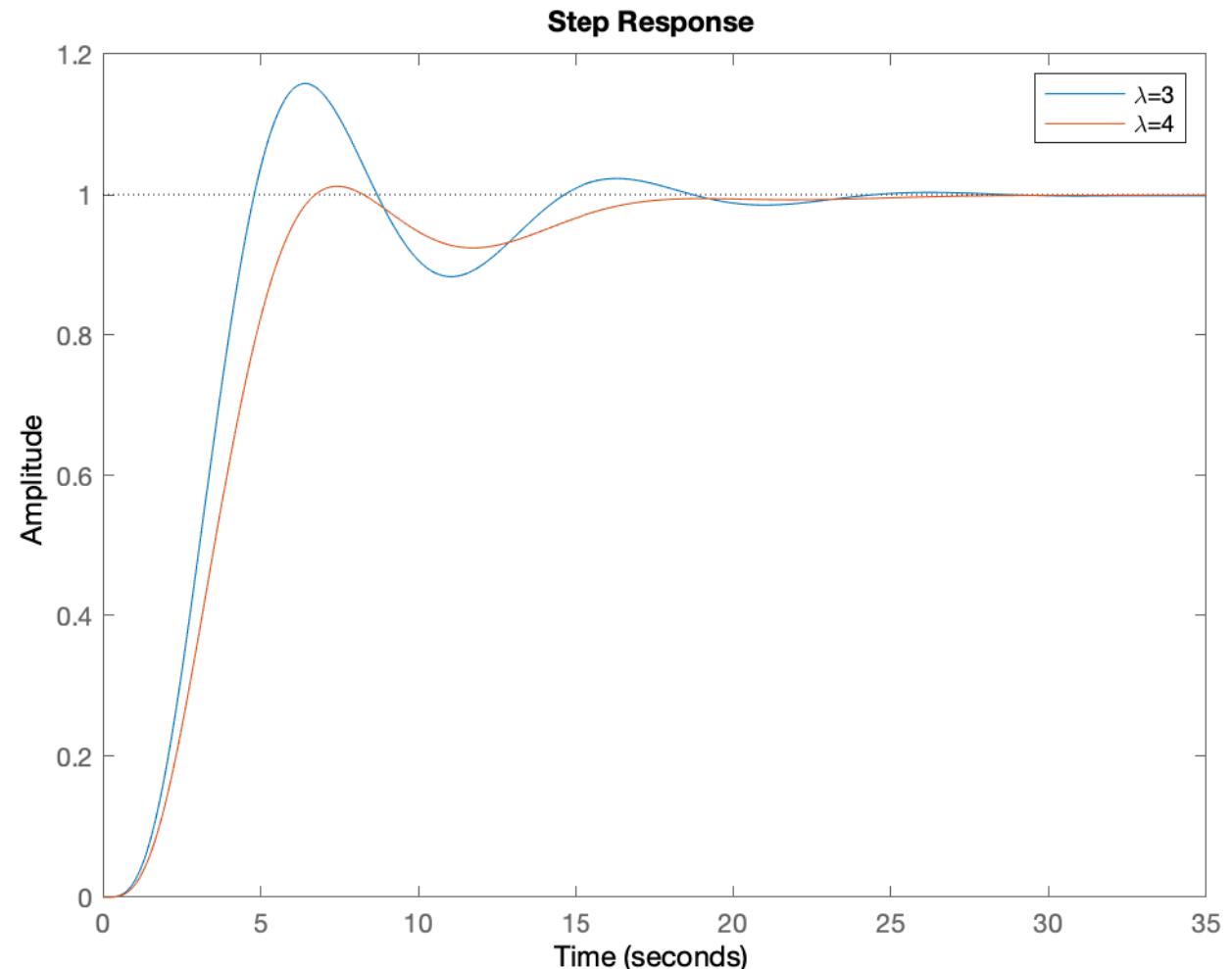
$$\frac{Y(s)}{D(s)} = \frac{1}{1 + C(s)G(s)}$$

$F(s)$ afeta apenas Y/R .

Controladores com dois graus de liberdade

Projeto com
modelo FOPTD e escolhendo
 $\lambda = 3$ e $\lambda = 4$.

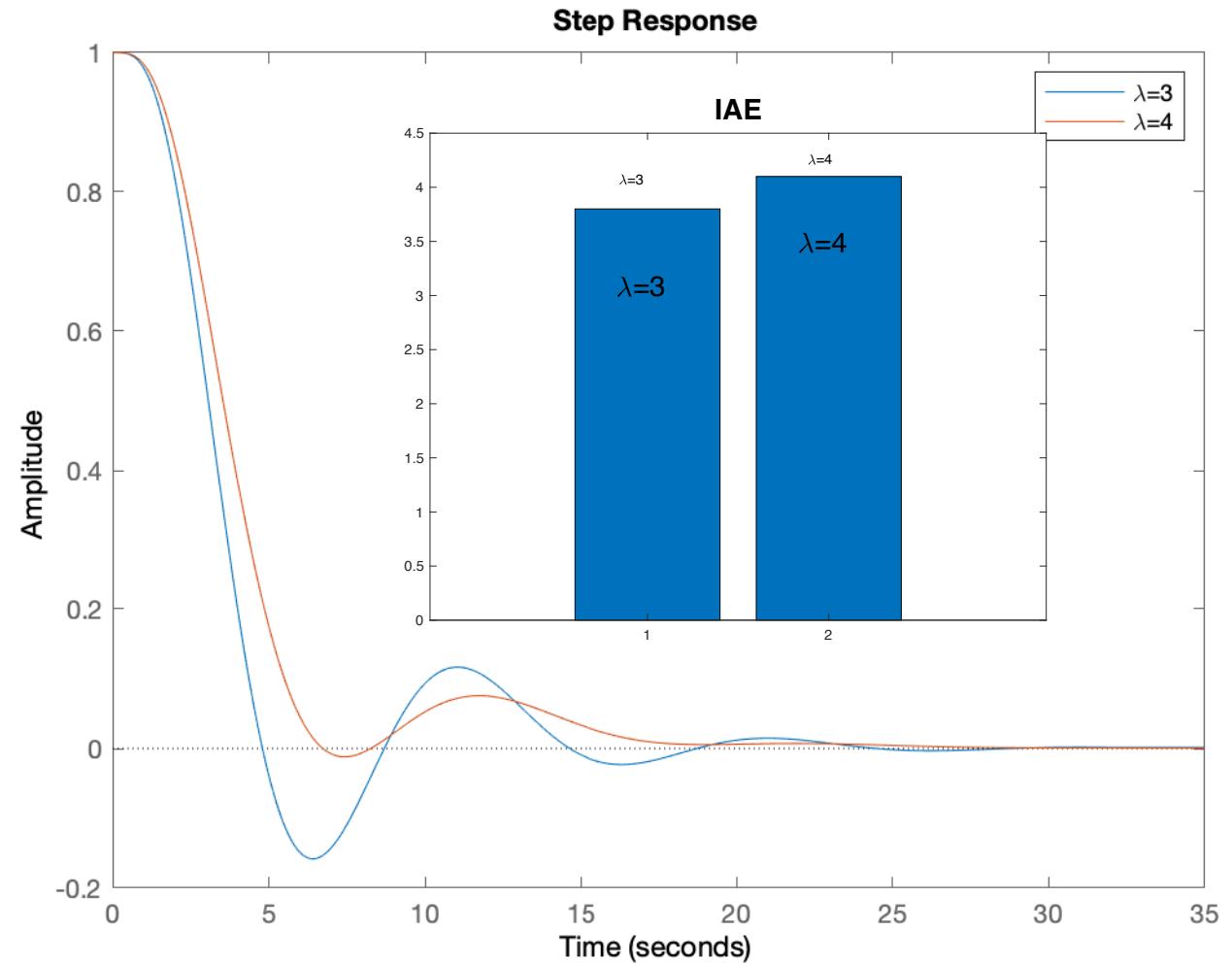
Resposta Y/R tem menor UP
para $\lambda = 4$



Controladores com dois graus de liberdade

Projeto com
modelo FOPTD e escolhendo
 $\lambda = 3$ e $\lambda = 4$.

A resposta Y/D com menor IAE
é para $\lambda = 3$, que dá resposta
pior para Y/R .



