MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO AMAZONAS
CAMPUS MANAUS - DISTRITO INDUSTRIAL

INSTITUTO FEDERAL
AMAZONAS
Campus Manaus-Distrito Industrial

Atividade Avaliativa 2 – Controle Discreto
12/08/2024
Cancelamento e Alocação de Polos, LGR e Controle DeadBeat

Aluno: Gabriel Almeida Santos de Oliveira.
Nº de matrícula: 2021000042.

1. **(O Sistema)** Considere um sistema para controle de temperatura, cujo modelo é dado por:

$$T_L(s) = \frac{0.95}{(16s+1)}[Q_{in}(s) + T_{Amb}(s)] \tag{1}$$

$$Q_{in}(s) = \frac{13.4}{(0.5s+1)}[U_F(s)] \tag{2}$$

Queremos manter a temperatura do fluido $T_L(t)$ (variável a ser controlada) rente a um valor desejado. Para tanto, aplicamos um sinal elétrico $U_F(t)$ (variável manipulada), que por meio de um banco de resistores altera o fluxo de calor fornecido ao sistema $Q_{in}(t)$. Temos como distúrbio variações na temperatura ambiente $T_{Amb}(t)$.

Todos os controladores devem ser projetados diretamente no plano 'z'. Discretizem a planta, e escolham um tempo de amostragem adequado.

2. **(Cancelamento e Alocação de Polos)**

a) Ajuste um controlador PI discreto por cancelamento de polos tal que a resposta do sistema para referência, em malha-fechada (MF), seja pelo menos 3 vezes mais rápida que a resposta em malha-aberta (MA) e sem nenhum pico. (Dica: cancele o polo mais rápido do processo).

Primeiramente foi calculado a função transferência da planta:

$$T_L(s) = \frac{0,95}{(16s+1)}[Q_{in}(s) + T_{amb}(s)]$$

$$Q_{in}(s) = \frac{13,4}{(0,5s+1)} U_F(s)$$

* Para projeto do controlador o distúrbio ($T_{amb}(s)$) será ignorado.

$$\therefore\ T_L(s) = \frac{0,95}{(16s+1)}\left[\frac{13,4}{(0,5s+1)} U_F(s)\right] \Rightarrow \frac{T_L(s)}{U_F(s)} = \frac{0,95 \cdot 13,4}{(16s+1)(0,5s+1)}$$

$$\frac{T_L(s)}{U_F(s)} = \frac{12,73}{(16 \cdot 0,5)(s+2)(s+0,0626)} = \frac{1,59}{(s+2)(s+0,0625)} = \frac{1,59}{s^2+2,0625+0,125}$$

* Polos no domínio contínuo:

$P_{s1} = -2$

$P_{s2} = -0,0625$

A partir da planta, para projeto do controlador foi desenvolvido o seguinte Código no matlab:

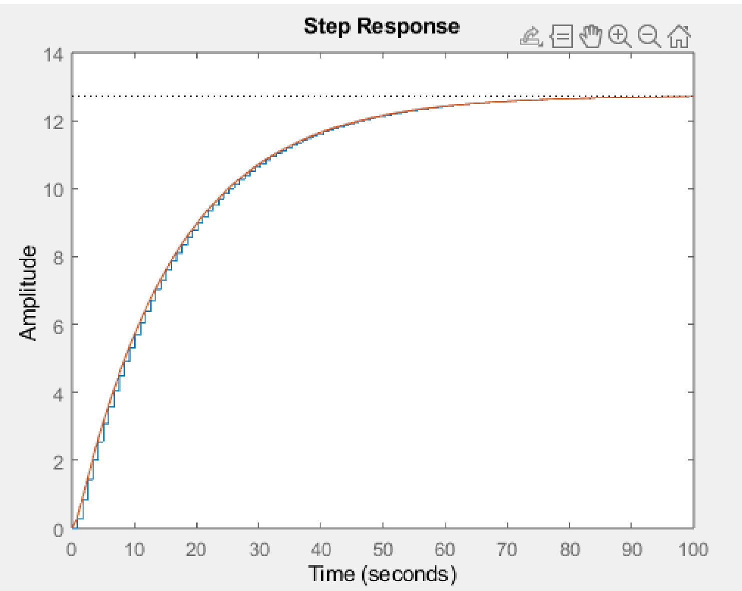
```
clc; clear all;

%% FT Planta
s=tf('s');
Gma = (0.95*13.4) / ((0.5*s + 1)*(16*s + 1))

poles = roots(Gma.denominator{1})

step_result = stepinfo(Gma);
tsd = step_result.SettlingTime/3
% pela resposta ao step é observado que a FTMA
%tem um tempo de assentamento
% de aproximadamente 61s.

%% Calculando Ts
Ts = tsd/25 %0.76

%% discretização com casamento de polos e zeros:
Gma_z = zpk(c2d(Gma, Ts, 'matched'))
% step(Gma_z, Gma)


%% verificação polos discretos
disp('# FTMA_z Poles: ')
polos = pole(Gma_z)
```

