

## Dúvidas quanto ao Projeto de Controlador PI usando Root Locus e contribuição angular

email enviado em 11 de junho de 2020 18:45  
(suprimidas as informações dos autores do email)

Estou enviando um PDF com os nossos comandos do matlab, bem como os gráficos gerados a cada passo para que o senhor possa entender nosso processo de desenvolvimento do PI. Nossa dúvida está na posição do zero do controlador, o qual achamos por contribuição angular. No arquivo, há dois tipos de tentativa: a primeira seguindo à risca o que estava no material do senhor quanto ao processo de achar o local do zero, e a segunda mudando a equação das somatórias dos polos e zeros, como o senhor poderá ver no arquivo.

Gostaríamos de saber como podemos proceder nesse caso, e se ele se encaixa na observação feita pelo senhor no material sobre amplitudes excessivas para a ação de controle.

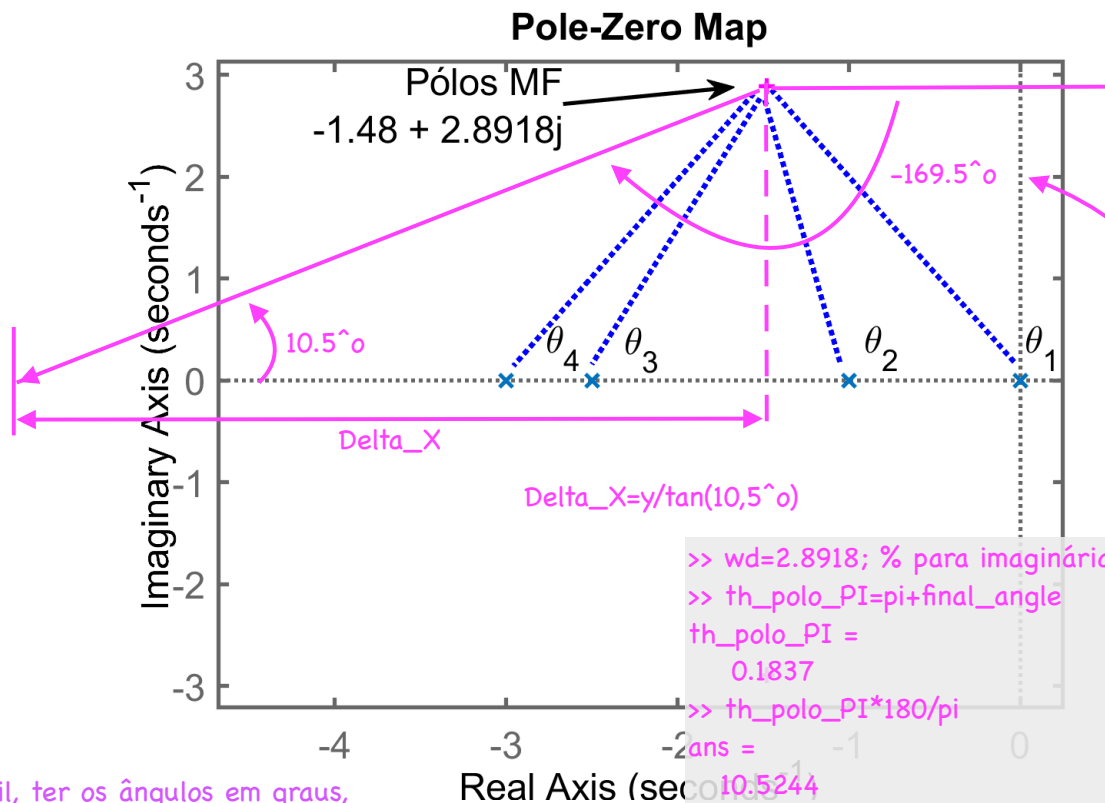
Segue arquivo original (PDF) acrescido de comentários e explicações ↓

```

zpk(G)
ans =
    7.5
-----
(s+3) (s+2.5) (s+1)
>> ts=2.7;
>> OS=20;
>> zeta
zeta =
    0.4559
>> wn=4/(zeta*ts)
wn =
    3.2492
>> sigma=wn*zeta
sigma =
    1.4815
>> wd=wn*sqrt(1-zeta^2)
wd =
    2.8918
>> polos_MF_pi=[-sigma+i*wd -sigma-i*wd]
polos_MF_pi =
    -1.4815 + 2.8918i  -1.4815 - 2.8918i
PI_aux=tf([7.5],poly([0 -1 -2.5 -3]));
>> zpk(PI_aux)
ans =
    7.5
-----
s (s+3) (s+2.5) (s+1)
>> figure;pzmap(PI_aux)
>> axis([-7 1 -3.5 3.5])
>> hold on
>> plot(polos_MF_pi,'m+')