Controladores com dois graus de liberdade

$\lambda = 3$ é melhor para resposta ao distúrbio $D(s)$

$\lambda = 4$ é melhor para resposta ao degrau $R(s)$

Usaremos $\lambda = 3$ e um filtro $F(s)$ em $R(s)$ para reduzir a sobreelevação:

$$F(s) = \frac{1}{1.5s + 1}$$

Zero em -0.6

Zero de malha fechada em -0.3

Controladores com dois graus de liberdade

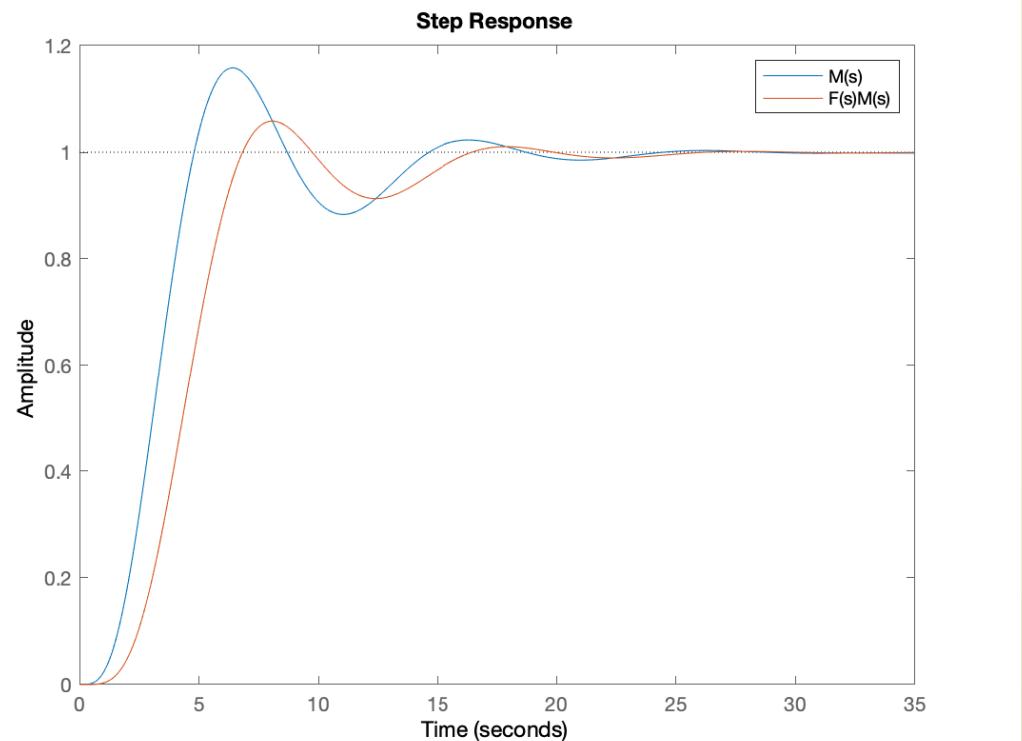
$\lambda = 3$ é melhor para resposta ao distúrbio $D(s)$

$\lambda = 4$ é melhor para resposta ao degrau $R(s)$

Usaremos um filtro $F(s)$ em $R(s)$ para reduzir a sobrelevação:

$$F(s) = \frac{1}{1.5s + 1}$$

Manteve-se o projeto para $\lambda = 3$ usando $F(s)$ melhorou-se a resposta de Y/R .



Controladores com dois graus de liberdade

Discutiremos agora uma técnica conhecida como ponderação da referência (setpoint weighting) na qual são escolhidos pesos para os ganhos proporcional e derivativo.

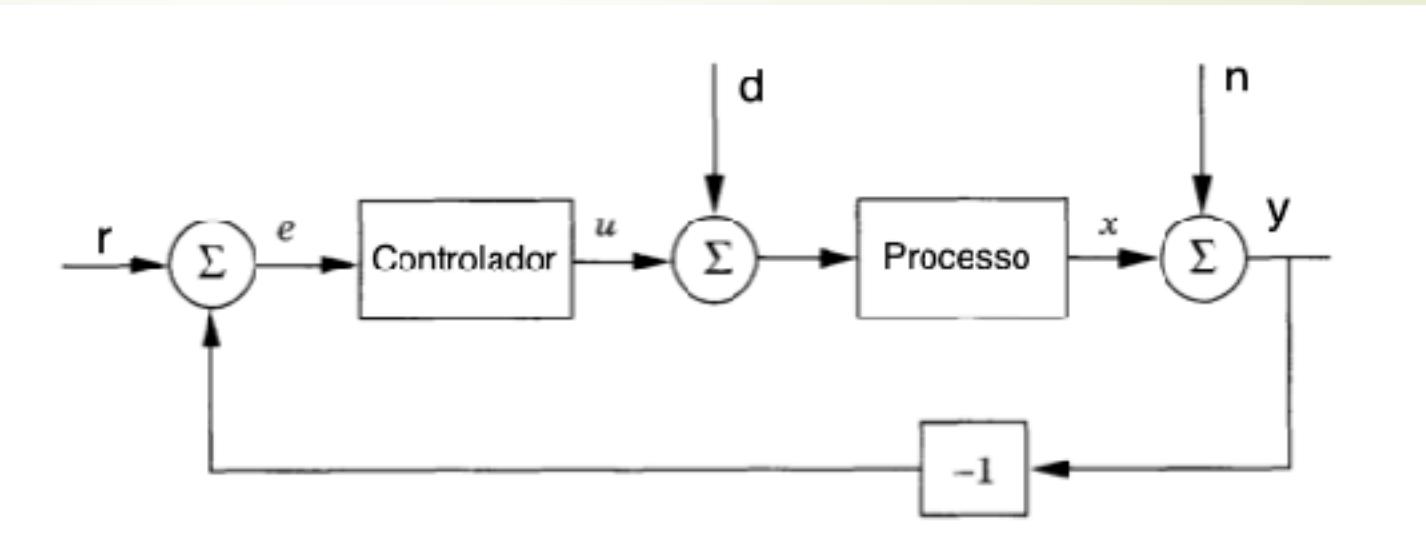
Nesta proposta, a referência e a saída são tratados de forma diferente, conforme abaixo:

$$u(t) = K(e_p + \frac{1}{T_i} \int_0^t e(s)ds + T_D \frac{de_d}{dt})$$

$$e = r - y$$

$$e_p = br - y$$

$$e_d = cr - y$$



Controladores com dois graus de liberdade

Escolhendo $b=0$, o efeito proporcional é aplicado apenas em y .

Escolhendo $c=0$, o efeito derivativo é aplicado apenas em y .

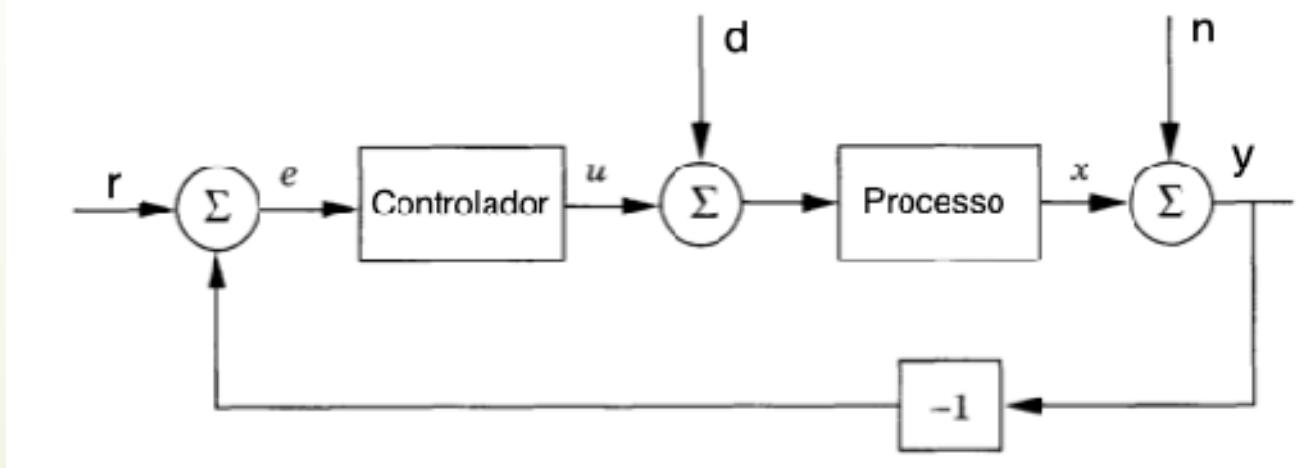
O ganho integral é responsável por fazer o erro entre r e y tender a zero, e portanto, é sempre aplicado a $r-y$.

$$u(t) = K(e_p + \frac{1}{T_i} \int_0^t e(s)ds + T_D \frac{de_d}{dt})$$

$$e = r - y$$

$$e_p = br - y$$

$$e_d = cr - y$$



Controladores com dois graus de liberdade

Como a escolha de b e c afeta apenas $r-y$, a resposta à distúrbios em n e d ocorre da mesma forma para diferentes valores de b e c .