- A Função de Transferência continua:

```
Gma =

     12.73
  ------------------
  8 s^2 + 16.5 s + 1

Continuous-time transfer function.
```

- O tempo de assentamento da malha aberta foi obtida pela resposta ao degrau, visto que se deseja 1/3 do tempo de assentamento da resposta ao degrau da MA, se calculou o tempo de assentamento desejado e o tempo de amostragem foi aproximado por 1/25 avos deste, o que resulta em aproximadamente 0.84 segundos.

- Abaixo a resposta ao degrau da MA e da discretização.



- A Função de Transferência discretizada por casamento de polos e zeros, a seguir seus polos discretos:

```
Gma_z =

    0.26547 (z+1)
  ----------------------
  (z-0.9488) (z-0.1859)

Sample time: 0.84139 seconds
Discrete-time zero/pole/gain model.

# FTMA_z Poles:

ans =

   0.9488
   0.1859
```

Se cálcula então qual o polo desejado para as especificações de desempenho pedidas,

* Calculando a E.C. desejada:

$\zeta_d = 1$; $t_{rd} = \dfrac{4}{\zeta_d . \omega_{nd}} \Rightarrow 20 = \dfrac{4}{1 . \omega_{nd}} \Rightarrow \omega_{nd} = \dfrac{4}{20} = 0{,}2$

↳ $M_p = 0\%$

$\therefore E.C.d = s^2 + 2.\zeta_d.\omega_{nd} s + \omega_n^2 = s^2 + 0{,}4 s + 0{,}04$

* polos:

$s_{pD} = \dfrac{-0{,}4 \pm \sqrt{(0{,}4)^2 - 4.1.0{,}04}}{2.1} = \dfrac{-0{,}4 \pm 0}{2} = \begin{cases} s_{pD1} = -0{,}2 \\ s_{pD2} = -0{,}2 \end{cases}$

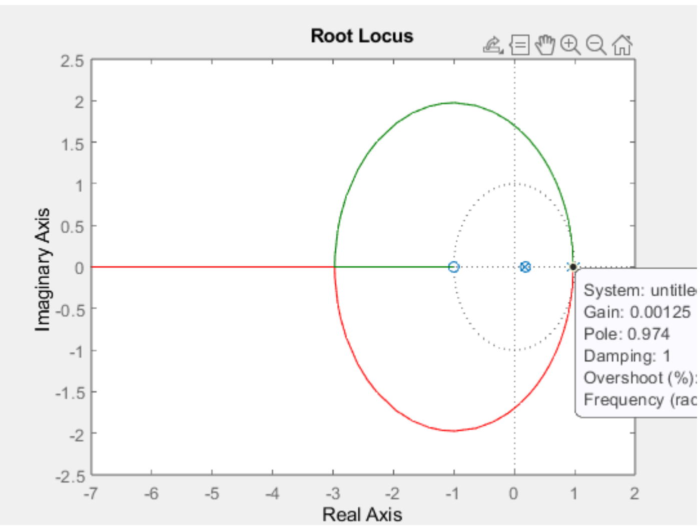
$\therefore E.C.d = (s+0{,}2)(s+0{,}2)$

Utilizando o matlab se obtem os polos dominantes discretos desejados:

```
%% Cálculo dos polos dominantes desejados
Pds = [-0.2 -0.2];
Pdz1 = exp(Ts*Pds(1))
Pdz2 = exp(Ts*Pds(2))
```

