Prova 1 (2025/1)

Rascunhos para a prova

Suponha a planta:

```
G=tf(1,poly([-6 -3 -1]));
zpk(G)

          1
  -----------------
  (s+6) (s+3) (s+1)

Continuous-time zero/pole/gain model.

dcgain(G)
ans =
     0.055556
1/(6*3)
ans =
     0.055556
G=tf(6*3*1,poly([-6 -3 -1]));
zpk(G)

         18
  -----------------
  (s+6) (s+3) (s+1)

Continuous-time zero/pole/gain model.
```

Planta usada:

$$G(s) = \frac{18}{(s+1)(s+3)(s+6)}$$

Frequências de cada pólo:

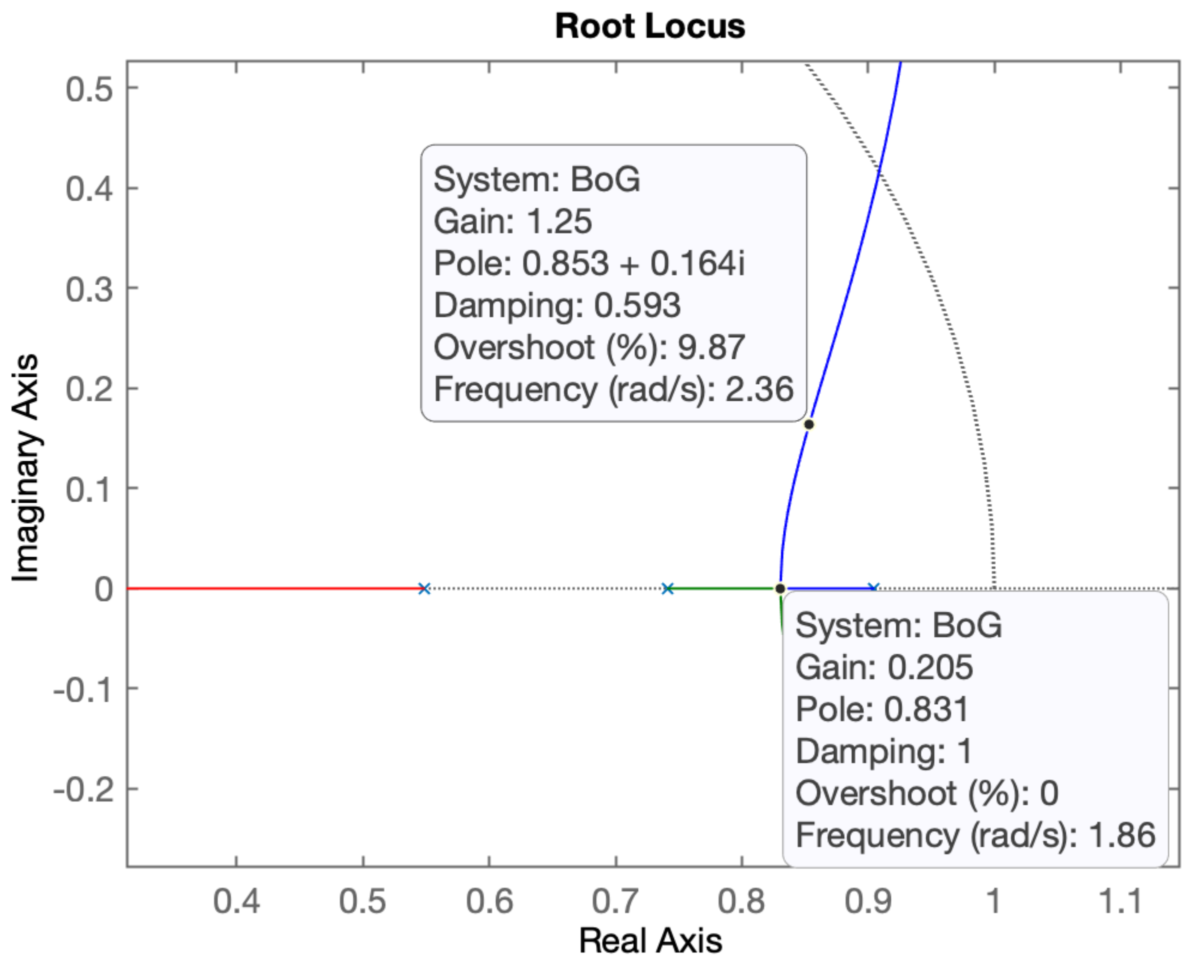
```
polos=pole(G)
polos =
           -6
           -3
           -1
freq=polos./(2*pi)
freq =
     -0.95493
     -0.47746
     -0.15915
```

Freq do pólo + rápido, quase 1 Hz --> Amostrar 10x mais rápido:

```
fs=10;
T=1/fs
T =
          0.1
BoG=c2d(G,T);
zpk(BoG)

   0.0023485 (z+2.924) (z+0.2074)
  --------------------------------
  (z-0.9048) (z-0.7408) (z-0.5488)

Sample time: 0.1 seconds
Discrete-time zero/pole/gain model.
```

Projeto Controlador Proporcional

```
rlocus(BoG)
% realizando zoom na área de interesse
```



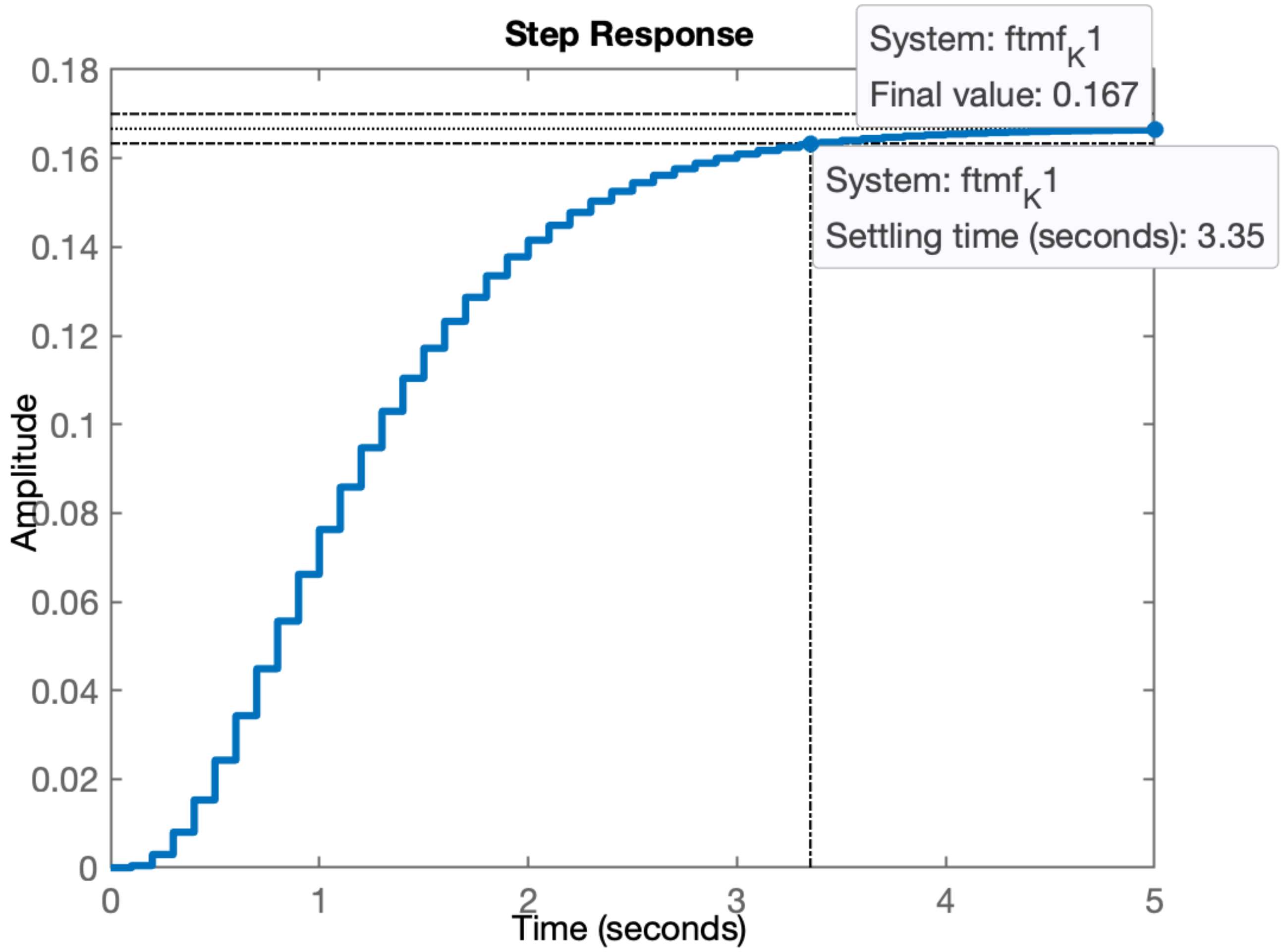
Requisitos resposta:

- Criticamene amortecida ($\zeta = 1$), 2 pólos complexos dominantes iguais, no ponto de break-out do RL, Resulta ganho de $K = 0,2$;

- Sub-amortecida ($0 < \zeta < 1$), para overshoot de 10%,
  Resulta ganho, $K \approx 1,25$.

Testando para avaliar erros em MF.

```
K1=0.2;  % resposta criticamente amortecida;
K2=1.25; % resposta subamortecida ($\%OS \approx 10\%$).

ftmf_K1=feedback(K1*BoG,1);
ftmf_K2=feedback(K2*BoG,1);
step(ftmf_K1)
% Hum... erro enorme em regime permanente
```

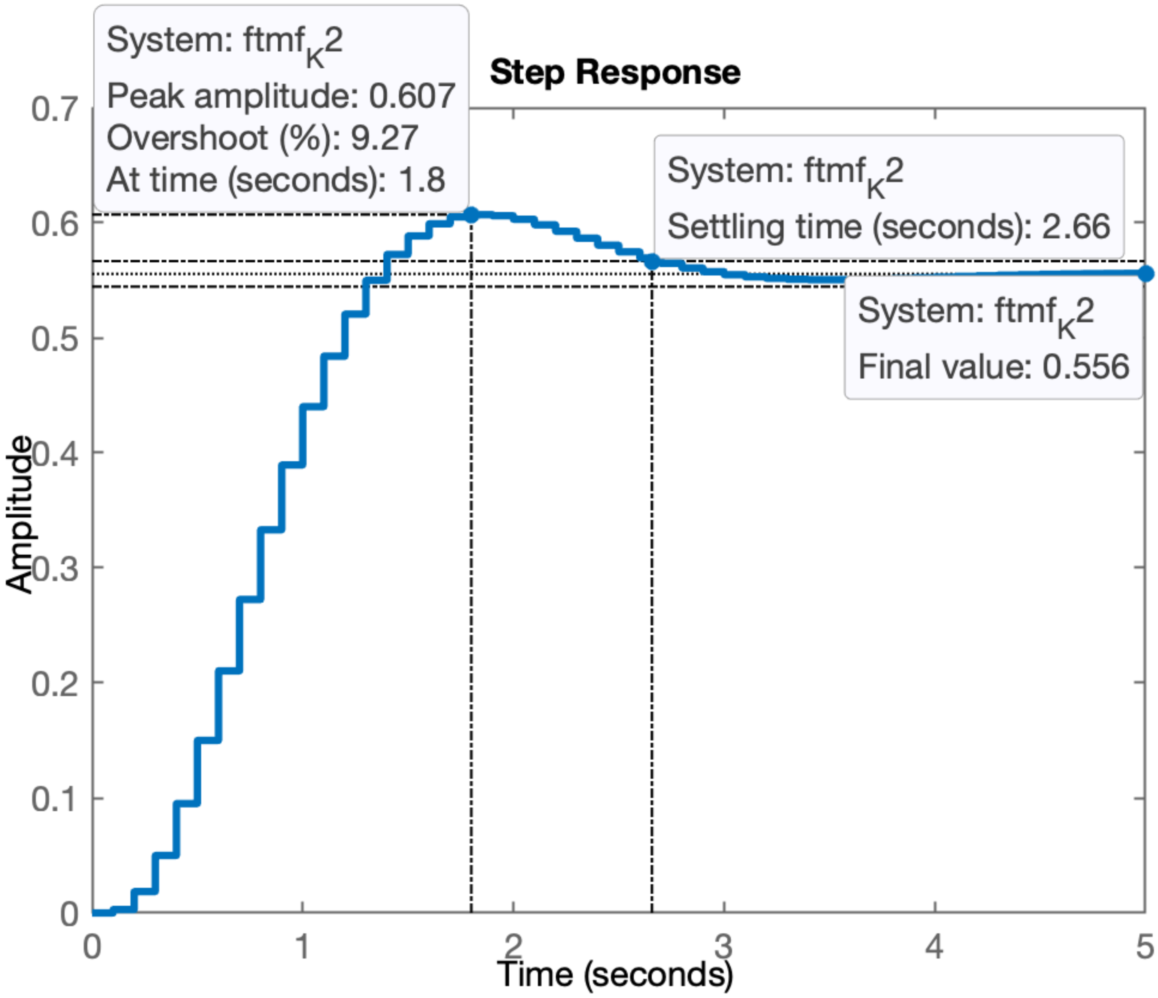



```
dcgain(ftmf_K1)
ans =
      0.16667
erro=(1-dcgain(ftmf_K1))/1*100
erro =
       83.333
stepinfo(ftmf_K1)
        RiseTime: 1.8
    SettlingTime: 3.4
     SettlingMin: 0.15033
     SettlingMax: 0.16667
       Overshoot: 0
```

```
      Undershoot: 0
            Peak: 0.16667
        PeakTime: 8.9
```

Conclusão: impraticável: erro muito grande (+83%).

```
figure; step(ftmf_K2)
```



O erro não fica muito menor:

```
dcgain(ftmf_K2)
ans =
      0.55556
erro=(1-dcgain(ftmf_K2))/1*100
erro =
       44.444
stepinfo(ftmf_K2)
        RiseTime: 0.8
    SettlingTime: 2.7
     SettlingMin: 0.5206
     SettlingMax: 0.60708
       Overshoot: 9.2737
      Undershoot: 0
            Peak: 0.60708
```

```
         PeakTime: 1.8
```

Para manter erro abaixo de 10%, qual seria o valor do ganho necessário? E qual seria a resposta obtida?