```

Até aqui, parece Ok



```

>> wd=2.8918; % para imaginária (y)
>> th_polo_PI=pi+final_angle
th_polo_PI =
    0.1837
>> th_polo_PI*180/pi
ans =
    10.5244
>> Delta_X=wd/tan(th_polo_PI)
Delta_X =
    15.5658
>> sigma=1.4815;
>> polo_PI= -sigma -Delta_X
polo_PI =
   -17.0473

```

Seria útil, ter os ângulos em graus, para confirmar se faz sentido...

```

>> th1=atan2(wd,-sigma)
th1 =
    2.0442
>> th1*180/pi
ans =
    117.1240
>> th2=atan2(wd,-0.4815)
th2 =
    1.7358
>> th2*180/pi
ans =
    99.4540
>> th3=atan2(wd,1.0185)
th3 =
    1.2322
>> th3*180/pi
ans =
    70.5999
>> th4=atan2(wd,1.5185)
th4 =
    1.0873
>> th4*180/pi
ans =
    62.2977
>> sum_th_polos=th1+th2+th3+th4
sum_th_polos =
    6.0995
>> sum_th_polos*180/pi
ans =
    349.4756

```

**%TENTATIVA 1: Seguindo a fórmula que está no material**

```

>> sum_th_zero=pi+sum_th_polos
sum_th_zero =
    9.2410

```

```

Note:
>> 9.2410*180/pi
ans =
    529.4703 % graus
>> ans-360 % ou:
ans =
    169.4703 % graus
>>

```

```

>> final_angle=pi-sum
final_angle =
   -2.9579
>> final_angle*180/pi
ans =
   -169.4756

```

aprox. mesmo valor, a questão é como usar este ângulo no RL !

```

>>delta_x=wd/tan(sum_th_zero)
delta_x =
-15.5620    Ok
>> zero_pi=-sigma-delta_x
zero_pi =
    14.0805    %VALOR POSITIVO??
>> PI_ang=tf([1 -zero_pi],[1 0])
PI_ang =
s - 14.08
-----
s
>> ftma_PI_ang=PI_ang*G;
>>zpk(ftma_PI_ang)
ans =
    7.5 (s-14.08)
-----
s (s+3) (s+2.5) (s+1)
>> figure;rlocus(ftma_PI_ang)
>> axis([-5 15 -3.5 3.5])
>> hold on
>> plot(polos_MF_pi,'m+')
>> sgrid(zeta,wn)
>> [K_PI_ang, polos_MF]=rlocfind(ftma_PI_ang)
Select a point in the graphics window
selected_point =
-1.4809 + 2.8889i
K_PI_ang =
    0.8009
polos_MF =
-5.1073 + 0.0000i
-1.4818 + 2.8890i
-1.4818 - 2.8890i
1.5708 + 0.0000i

```

Aqui, começam os erros....

Vocês devem considerar como calcularam sigma (seu sinal) e o que acontece graficamente no RL. O sigma de vocês é positivo! O Delta\_X é negativo. Observem a figura com as anotações que fiz e vão perceber que a maneira correta de calcular a posição do pólo do PI seria, no caso de vocês aqui, seria fazer:

>> zero\_pi = -sigma + delta\_x

↑  
O único erro!