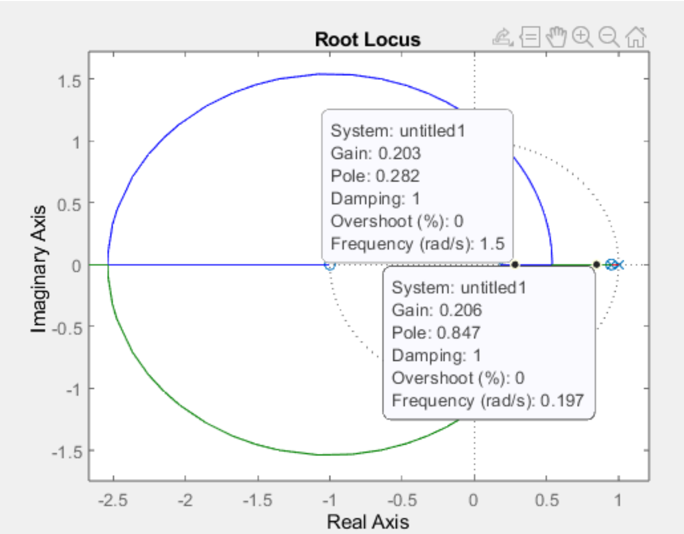
```
Pdz1 =

    0.8451


Pdz2 =

    0.8451
```

O controlador PI tem o formato: C(Z) = Kc*(Z – z1) / (Z - 1). Onde, neste caso ‘z1’ é o polo o qual se deseja cancelar. Intuitivamente, uma vez que se deseja acelerar a resposta da planta em MF, tende-se a crer que o polo mais lento deve ser cancelado. Porém, observando o LR abaixo da planta vezes o controlador, ao cancelar o polo mais lento, se nota que seria necessário um ganho muito pequeno (da ordem de 1e-3) para que a planta cumprisse o requisito de nenhum sobressinal, isto é, os polos dominantes não possuirem parte complexa. E, sabendo a posição dos polos desejados, se percebe que caso seja cancelado o polo mais lento (P_z2 = 0.1859) o LR não passa pelo polo dominantes desejado. Portanto, é cancelado o polo mais rapido (P_z1 = 0.9488).




```
%% Calculo controlador
z = tf('z', Ts);
[faster_pole, ind] = max(polos)
pole2cancel = faster_pole;
%pole2cancel = 0.1859;
C_z = (z - pole2cancel) / (z - 1);
rlocus(Gma_z*C_z)
```

Observando o lugar das raizes da planta vezes o controlador cancelando o polo mais raipdo, se encontra o ganho que leva o polo dominante até o polo discreto desejado:



- Que resulta no controlador:

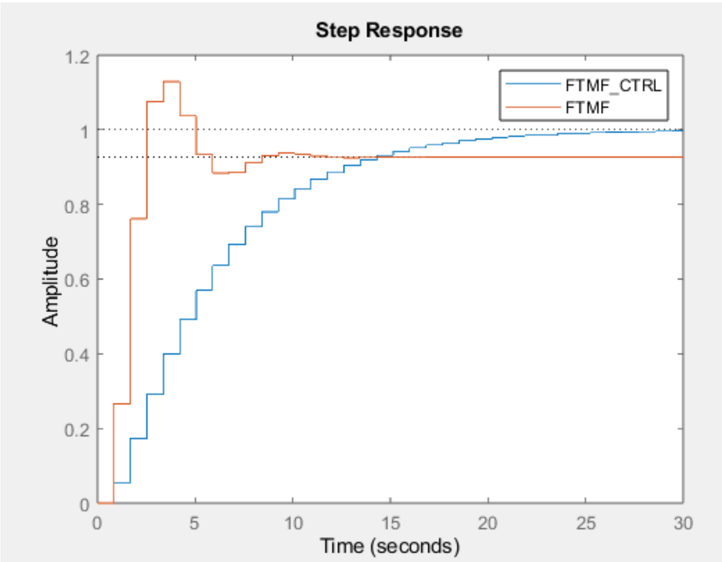
```
Kc = 0.208;
C_z = zpk(Kc*C_z)
```

```
C_z =

 0.208 (z-0.9488)
 ----------------
      (z-1)
```