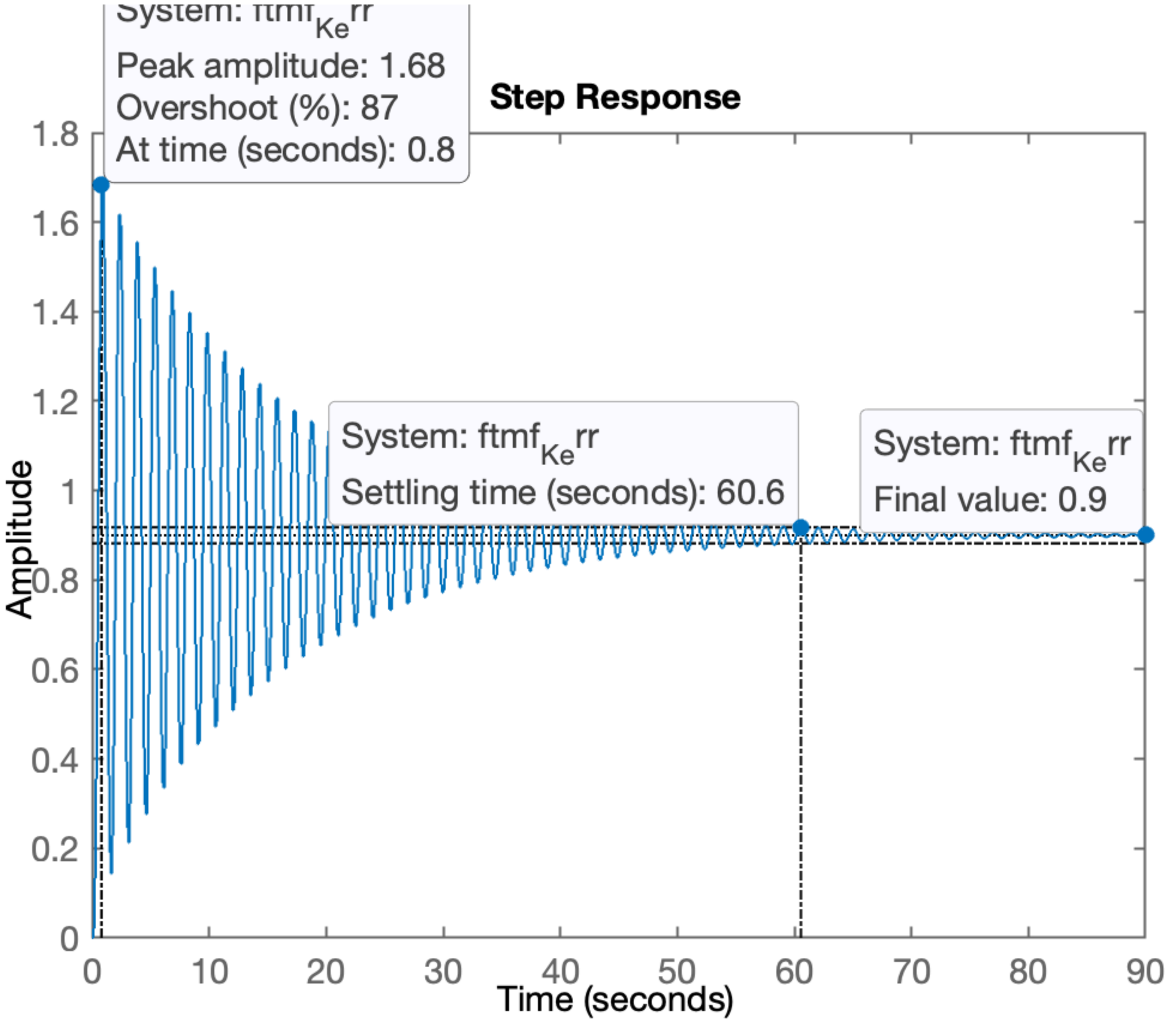
- Suponha ainda $\%OS \leq 10\%$.

```
OS=10;
zeta=(-log(OS/100))/(sqrt(pi^2+(log(OS/100)^2)))
zeta =
      0.59116
```

Calculando valor do ganho para $e(\infty) \leq 10\%$:

```
Kp=(1-0.1)/0.1
Kp =
     9
dcgain(BoG)
ans =
            1
K_err=Kp/dcgain(BoG)
K_err =
            9
ftmf_K_err=feedback(K_err*BoG,1);
figure; step(ftmf_K_err)
```

Muitas oscilações, overshoot muito elevado, impraticável.




```
stepinfo(ftmf_K_err)
        RiseTime: 0.3
    SettlingTime: 60.6
     SettlingMin: 0.14403
     SettlingMax: 1.6828
       Overshoot: 86.973
      Undershoot: 0
            Peak: 1.6828
        PeakTime: 0.8
```

Projeto de PI

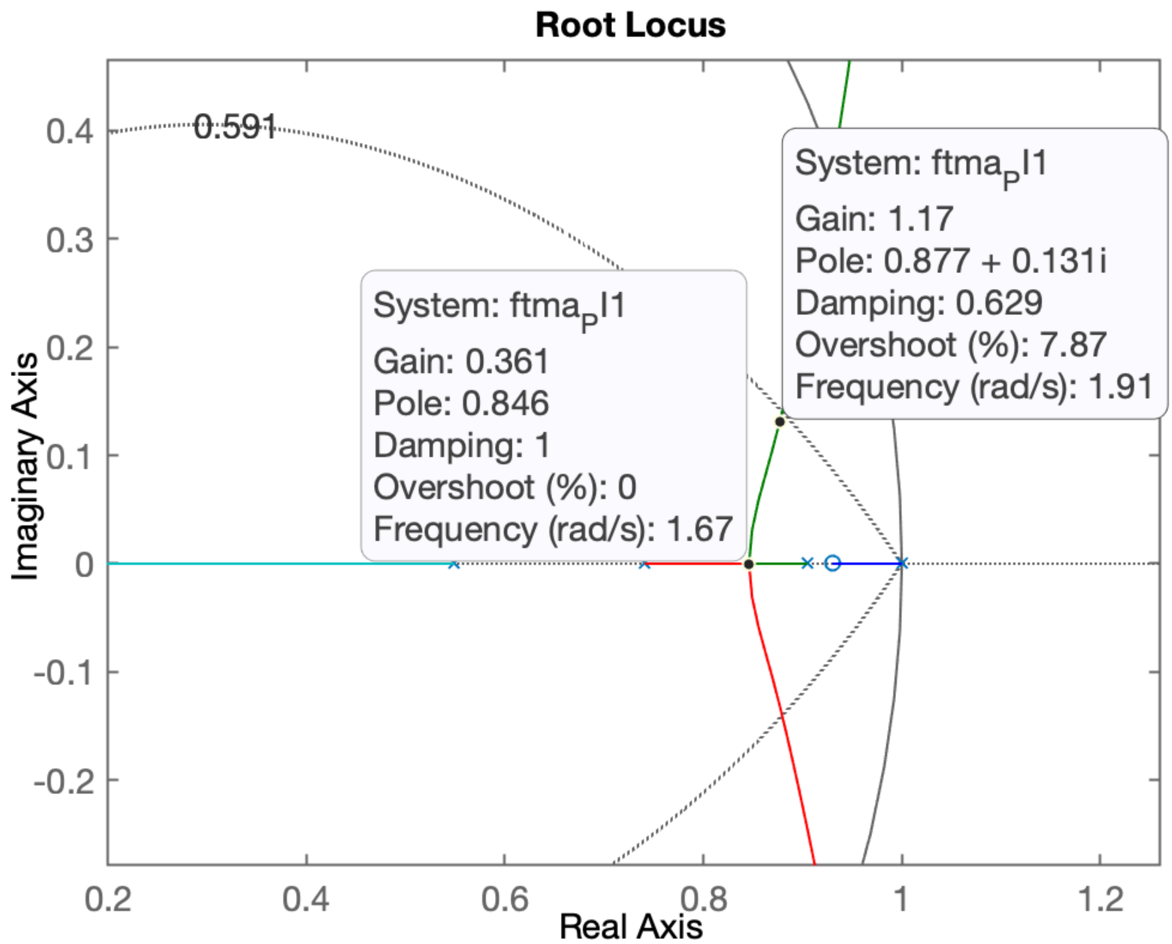
Implica colocar o pólo do mesmo em $z = 1$. Falta estimar um bom local para o zero do PI.