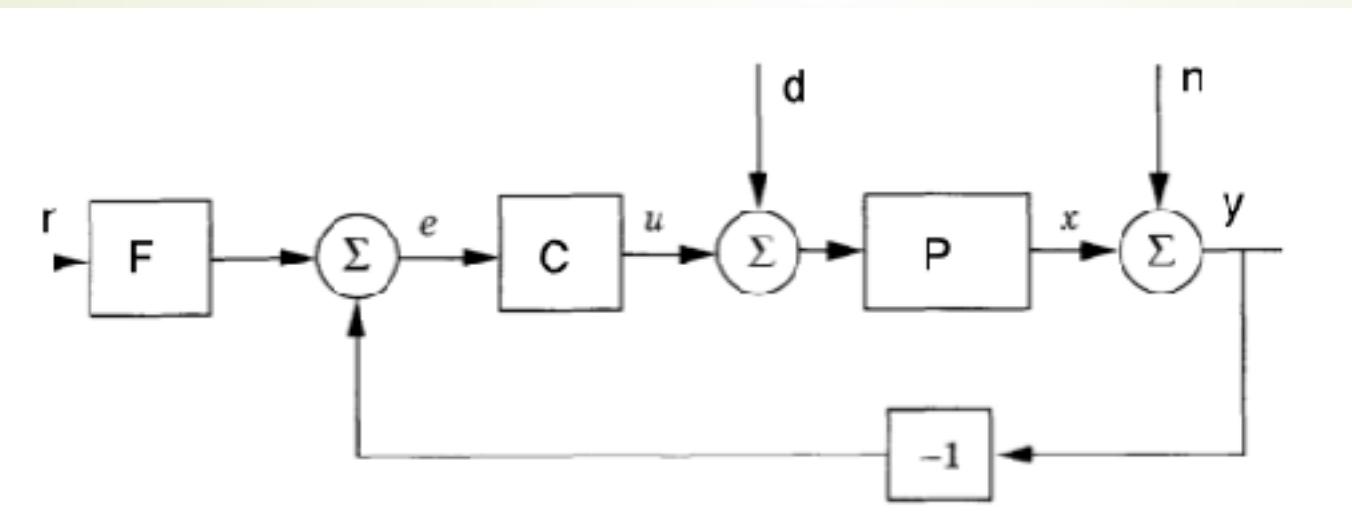
Entretanto, a resposta para mudanças na referência dependerá desses parâmetros b e c .

Controladores com dois graus de liberdade

Esta estratégia pode ser apresentada como na figura abaixo, tendo os controladores C e F, assim definidos:

$$C(s) = K \left(1 + \frac{1}{sT_i} + sT_d \right)$$

$$F(s) = \frac{b + \frac{1}{sT_i} + scT_d}{1 + \frac{1}{sT_i} + sT_d} = \frac{cT_i T_d s^2 + bsT_i + 1}{T_i T_d s^2 + sT_i + 1}$$



Controladores com dois graus de liberdade

Seja o modelo já utilizado de ordem 4 $G(s)$ aproximado por $G_1(s)$,

$$G(s) = \frac{1}{(s + 1)^4} \quad G_1(s) = \frac{e^{-1.8s}}{1.45s + 1}$$

Fazendo o projeto de um controlador PID com $\lambda = 1.5$, resulta:

```
[c,iae]=pidtuning(g0,'method','lambda','type','PID','param',1.5)
```

$$C(s) = K\left(1 + \frac{1}{sT_i} + sT_d\right)$$

$$K = 0.81$$

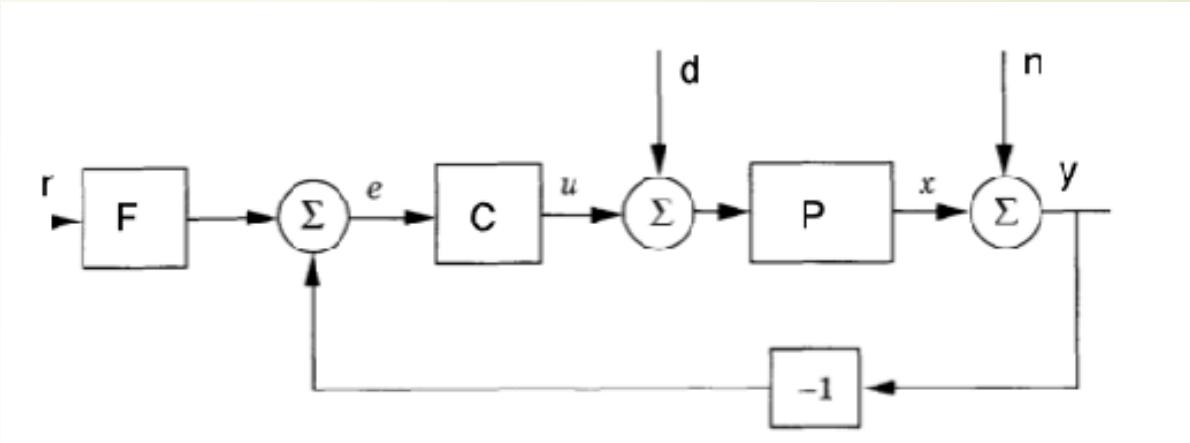
$$T_i = 3.57$$

$$T_d = 0.55$$

$$C(s) = 0.81 + \frac{1}{2.89s} + 0.45s$$

Controladores com dois graus de liberdade

$$F(s) = \frac{cT_i T_d s^2 + b s T_i + 1}{T_i T_d s^2 + s T_i + 1}$$



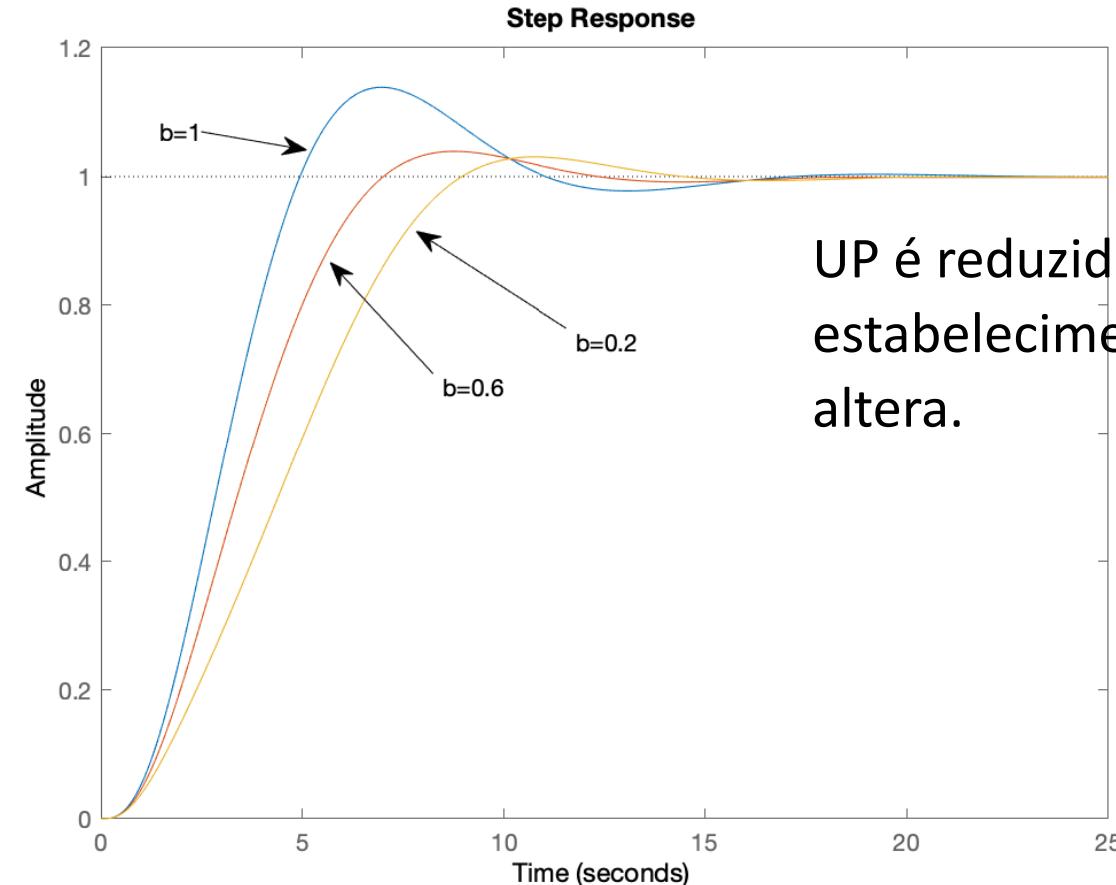
Analisaremos agora o efeito das escolhas de b e c

$$u(t) = K(e_p + \frac{1}{T_i} \int_0^t e(s)ds + T_D \frac{de_d}{dt}) \quad e_p = br - y$$

Variando b , variamos o efeito da mudança da referência no efeito proporcional:

Controladores com dois graus de liberdade

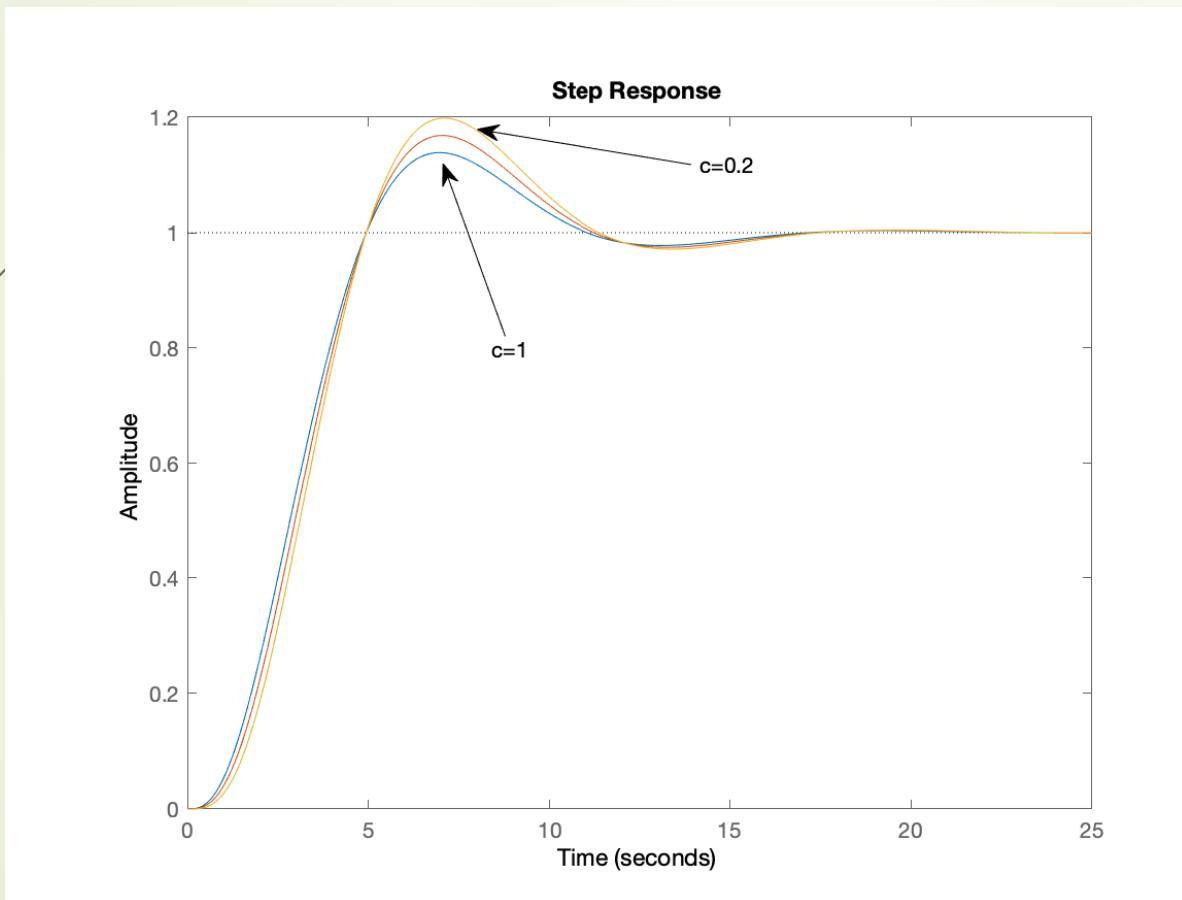
A redução de b (com c=1), reduz a sobreelevação.



UP é reduzido e o tempo de estabelecimento quase não se altera.

Controladores com dois graus de liberdade

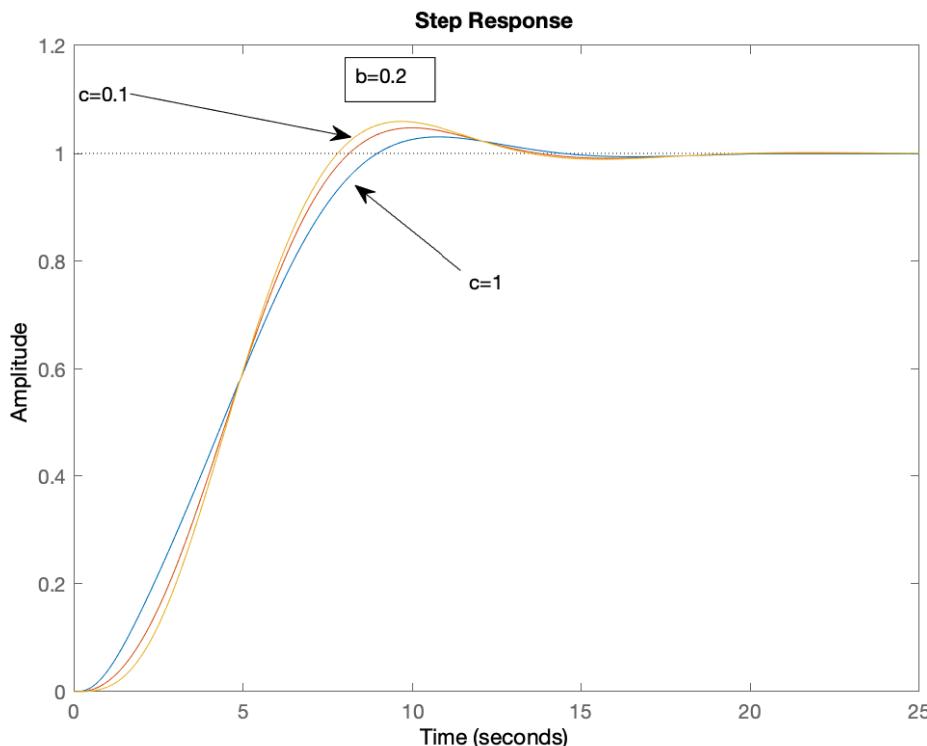
A redução de c (com $b=1$), tem pouco efeito na sobreelevação neste caso.



$$c=[1, 0.6, 0.2]$$

Controladores com dois graus de liberdade

Agora usou-se $b=0.2$ variando c .



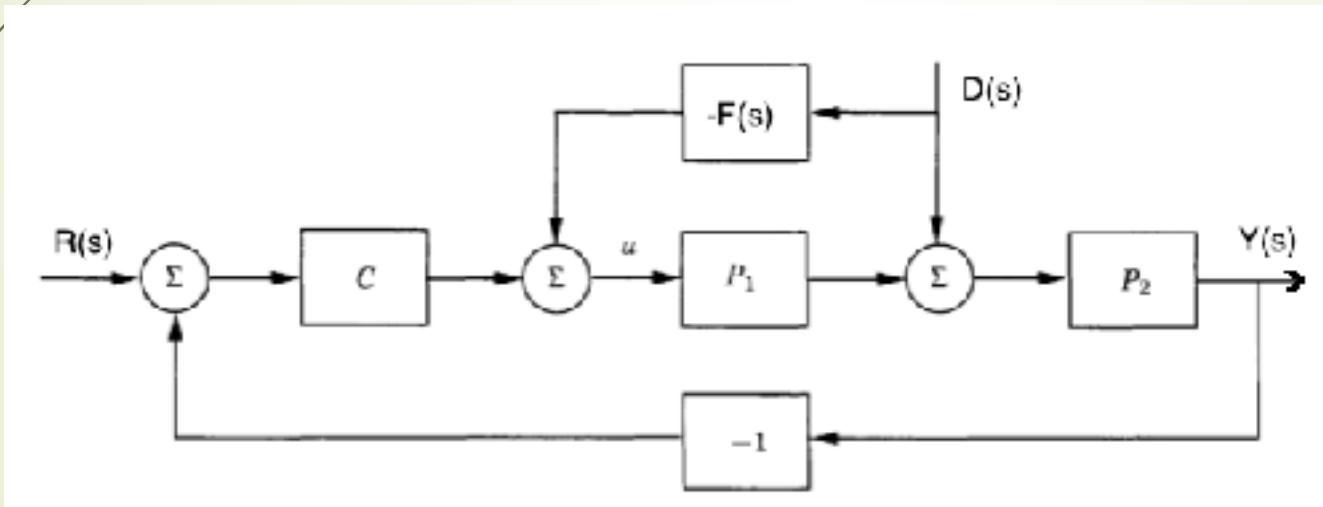
$$b=0.2$$
$$c=[1, 0.4, 0.1]$$

Reducir o valor de c pode reduzir o tempo de subida.

Controladores com dois graus de liberdade

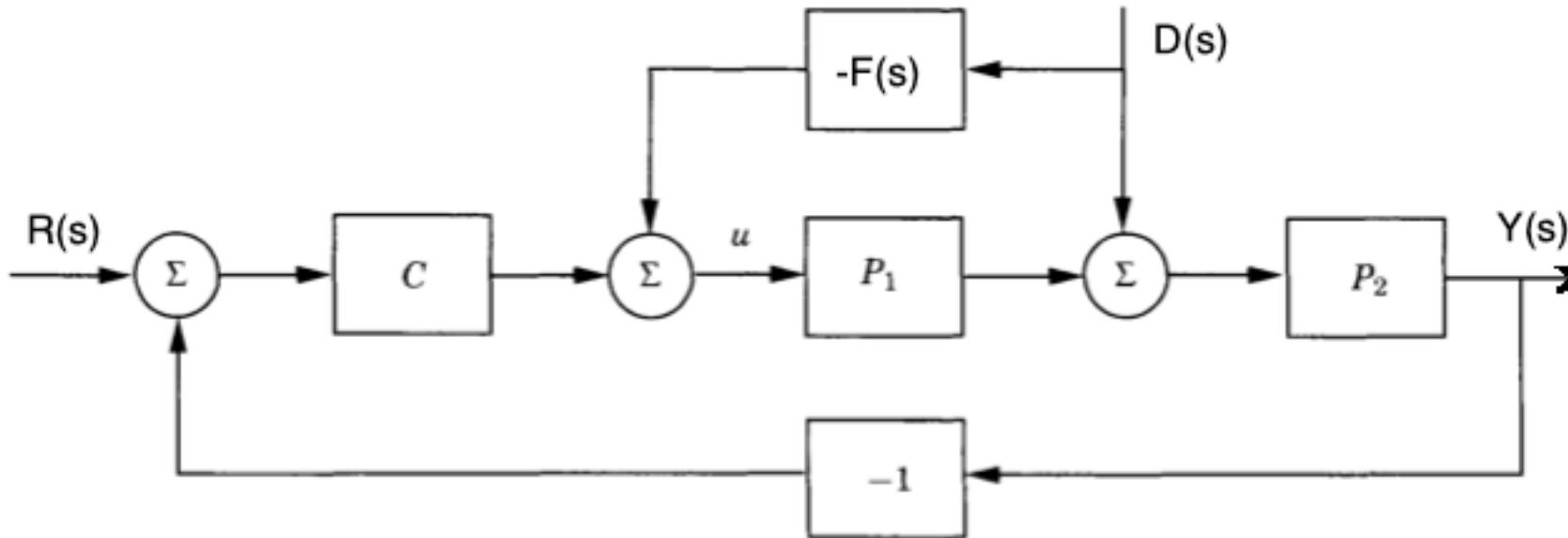
Feedforward: Cancelamento de um distúrbio medido

A proposta agora é diferente: a entrada de um dos controladores é o sinal de distúrbio, o qual assume-se que pode ser medido.