Então, através do código abaixo se obteve o sistema de malha fechada assim como sua resposta ao degrau:

```
%% Calculado a FTMF resultante
GC_z = minreal(Gma_z*C_z);
GC_mf_z = feedback(GC_z, 1)
G_mf_z = feedback(Gma_z, 1);
step(GC_mf_z, G_mf_z)
legend("FTMF_CTRL", "FTMF")
controled_step_result = stepinfo(GC_mf_z)
controled_step_result.SettlingTime
```

```
GC_mf_z =

    0.055217 (z+1)
  ----------------------
  (z-0.2851) (z-0.8455)

Sample time: 0.84139 seconds
Discrete-time zero/pole/gain model.
```




```
controled_step_result =

  struct with fields:

        RiseTime: 10.9381
    SettlingTime: 21.0348
     SettlingMin: 0.9050
     SettlingMax: 1.0000
       Overshoot: 0
      Undershoot: 0
            Peak: 1.0000
        PeakTime: 66.4698
```

Observa-se que o tempo de assentamento do sistema com o controlador é aproximadamente 1/3 do tempo de assentamento da malha aberta assim como o overshoot é nulo. Logo, se atingiu as especificações de desempenho desejadas. Toda via, se observa que o ganho de malha fechada é bem pequeno (ordem de 1e-2), característico do metodo de cancelamento de polos, e pode vir a ser um problema na aplicação prática do controlador.

....

b) Ajuste um controlador PI ou PID discreto por alocação de polos tal que a MF do sistema para referência apresente dois polos reais e iguais e que a dinâmica em MF seja 3 vezes mais rápida que a resposta em malha-aberta (MA) e sem nenhum pico.

Dado que a planta e as especificações de desempenho são as mesmas da questão anterior, a porção incial do projeto doc ontrolador não teve nenhuma alteração, como se observa no código utilizado abaixo:

```
clc; clear all;

%% FT Planta
s=tf('s');
Gma = (0.95*13.4) / ((0.5*s + 1)*(16*s + 1))

step_result = stepinfo(Gma);
tsd = step_result.SettlingTime/3


%% Calculando Ts
Ts = tsd/25 %0.76s


%% discretização com casamento de polos e zeros:
Gma_z = zpk(c2d(Gma, Ts, 'matched'))
```

- De onde se obtem as funções de transferência continua e discreta:

```
Gma =

      12.73
  ------------------
  8 s^2 + 16.5 s + 1
```

```
Gma_z =

    0.26547 (z+1)
  ---------------------
  (z-0.9488) (z-0.1859)
```

A partir das especificações de desempenho pedidas, ts da MF 1/3 do ts de Malha aberta e zero sobressinal, se calcula os polos dominantes desejados:

```

%% Cálculo do wn
zeta = 1; % nenhum sobressinal
% visto que se deseja 1/3 do tempo de assentamento de MA (~61s),
% se arredonda o ts desejada pra 20 para evitar parte complexa no
% polo desejado
wn = 4 / (20*zeta);

%% calculo da equação característica continua desejada
EC_d = s^2 + 2*wn*zeta*s + wn^2
```

De onde se obtem a equação característica desejada:

Discretizando os polos desejados continuos:

```
%% Cálculo dos polos dominantes discretos desejados na MF
Pds = pole(1/EC_d)
Pdz1 = exp(Ts*Pds(1))
Pdz2 = exp(Ts*Pds(2))
z = tf('z', Ts);
EC_dz = (z - Pdz1) * (z - Pdz2)
```

- De onde se obtem os polos discretos desejados:

```
EC_d =

  s^2 + 0.4 s + 0.04
```

```
Pdz1 =

  0.8451 + 0.0000i

Pdz2 =

  0.8451 - 0.0000i
```

- E a equação característica desejada discreta:

```
EC_dz =

  z^2 - 1.69 z + 0.7142
```

Se optou por utilizar um controlador PID, de posse da equação característica desejada se desenvolve o seguinte sistema de equações abaixo para se chegar ao valor dos zeros e do ganho do controlador.