- **Opção inicial** colocando o zero do PI próximo do pólo mais lento da planta:

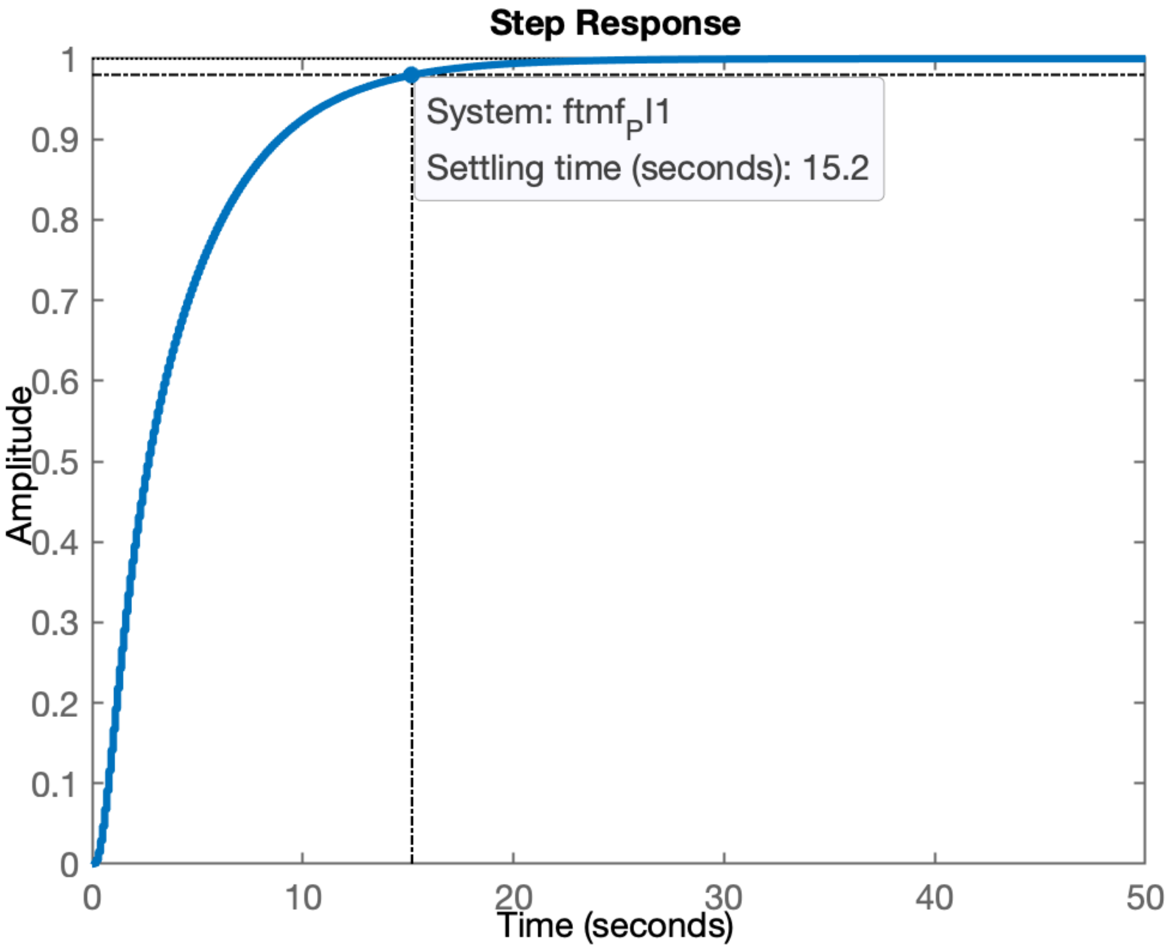
```
zpk(BoG)

   0.0023485 (z+2.924) (z+0.2074)
  --------------------------------
  (z-0.9048) (z-0.7408) (z-0.5488)
```

Zero do PI em $0,9048 < z_{PI} < 1$.

```
C_PI1=tf([1 -0.93], [1 -1],T);
zpk(C_PI1)

  (z-0.93)
  --------
   (z-1)

Sample time: 0.1 seconds
Discrete-time zero/pole/gain model.

ftma_PI1=C_PI1*BoG;
figure; rlocus(ftma_PI1)
% Zoom na região de interese revela
zgrid(zeta,0)
```




```
ftmf_PI1=feedback(0.36*ftma_PI1, 1);
step(ftmf_PI1)
pole(ftmf_PI1)
ans =
      0.97495
      0.85147
      0.84086
      0.52634
```



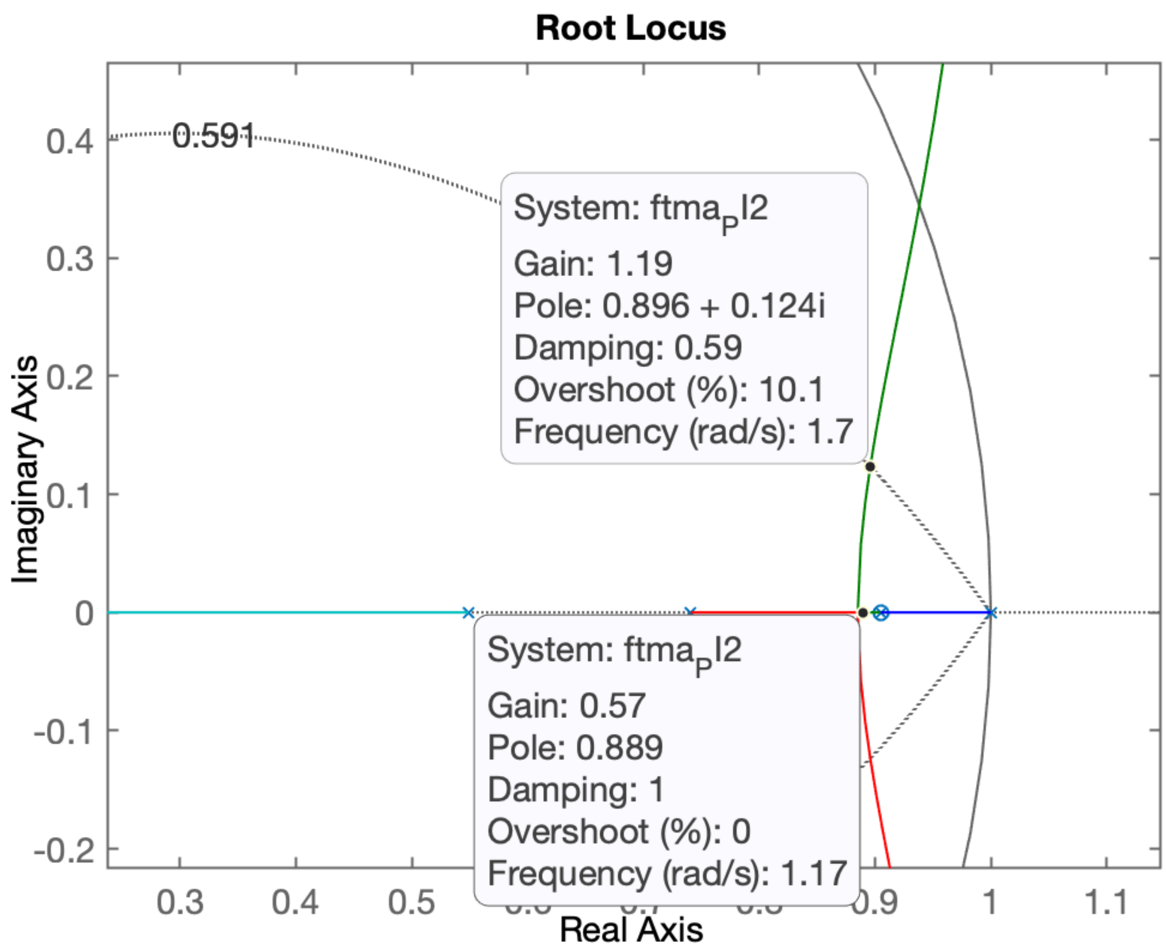
Infelizmente o tempo de assentamento é elevado demais:

```
stepinfo(ftmf_PI1)
        RiseTime: 8.1
    SettlingTime: 15.2
     SettlingMin: 0.90149
     SettlingMax: 1
       Overshoot: 0
      Undershoot: 0
            Peak: 1
        PeakTime: 53.9
```