Pela regra de ponto pertencente ao RL, pode ser:

$$\sum \theta_{\text{Zeros}} - \sum \theta_{\text{Pólos}} = \pm 180^\circ (2k+1)$$

então vocês podem fazer:

$$\sum \theta_{\text{Zeros}} - \sum \theta_{\text{Pólos}} = +180^\circ \quad (1)$$

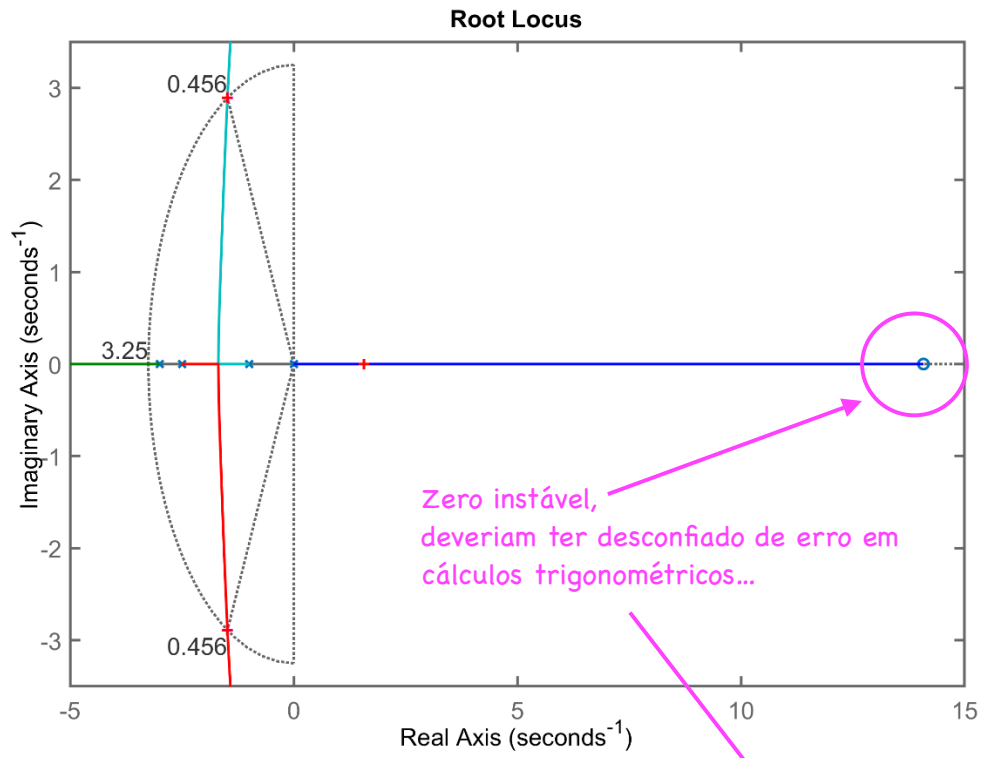
ou

$$\sum \theta_{\text{Zeros}} - \sum \theta_{\text{Pólos}} = -180^\circ \quad (2)$$

No caso do material, foi usada a opção (1), e isolando o  $\theta_{\text{Zero\_PI}}$ , resultou na equação:

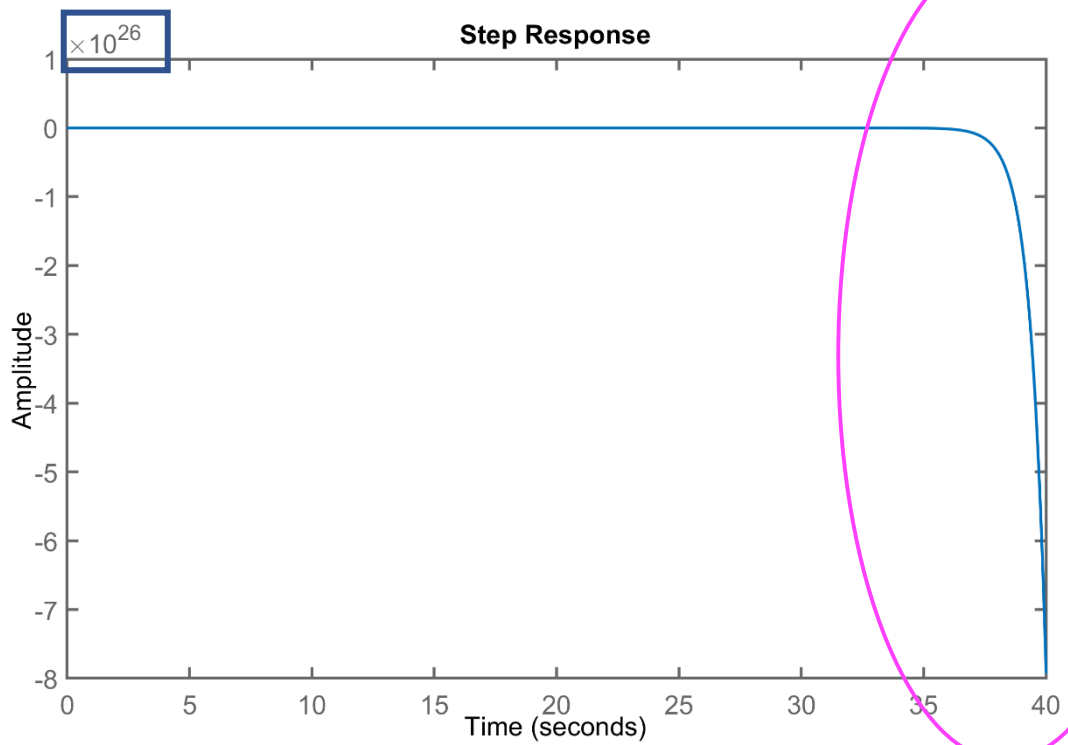
$$\theta_{\text{Zero\_PI}} = 180^\circ + \sum \theta_{\text{Pólos}}$$