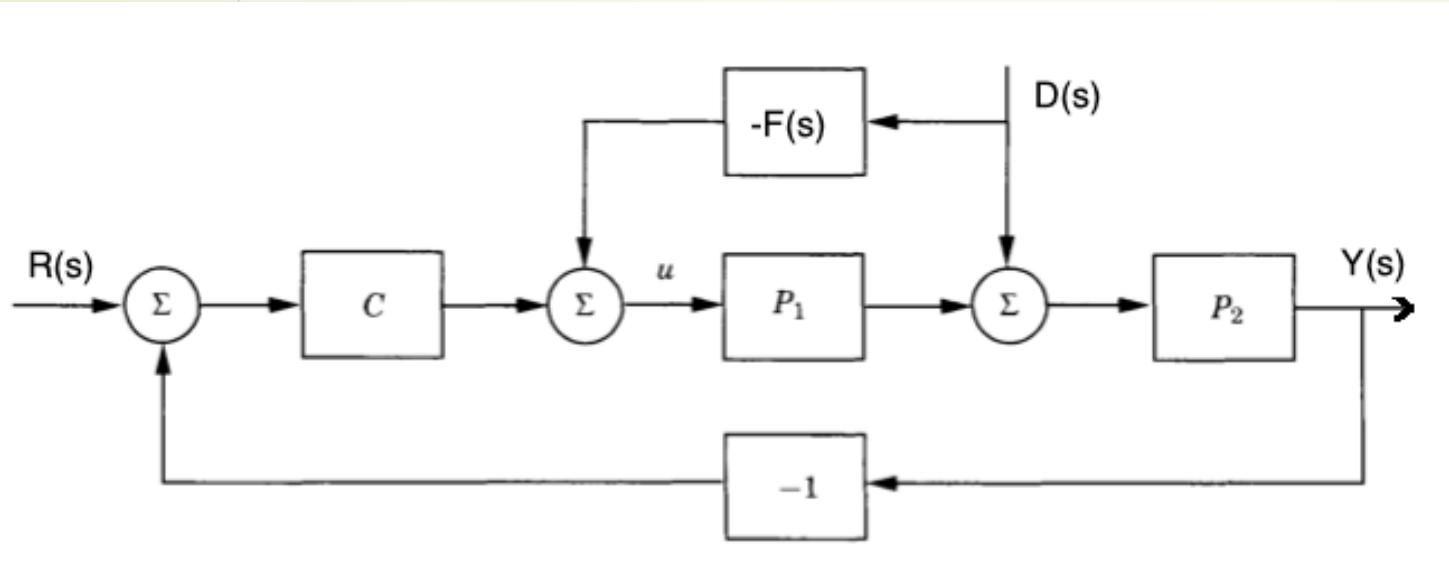
Controladores com dois graus de liberdade

Feedforward: Cancelamento de um distúrbio medido



O distúrbio D afeta a saída Y , e busca-se um compensador $F(s)$ que gere um sinal somado ao sinal de controle para cancelar o efeito de D .

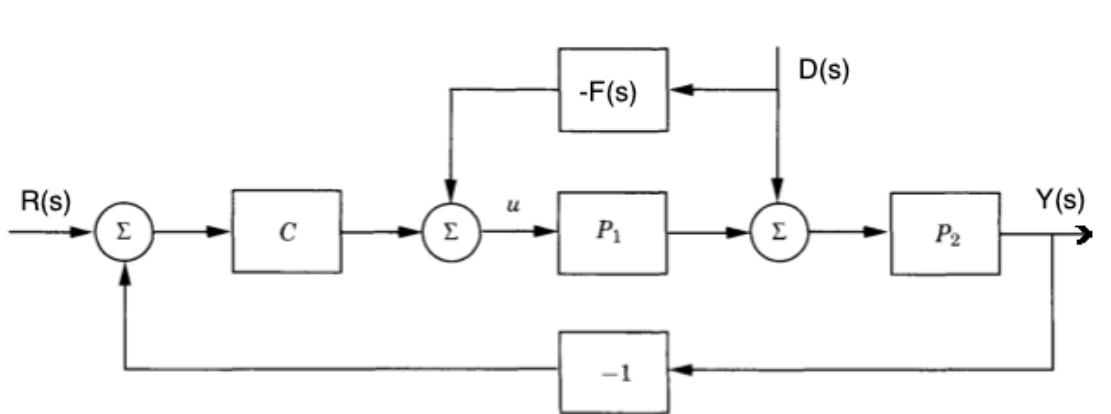
Controladores com dois graus de liberdade



A FT que relaciona D com Y é dada por
$$\frac{Y(s)}{D(s)} = \frac{P_2(1 - P_1 F)}{1 + PC} = P_2(1 - P_1 F)S$$

$S = \frac{1}{1 + PC}$, a função de sensibilidade.

Controladores com dois graus de liberdade



$$\frac{Y(s)}{R(s)} = \frac{CP_1P_2}{1 + CP_1P_2}$$

A FT que relaciona D com Y é dada por $\frac{Y(s)}{D(s)} = \frac{P_2(1 - P_1F)}{1 + PC} = P_2(1 - P_1F)S$

$S = \frac{1}{1 + PC}$, a função de sensibilidade e $P = P_1P_2$.

As duas forma de reduzir o efeito de $D(s)$ são:

- Fazer $1 - P_1F$ pequeno
- Fazer PC grande

O compensador ideal é $F = P_1^{-1}$

Controladores com dois graus de liberdade

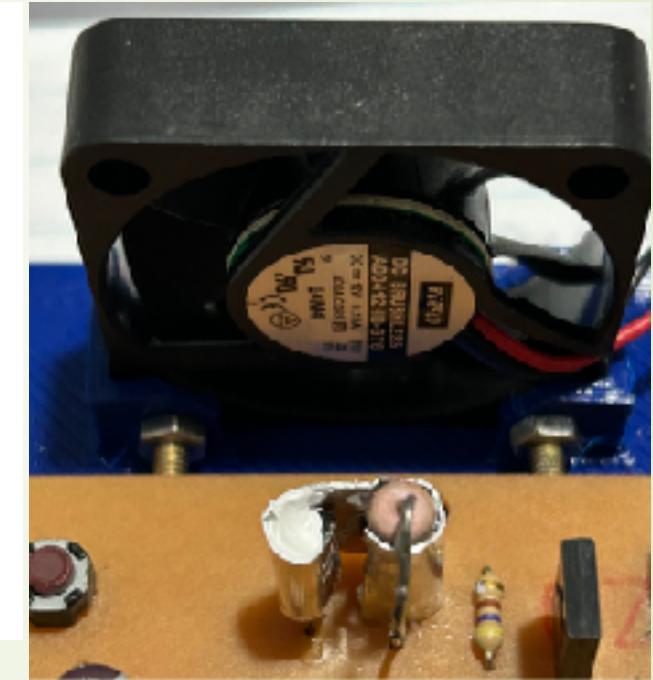
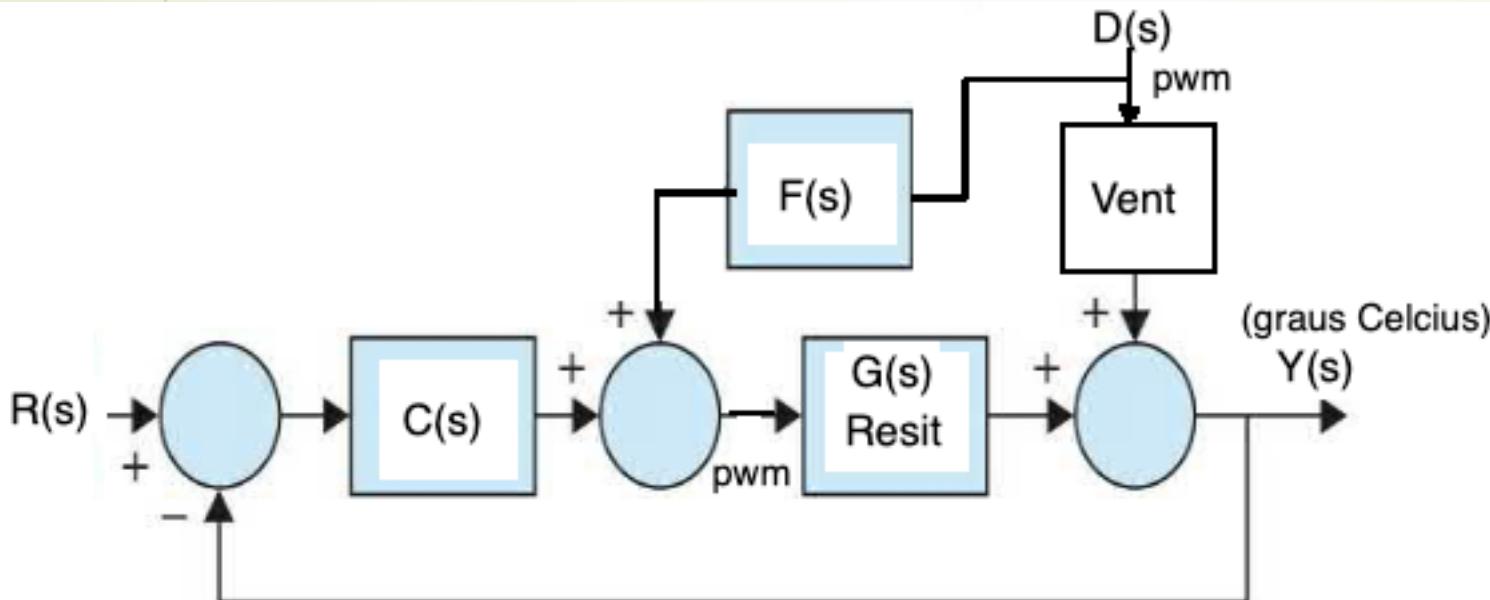
$$\frac{Y(s)}{D(s)} = \frac{P_2(1 - P_1F)}{1 + P_1P_2C} \quad \frac{Y(s)}{R(s)} = \frac{CP_1P_2}{1 + CP_1P_2}$$