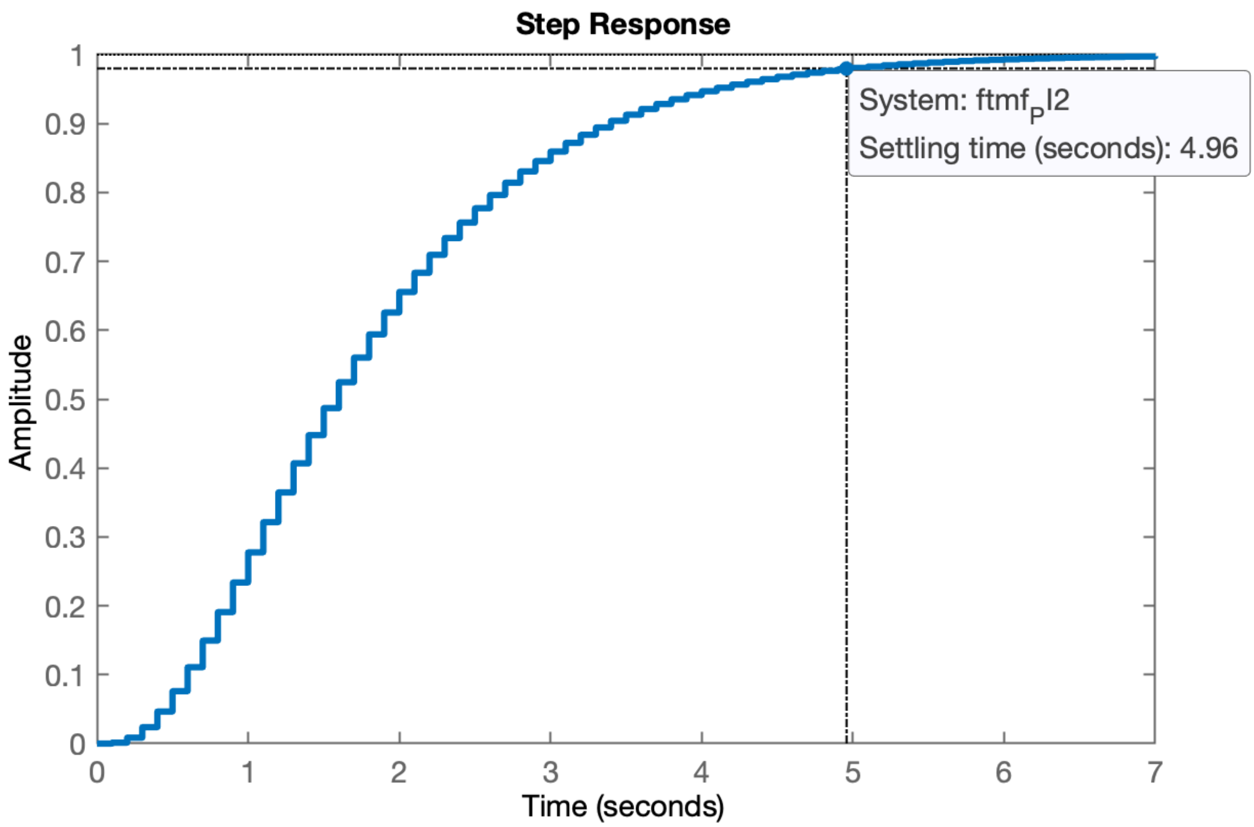
O **PI mais rápido** poderia ser obtido cancelando o pólo mais lento da planta...

```
polos
polos =
           -6
           -3
           -1
polosd=pole(BoG)
polosd =
      0.90484
      0.74082
      0.54881
C_PI2=tf(poly(polosd(1)), [1 -1],T);
```

```
zpk(C_PI2)

  (z-0.9048)
  ----------
    (z-1)

Sample time: 0.1 seconds
Discrete-time zero/pole/gain model.

ftma_PI2=C_PI2*BoG;
figure; rlocus(ftma_PI2)
zgrid(zeta,0)
% Um zoom na região de interesse revela:
```




```
ftmf_PI2=feedback(0.57*ftma_PI2, 1); % resposta super-amortecida
figure; step(ftmf_PI2)
```