Obs.: No Matlab, todos os cálculos de ângulos devem ser feitos em radianos.



```
>> ftfm_PI_ang=feedback(K_PI_ang*ftma_PI_ang,1);
```

```
>> figure;step(ftmf_PI_ang)
```



## %TENTATIVA 2: Modificando a fórmula e fazendo:

```
>> sum_th_zero=pi-sum_th_polos      %Subtraindo (180 – ângulo dos polos), e não somando como antes
```

```
sum_th_zero =
```

```
-2.9579
```

```
>> delta_x=wd/tan(sum_th_zero)
```

```
delta_x =
```

```
15.5620
```

```
>> zero_pi=-sigma-delta_x
```

```
zero_pi =
```

```
-17.0435
```

```
>> PI_ang=tf([1 -zero_pi],[1 0])
```

```
PI_ang =
```

```
s + 17.04
```

```
-----
```

```
s
```

Continuous-time transfer function.

```
>> ftma_PI_ang=PI_ang*G;
```

```
>> zpk(ftma_PI_ang)
```

```
ans =
```

```
7.5 (s+17.04)
```

```
-----
```

```
s (s+3) (s+2.5) (s+1)
```

Continuous-time zero/pole/gain model.

```
>> figure;rlocus(ftma_PI_ang)
```

```
>> sgrid(zeta,wn)
```

```
>> plot(polos_MF_pi,'m+')
```

```
>> hold on
```

```
>> plot(polos_MF_pi,'m+')
```

```
>> [K_PI_ang,polos_MF]=rlocfind(ftma_PI_ang)
```

Select a point in the graphics window

```
selected_point =
```

```
-0.2984 + 0.5770i
```

```
K_PI_ang =
```

```
0.0295
```

Ok, correto!

Ok

polos\_MF =

-2.9620 + 0.6350i

-2.9620 - 0.6350i

mas...

Esqueceram de um detalhe:

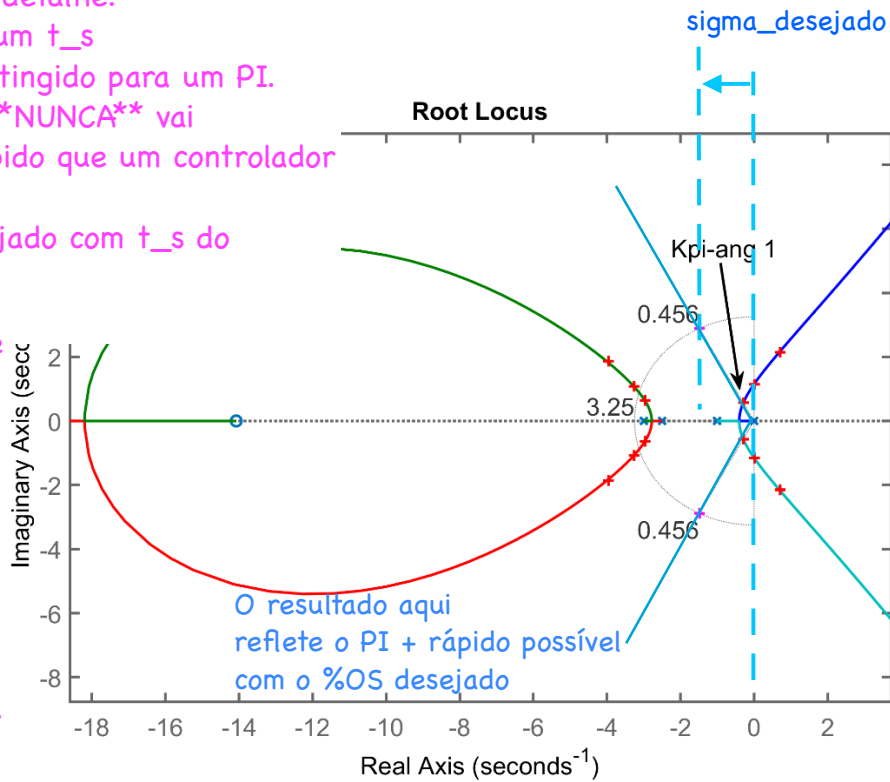
este sigma é para um  $t_s$  impossível de ser atingido para um PI.