Novamente, o controlador C afeta os polos da equação característica.

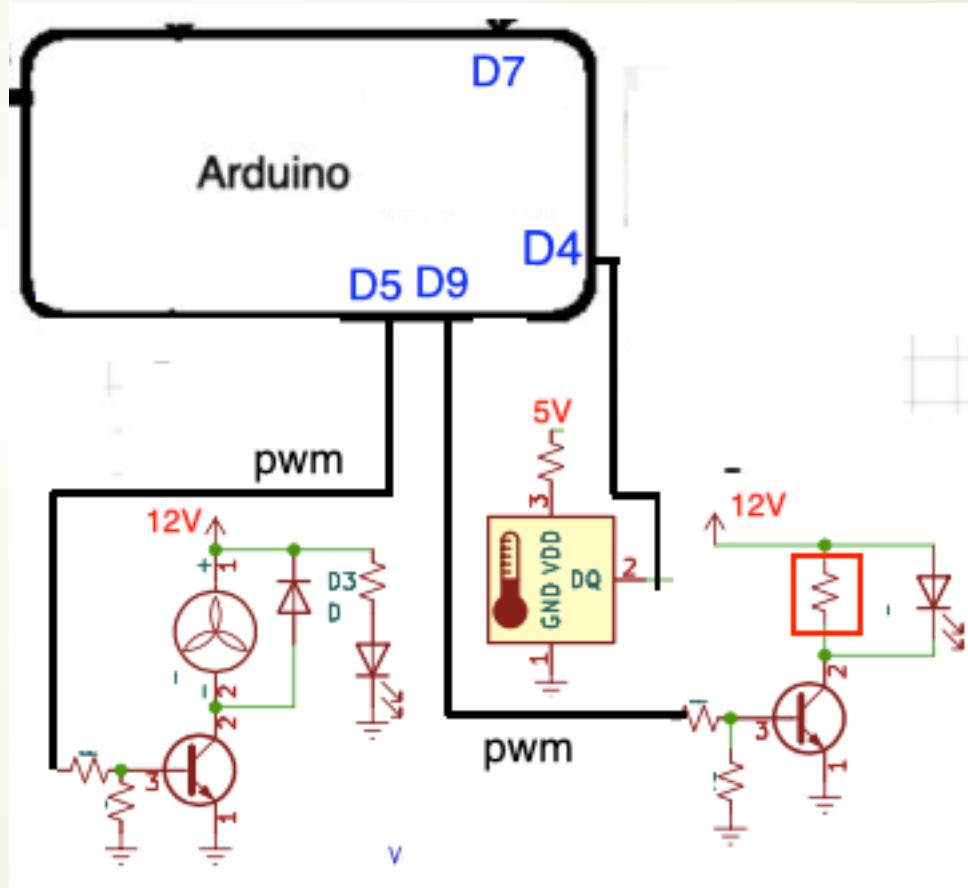
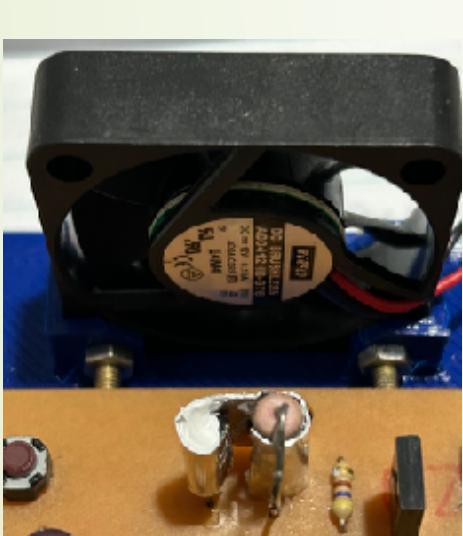
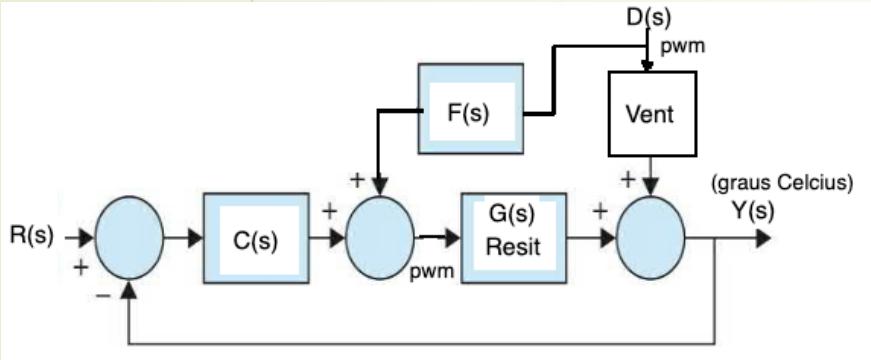
O controlador F não afeta a relação de R para Y, mas sim, de D para Y.

Controladores com dois graus de liberdade

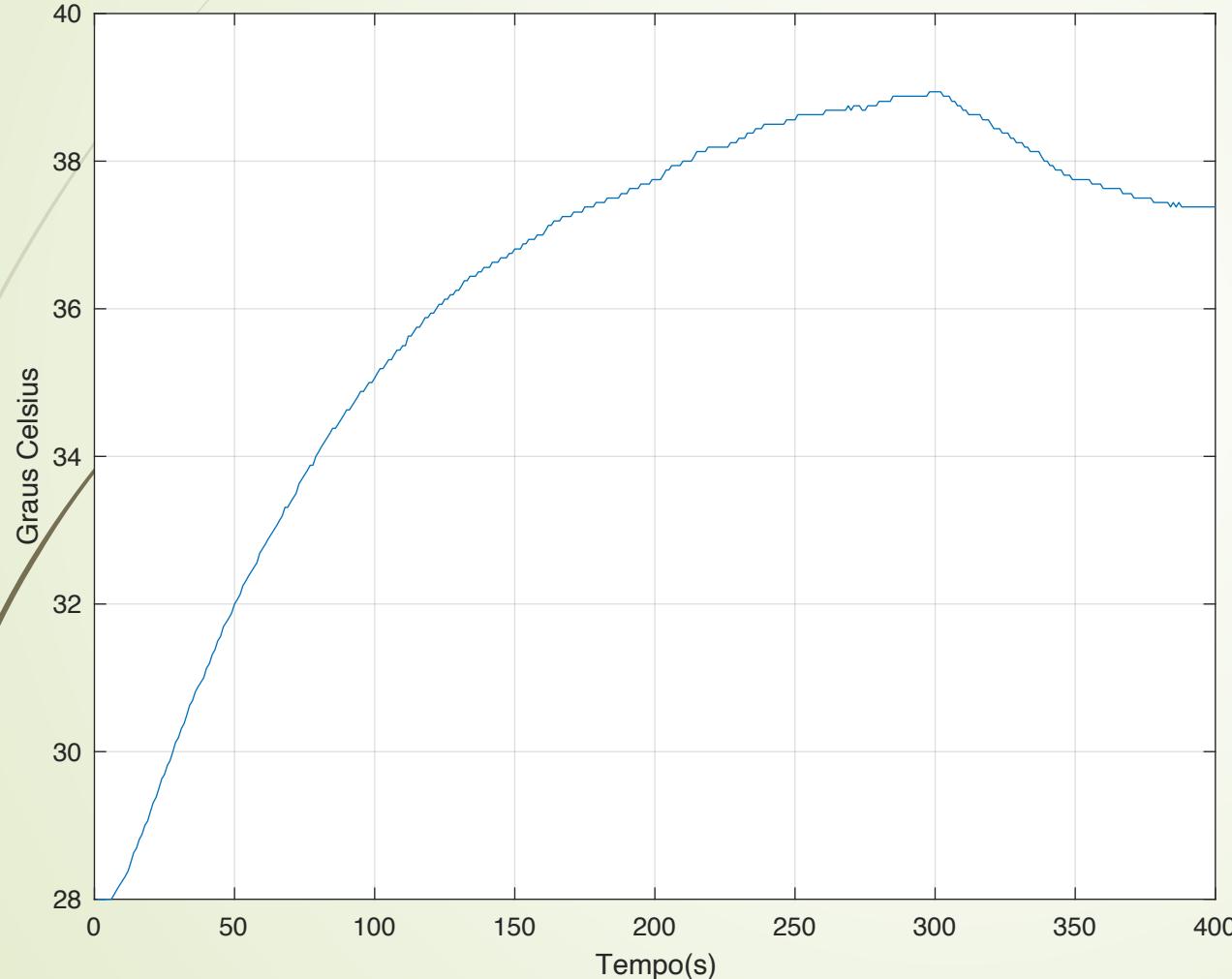
Exemplo 4: usaremos o sistema de controle de temperatura de Laboratório de Controle.