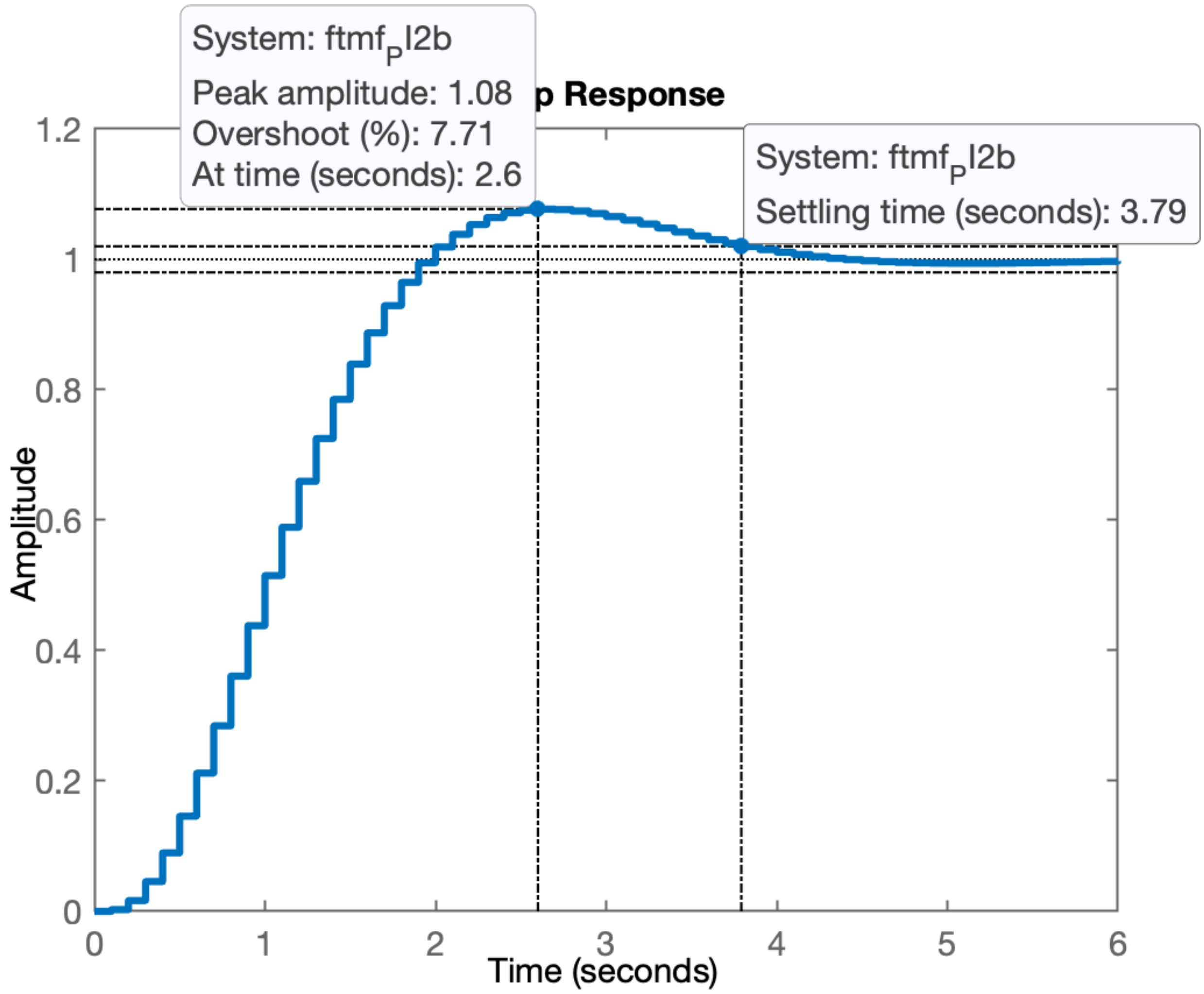


Agora o tempo de assentamente foi bastante reduzido:

```
stepinfo(ftmf_PI2)
        RiseTime: 2.8
    SettlingTime: 5
     SettlingMin: 0.90406
     SettlingMax: 0.99997
       Overshoot: 0
      Undershoot: 0
            Peak: 0.99997
        PeakTime: 10.9
```

Se o ganho ainda for incrementando permitido $\%OS \approx$ 10% obtemos:

```
ftmf_PI2b=feedback(1.1*ftma_PI2, 1); % resposta super-amortecida, %OS=10%
figure; step(ftmf_PI2b)
```



Conseguimos uma resposta abaixo dos 4 segundos:

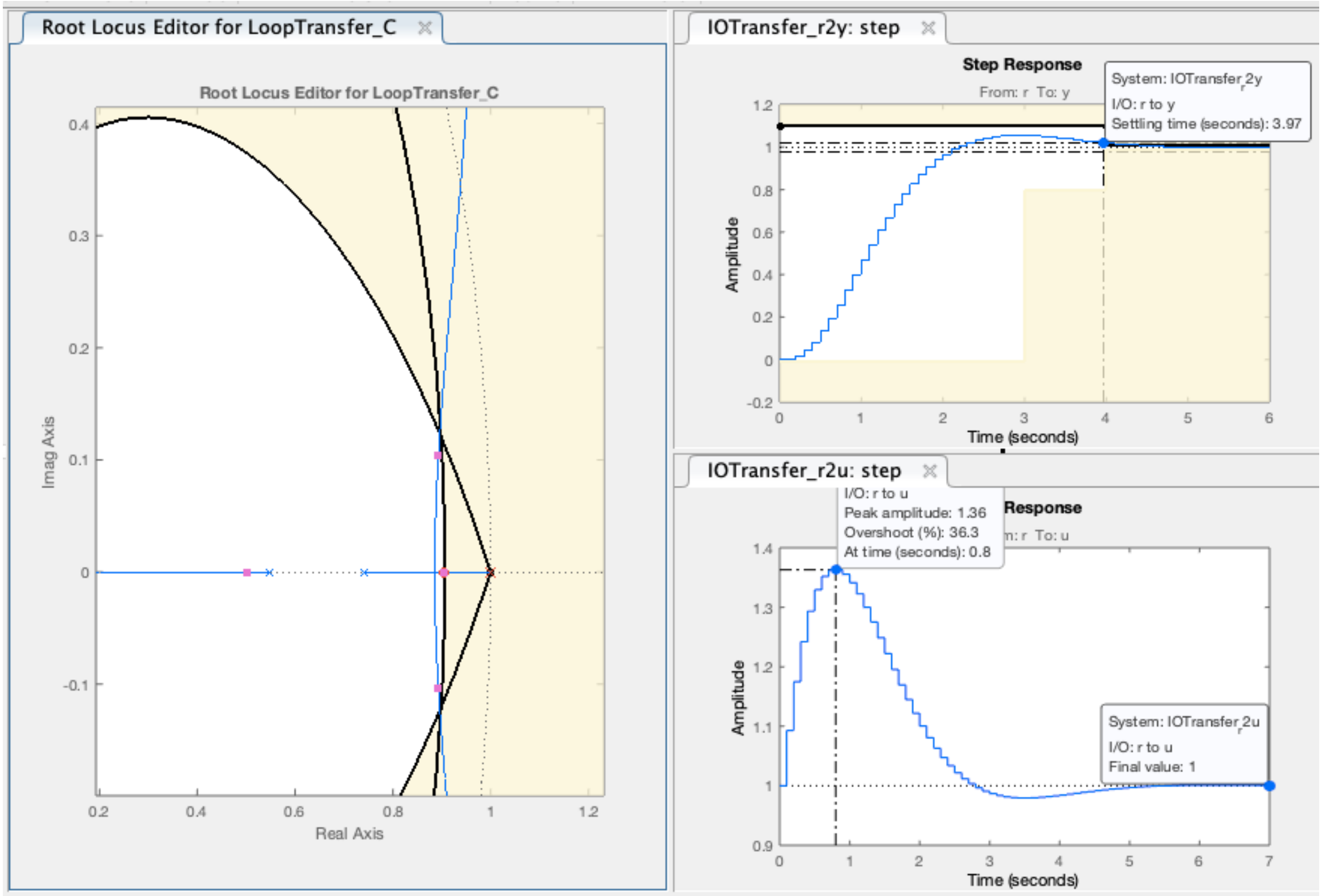
```
stepinfo(ftmf_PI2b)
        RiseTime: 1.2
    SettlingTime: 3.8
     SettlingMin: 0.92893
     SettlingMax: 1.0771
       Overshoot: 7.7076
      Undershoot: 0
            Peak: 1.0771
        PeakTime: 2.6
```

Na verdade, como o valor de $y[kT]$ ainda está abaixo de 1.1, poderiámos ter usado um ganho maior ainda...

E se fosse um **Lag** ?