\# Planta:

$$G(z) = \frac{0{,}2654(z+1)}{(z-0{,}9488)(z-0{,}7859)}$$

$$C_{PID}(z) = \frac{K_c(z-z_1)(z-z_2)}{z(z-1)}$$

$$E.C.d_z : (z-0{,}8451)(z-0{,}8451) = z^2 - 1{,}69z^1 + 0{,}7142z^0$$

$$FTMF: \frac{N_G \cdot N_c}{P_G P_c + N_G N_c}$$

$$\therefore GC_{MF}(z) = \frac{0{,}2654(z+1)K_c(z-z_1)(z-z_2)}{(z-0{,}9488)(z-0{,}7859)z(z-1) + 0{,}2654K_c(z+1)(z-z_1)(z-z_2)}$$

• Antes que seja montado o sistema de equações, é possível observar que a ordem da E.C. da FTMF é maior do que a ordem da E.C. desejada, o que irá resultar em um sistema sem solução. Então, considerando que tanto o polo 'z' do controlador como o polo (z-0,7859) são muito lentos e pouco influenciam na resposta transitória do sistema, estes serão ignorados para o calculo das constantes do controlador.

$$\therefore E.C.FTMF: (z-0{,}9488)(z-1) + 0{,}2654K_c(z+1)(z-z_1)(z-z_2)$$

$$z^2 - z - 0{,}9488z + 0{,}9488 + (0{,}2654K_c z + 0{,}2654K_c)(z^2 - z_2 z - z_1 z + z_1 z_2)$$

* será definido $A = 0{,}9488$; $B = 0{,}2654$

$$z^2 - z - Az + A + BK_c z^3 - BK_c z_2 z^2 - BK_c z_1 z^2 + BK_c z_1 z_2 z + BK_c z^2 - BK_c z_2 z - BK_c z_1 z + BK_c z_1 z_2$$

INSTITUTO FEDERAL
AMAZONAS
Campus Manaus-Distrito Industrial

* Os sistemas de equações ainda são incompatíveis, será, portanto, acrescentado um polo no infinito (em 0 no domínio discreto) à E.C. desejada para não alterar significamente seu comportamento e viabilizar a resolução do sistema de equações.

$\therefore EC_d = z(z-0{,}8451)(z-0{,}8451) = (z^2-0{,}8451z)(z-0{,}8451) = z^3 - 0{,}8451z - 0{,}8451z^2 + 0{,}7142z + 0z^0$

• Igualando as equações características:

$z^3: BKc = 1$

$z^2: 1 - BKc\zeta_2 - BKc\zeta_1 + BKc = -0{,}8451$

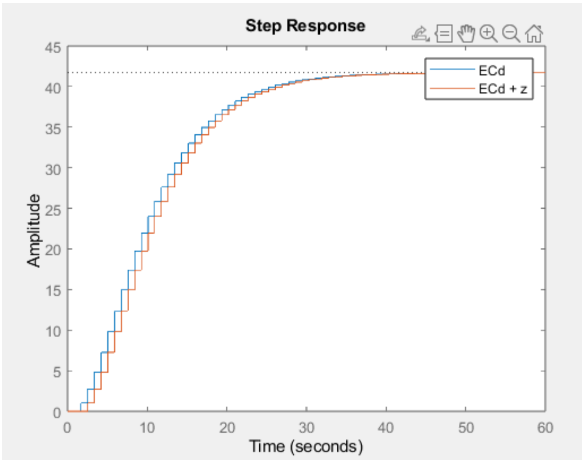
$z^1: -1 - A - BKc\zeta_1\zeta_2 - BKc\zeta_2 - BKc\zeta_1 = -0{,}8451 + 0{,}7142$

$z^0: A + BKc\zeta_1\zeta_2 = 0$

No gráfico ao lado se observa que a introdução de um Polo no zero da equação característica pouco altera a dinâmica da mesma.