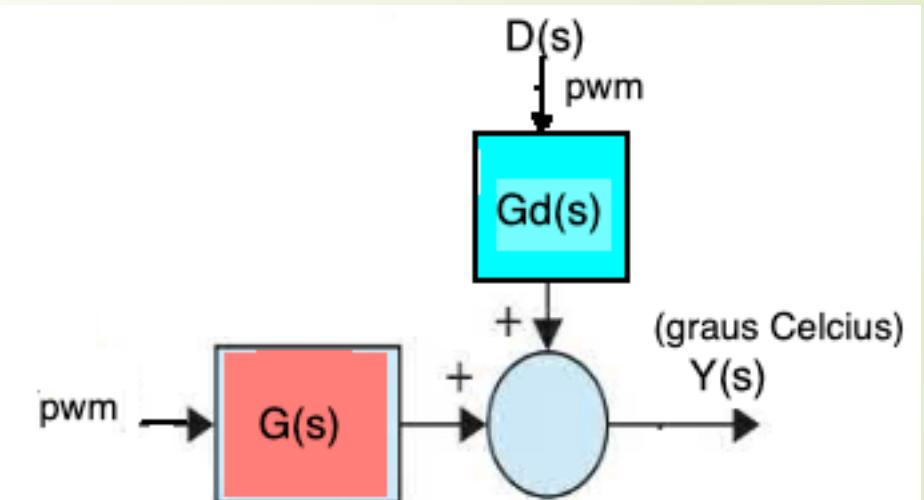
Controladores com dois graus de liberdade



Controladores com dois graus de liberdade

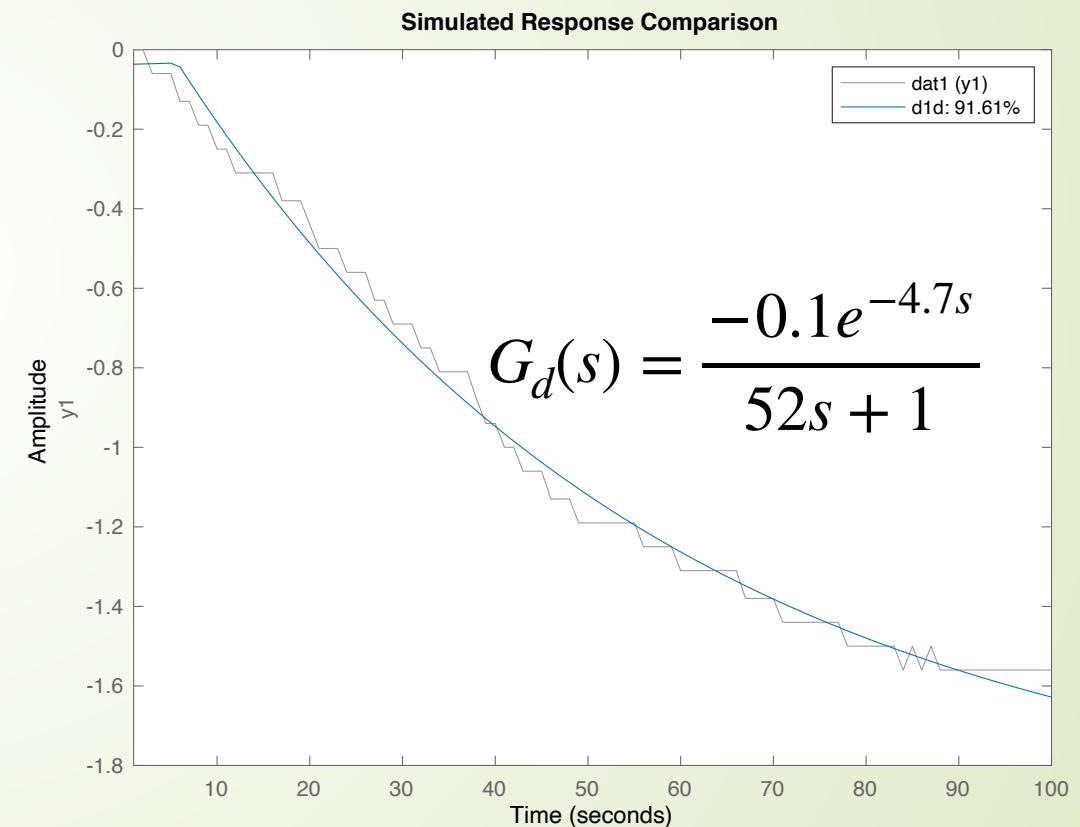
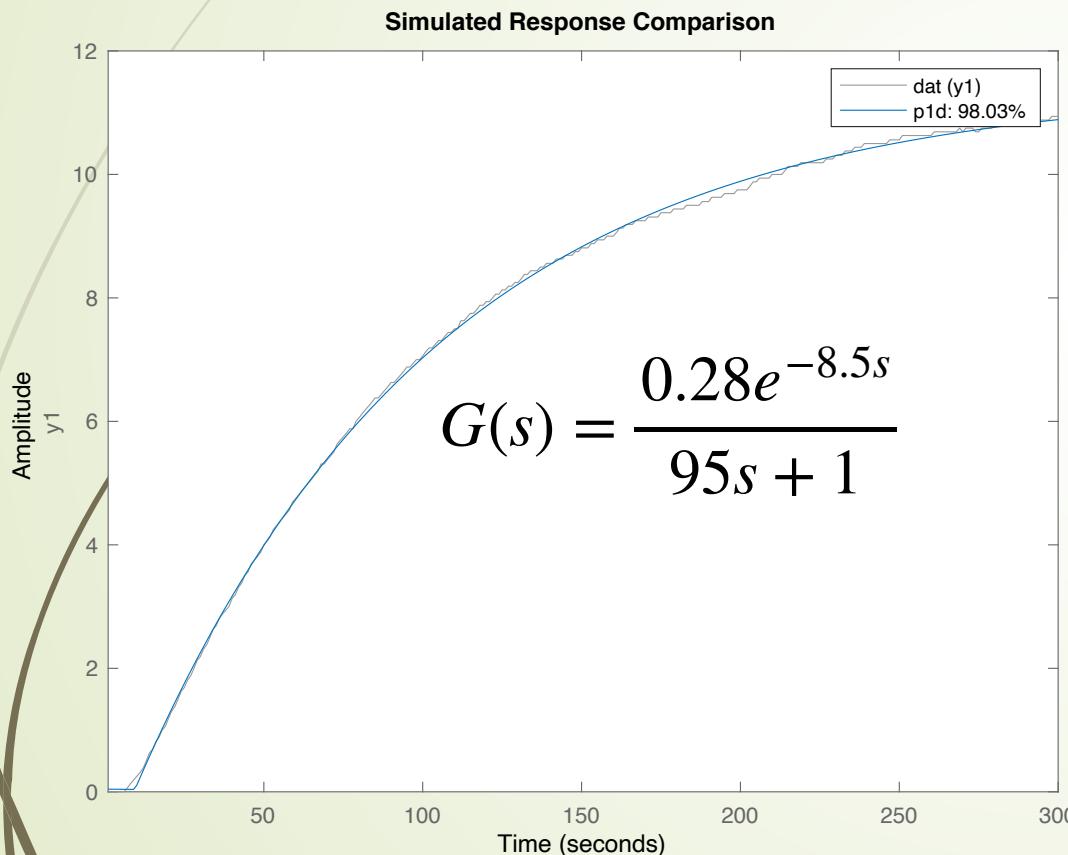


Resposta em malha aberta para um degrau de amplitude 40 na entrada e um distúrbio de amplitude 20 em D no instante 300.