Projeto de Lag

Usando App Control System Designer:

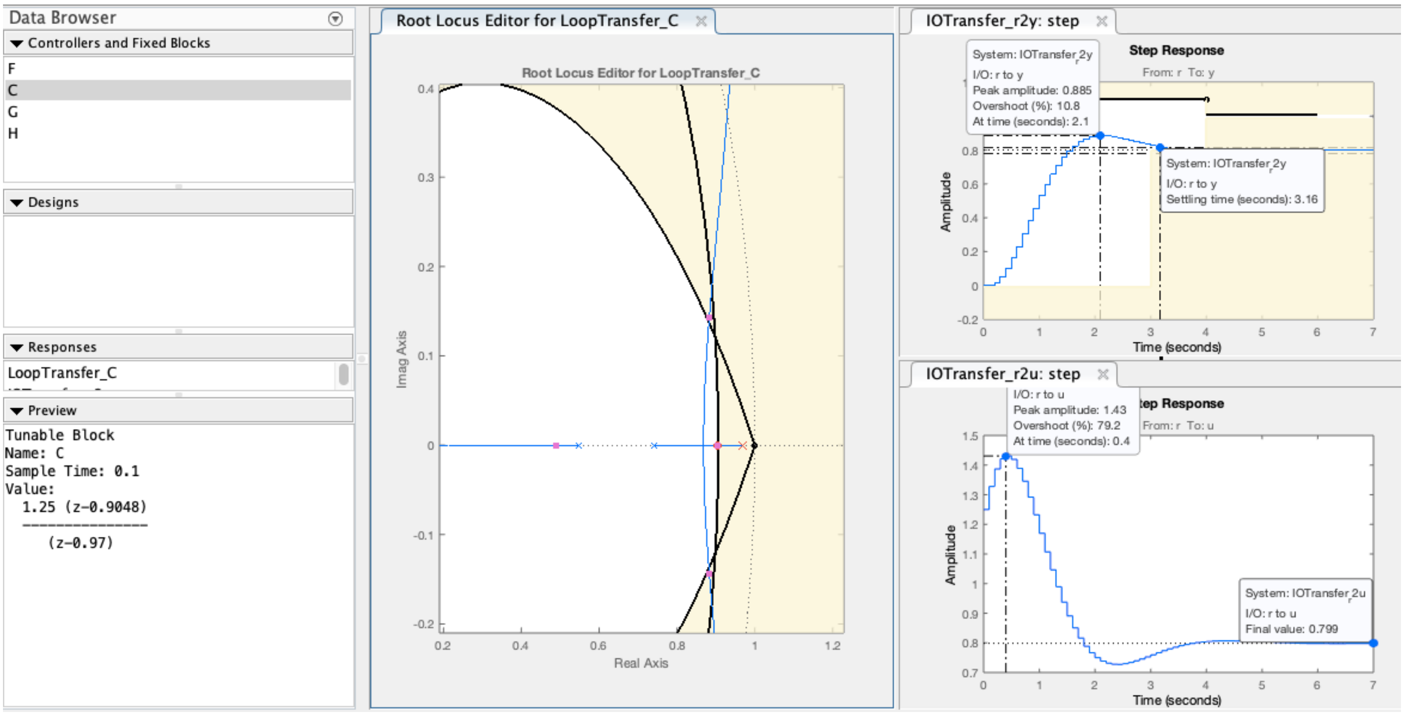


Menor tempo obtido com %OS = 10% --> 3,61 segundos (K_PI2=1,2).

Tentando "apressar" as coisas com um Lag...

```
C_PI2=tf(poly(polosd(1)), [1 -1],T);
C_Lag=tf(poly(polosd(1)), [1 -0.97],T);
```

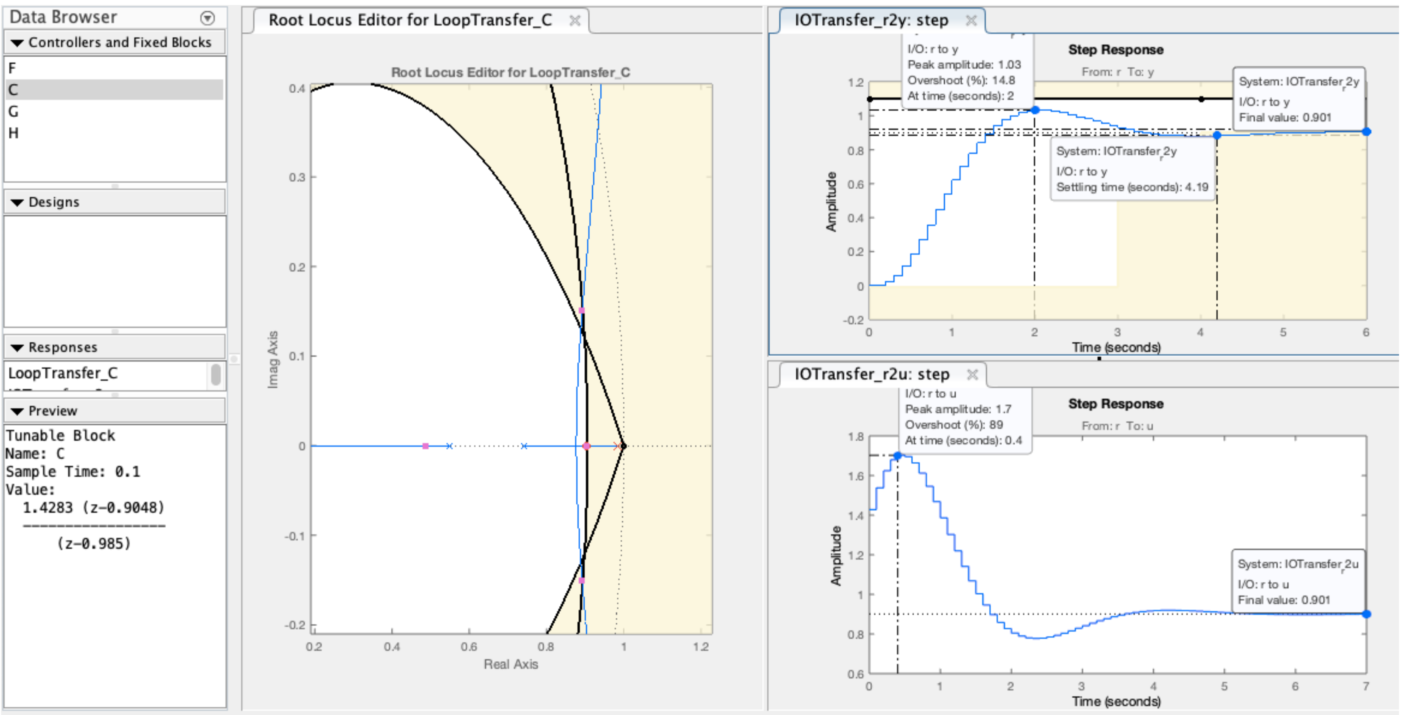
Um resultado muito bom foi obtido com:



Mas com erro maior que os 10%.

O pólo em $z = 0.97$ não parece permitir $e(\infty) \leq 10\%$.