Código utilizado para obtenção do gráfico:

```
EC_dz2 = z*(z - Pdz1) * (z - Pdz2)
step(1/EC_dz, 1/EC_dz2)
legend("ECd", "ECd + z")
```



Utilzando o matlab para resolução do sistema de equação se descobre que o mesmo não possui solução mesmo com as adaptações feitas:

```
%% calculo constantes do controlador
A = 0.9488; B = 0.2654;
syms Kc z1 z2;
eq4 = B*Kc == 1;
eq3 = 1 - B*Kc*z2 - B*Kc*z1 + B*Kc == -0.8451;
eq2 = -1 - A - B*Kc*z1*z2 - B*Kc*z2 - B*Kc*z1 == -0.8451 + 0.7142;
eq1 = A + B*Kc*z1*z2 == 0;
sol = (vpasolve([eq1;eq2;eq3;eq4]));
Kc = double(sol.Kc)
zc_1 = double(sol.z1)
zc_2 = double(sol.z2)
```

```
Kc =

  0×1 empty double column vector

zc_1 =

  0×1 empty double column vector

zc_2 =

  0×1 empty double column vector
```

Se recorreu então a analisar o lugar das raizes e inferir/testar diferentes posições para os zeros do controlador, até que se chegou ao seguinte controlador que melhor se aproxima as especificações de desempenho pedidas:

```
%% Teste Controladores

%z1 = 0.28 | z2 = 0.9
% kc = 0.4 -> ts = 28.6s ; Ovsh = 9.4
% kc = 0.5 -> ts = 25.2s ; Ovsh = 9.1
% kc = 0.6 -> ts = 22.7s ; Ovsh = 8.7
% kc = 0.7 -> ts = 20.2s ; Ovsh = 8.5
% kc = 0.9 -> ts = 17.6s ; Ovsh = 8.1
% kc = 1 -> ts = 15.9s ; Ovsh = 8.2

%z1 = 0.1859 | z2 = 0.92
% Kc = 0.1 -> ts = 63.1 ; Ovsh = 4
% kc = 0.36 -> ts = 29.9s ; Ovsh = 5.9
% Kc = 0.8 -> ts = 15.9 ; Ovsh = 6.8
Kc = 0.8
C_z = Kc*( (z - 0.1859)*(z - 0.92) ) / ( z*(z - 1) )

GC_z = Gma_z*C_z;
 %rlocus(GC_z)


%% Calculado a FTMF resultante
GC_mf_z = feedback(GC_z, 1)
G_mf_z = feedback(Gma_z, 1);
step(GC_mf_z, G_mf_z)
legend("FTMF_CTRL", "FTMF")
controled_step_result = stepinfo(GC_mf_z)
controled_step_result.SettlingTime
```

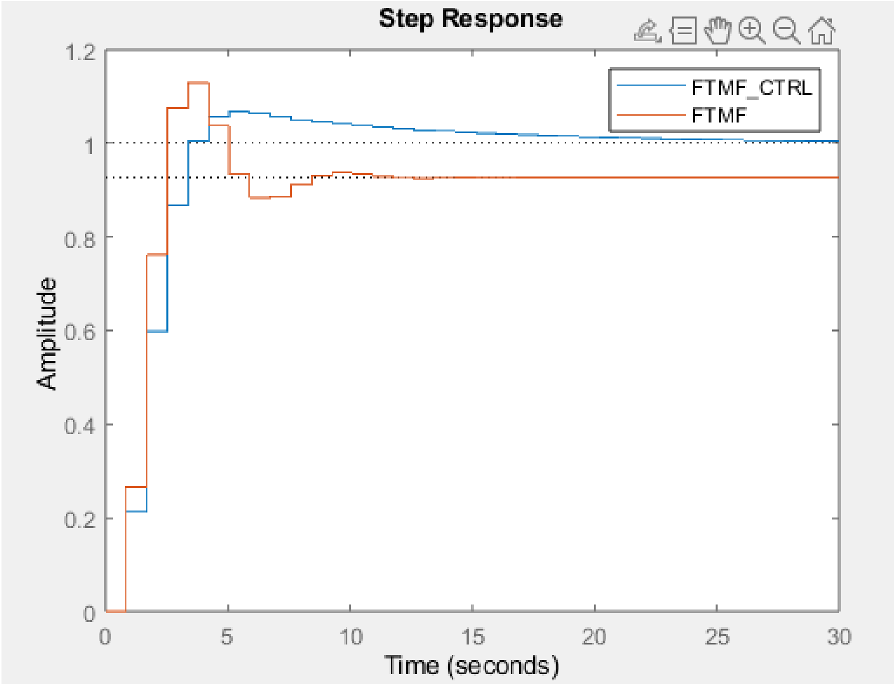
- Controlador:

```
C_z =

  0.8 (z-0.92) (z-0.1859)
  ------------------------
        z (z-1)
```