Notar que um PI, **\*\*NUNCA\*\*** vai responder mais rápido que um controlador Proporcional.

Comparar  $t_s$  desejado com  $t_s$  do proporcional.

Isto quer dizer que analiticamente, não é possível incluir um PI na malha e obter o  $t_s$  desejado.

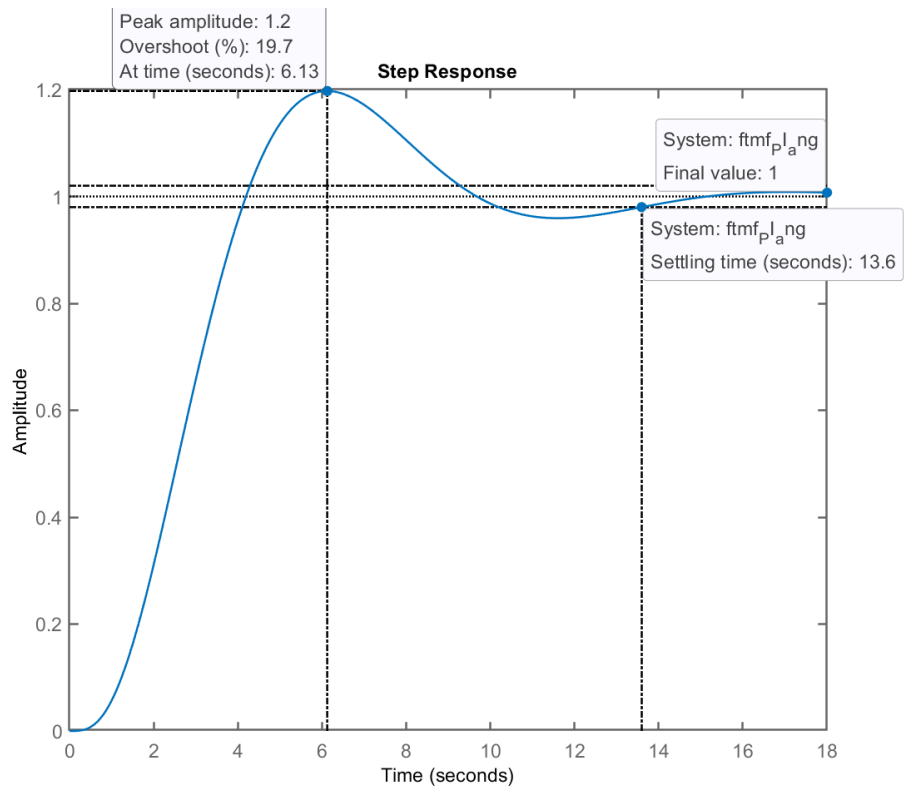
No máximo o %OS desejado ( $K_{PI}$  na linha guia do  $\zeta$  - Ok)



Os outros pontos de ganho indicados em vermelho não foram mostrados pois o gráfico ao aplicar o step não se assemelhou ao que era esperado.

```
>> ftmf_PI_ang=feedback(K_PI_ang*ftma_PI_ang,1);  
>> ftmf_PI_ang2=feedback(K_PI_ang2*ftma_PI_ang,1);  
>> ftmf_PI_ang3=feedback(K_PI_ang3*ftma_PI_ang,1);  
>> figure;step(ftmf_PI_ang)
```

???



ok