Melhora algo colocando o pólo em $z = 0,985$, mas ainda fica algo complicado conseguir $e(\infty) \leq 10\%$, com $t_s \leq 4$ segundos. Na realidade, desta forma conseguimos $t_s = 4,19$ segundos:

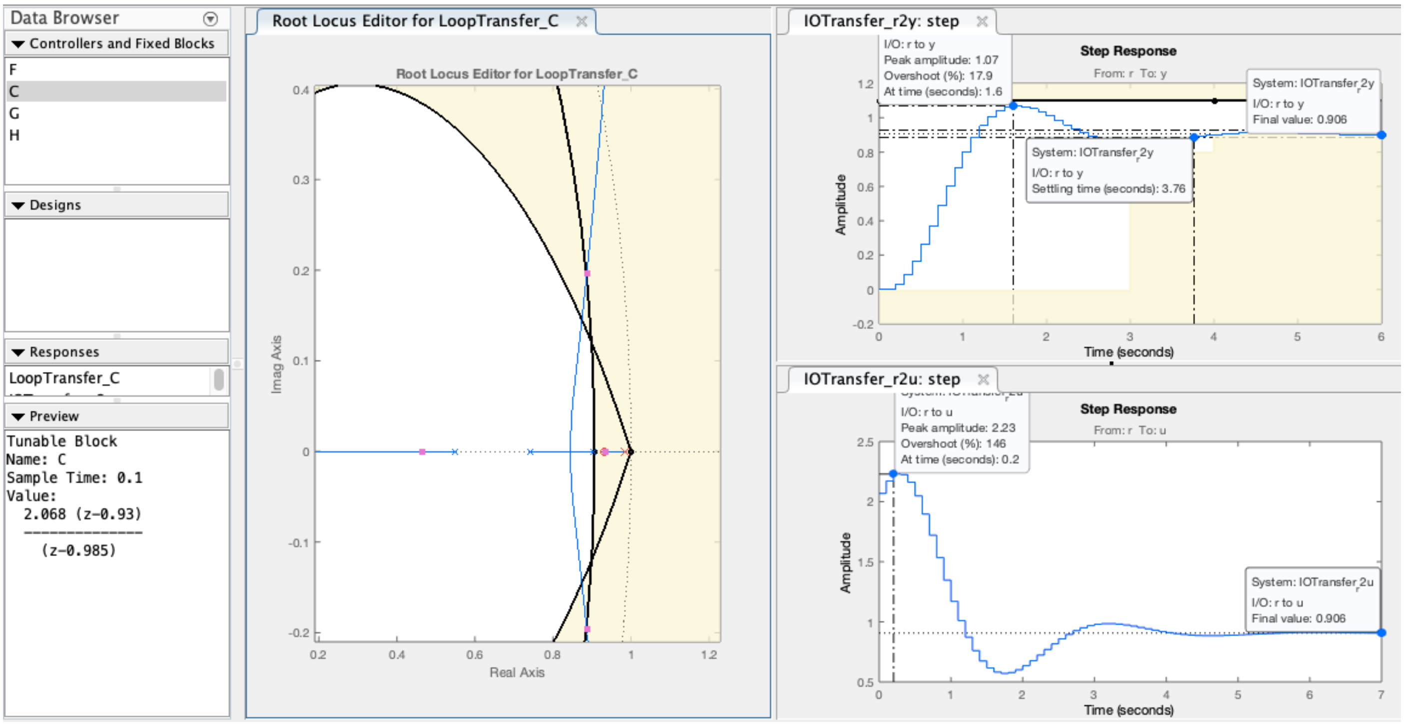


Variando o local do zero do Lag...

```
zpk(C_Lag3)

  2.068 (z-0.93)
  --------------
    (z-0.985)

Name: C
Sample time: 0.1 seconds
Discrete-time zero/pole/gain model.
```

Note que o zero do controlador não cancela mais o pólo mais lento da planta.

Esta próximo...



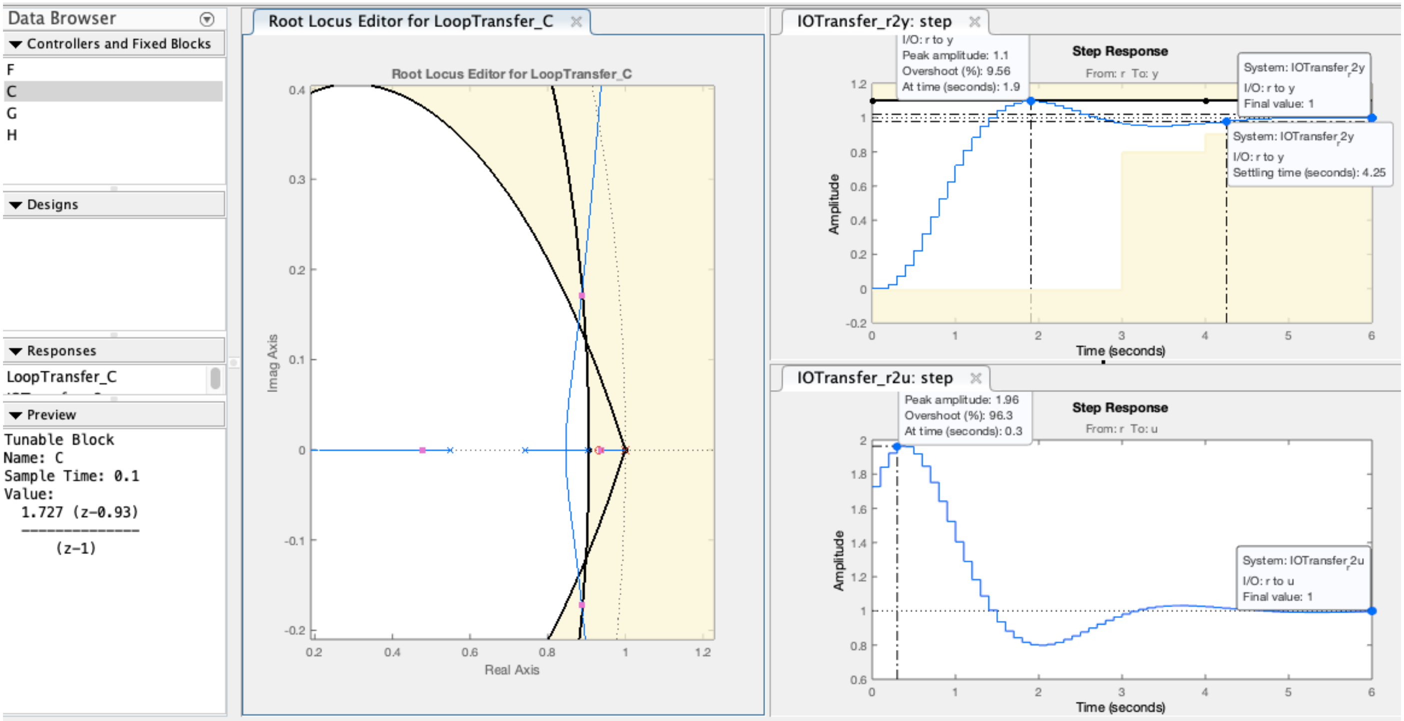
A título de curiosidade, se este Lag3 --> PI3:

```
C_PI3=tf(poly(0.93), [1 -1],T);
zpk(C_PI3)

  (z-0.93)
  --------
   (z-1)

Sample time: 0.1 seconds
Discrete-time zero/pole/gain model.
```

Poderia ser obtido um ts = 4,25 segundos. Note que o Lag teria sido mais rápido, ts = 3,76.

Mas, tempos menores que 4 seugndos só com App Control System Designer:

```
zpk(C_PI3)

  1.727 (z-0.93)
  --------------
      (z-1)

Name: C
Sample time: 0.1 seconds
Discrete-time zero/pole/gain model.
```



Fim.

```
save prova1
diary off
```

Fernando Passold, em 18/05/2025