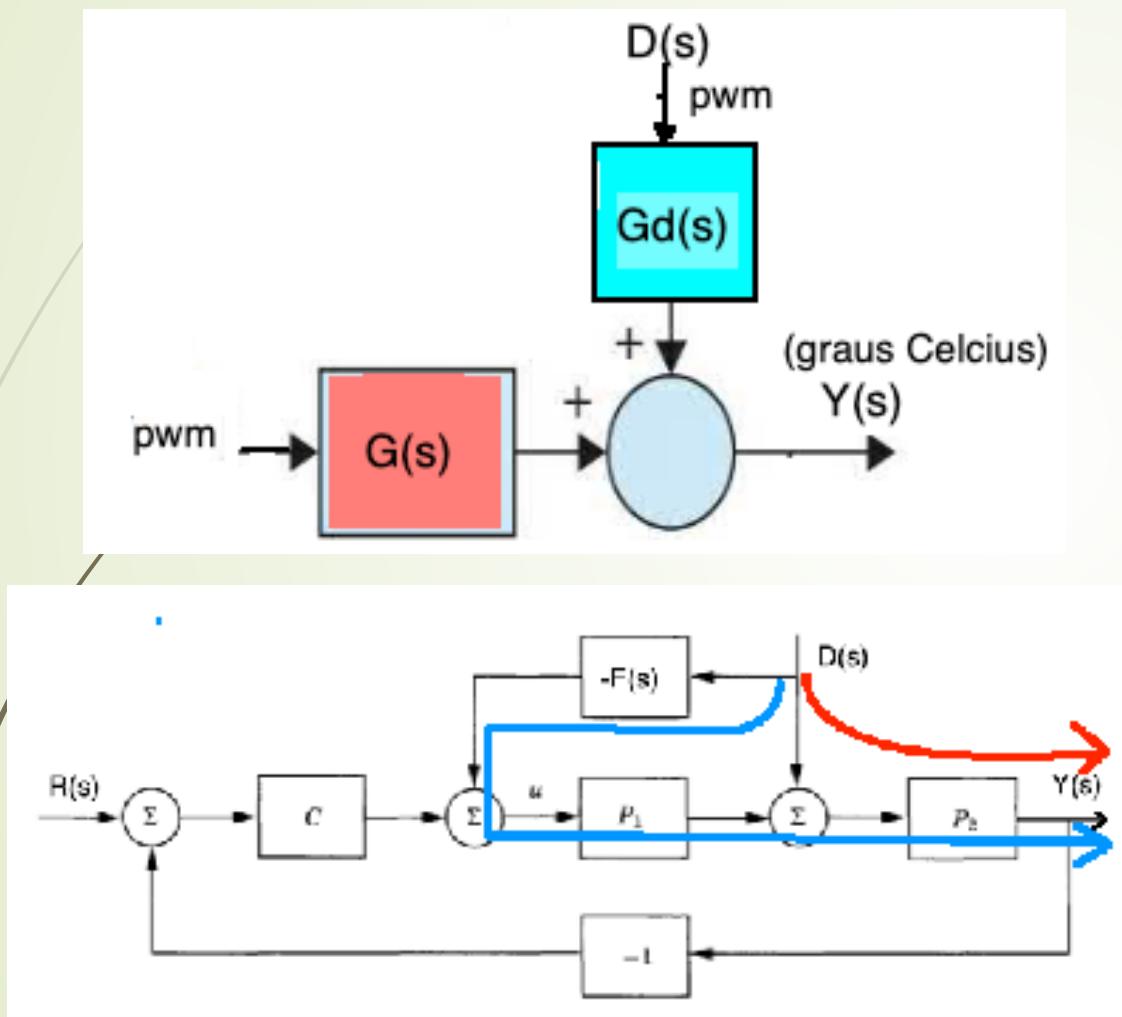
Controladores com dois graus de liberdade

Resposta em malha aberta para um degrau de amplitude 40 na entrada e um distúrbio de amplitude 20 em D no instante 300.



Modelos estimados usando os dados

Controladores com dois graus de liberdade



$$G_d(s) = \frac{-0.1e^{-4.7s}}{52s + 1}$$
$$G(s) = \frac{0.28e^{-8.5s}}{95s + 1}$$

Usar $G(s)$ e $G_d(s)$ para fazer o projeto.

$$\frac{Y(s)}{D(s)} = \frac{P_2(1 - P_1F)}{1 + PC}$$

O sinal de controle U está atrasado em relação do distúrbio D !

Melhor chegar atrasado do que não chegar!_!

Controladores com dois graus de liberdade

$$\frac{Y(s)}{D(s)} = \frac{P_2(1 - P_1 F)}{1 + PC}$$

$$P_1 P_2 = G$$

$$P_2 = G_d$$

$$P_1 G_d = G$$

$$P_1 = G G_d^{-1}$$

$$F = P_1^{-1} = G^{-1} G_d = \frac{95s + 1}{0.28e^{-8.5s}} \frac{-0.1e^{-4.7s}}{52s + 1} = 0.35 \frac{95s + 1}{52s + 1} e^{-4.7s + 8.5s}$$

$$F = 0.35 \frac{95s + 1}{52s + 1}$$

$$G(s) = \frac{0.28e^{-8.5s}}{95s + 1}$$

$$G_d(s) = \frac{-0.1e^{-4.7s}}{52s + 1}$$

Eliminar $e^{3.8s}$, pois avanço não pode ser implementado.

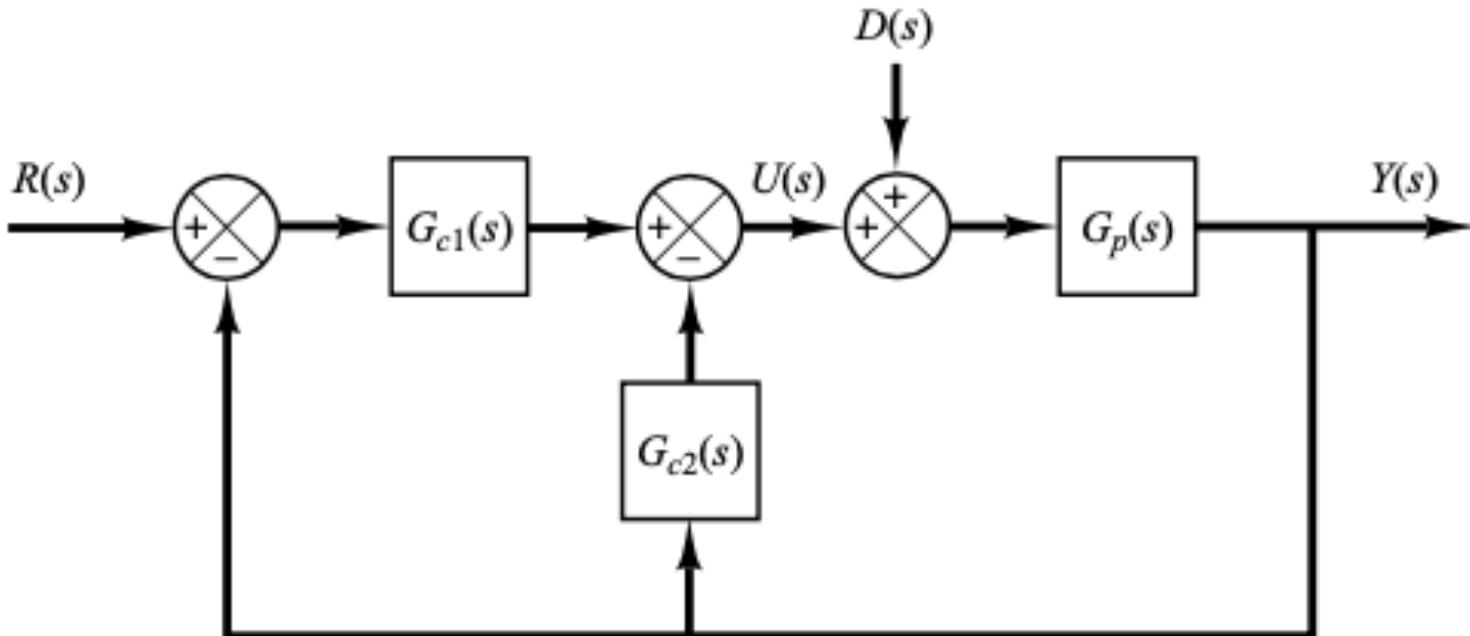
Controladores com dois graus de liberdade

Nesta configuração um dos controladores foi calculado especificamente para reduzir o efeito do distúrbio.

Nas outras configurações, um dos controladores foi calculado para melhorar a resposta da referência R para a saída Y.

Controladores com dois graus de liberdade

Abaixo mais uma configuração com dois graus de liberdade



$$\frac{Y(s)}{R(s)} = \frac{G_{c1}G_p}{1 + (G_{c1} + G_{c2})G_p}$$

$$\frac{Y(s)}{D(s)} = \frac{G_p}{1 + (G_{c1} + G_{c2})G_p}$$

Controladores com dois graus de liberdade

Nesta configuração o controlador G_{c1} aparece apenas na relação Y/R, e pode ser utilizado para melhorar a resposta de Y em relação a R.

$$\frac{Y(s)}{R(s)} = \frac{G_{c1}G_p}{1 + (G_{c1} + G_{c2})G_p}$$

$$\frac{Y(s)}{D(s)} = \frac{G_p}{1 + (G_{c1} + G_{c2})G_p}$$

Observe que nesta configuração, os dois compensadores G_{c1} e G_{c2} aparecem na equação característica.

Controladores com dois graus de liberdade

Resumindo:

Nas configurações com dois graus de liberdade, em geral um dos controladores é utilizado para definir os polos da malha fechada da equação característica, enquanto o outro é utilizado para melhorar a resposta em relação à entrada de referência ou ao distúrbio.

Há muitas configurações diferentes que podem ser exploradas.