- Função de transferência de Malha fechada

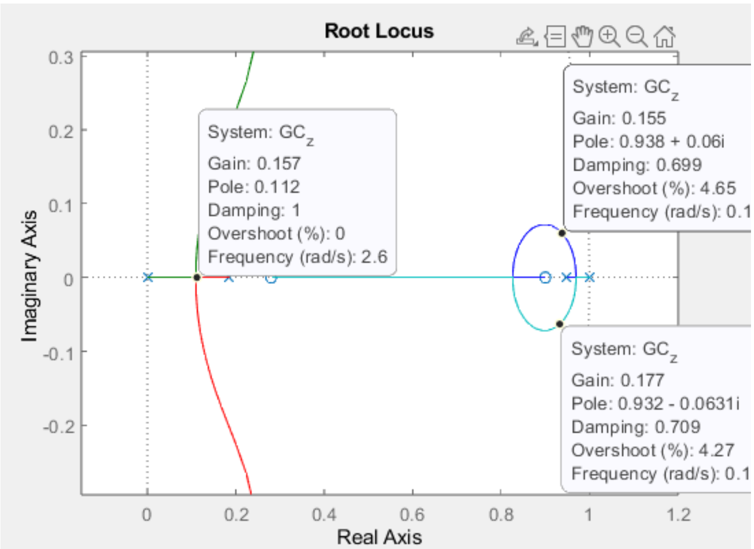
```
GC_mf_z =

    0.21237 (z-0.92) (z+1) (z-0.1859)
  ---------------------------------------------
  (z-0.913) (z-0.186) (z^2 - 0.8233z + 0.2139)
```

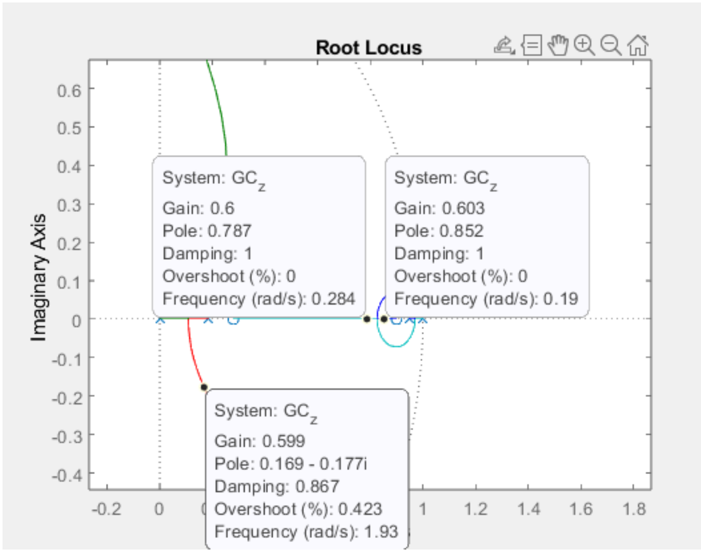
Resposta ao degrau e especificações de desempenho obtidas:




```
controled_step_result =

  struct with fields:

        RiseTime: 2.5242
    SettlingTime: 15.9864
     SettlingMin: 1.0001
     SettlingMax: 1.0684
       Overshoot: 6.8382
      Undershoot: 0
            Peak: 1.0684
        PeakTime: 5.0483
```

Observando o lugar das raizes de diferentes configurações:





Se observa que não será possível remover o overshoot apenas alterando a posição dos zeros pois ganhos que levam os polos não dominantes a permanecrem na reta real, os polos dominantes se distanciam do polo desejado. Outro fator que contribui para a presença do overshoot é o zero dominantes, que acelera a resposta do sistema.

Uma forma de tentar eliminar o sobressinal é aplicando um filtro de referência que atenue a ação do zero dominante, pois este atua acelerando a resposta do sistema, atenuando seu efeito se espera que a resposta transitória do sistema atrase um pouco, mas que, em contra partida seja eliminado o sobressinal. O filtro de referência foi projeto a partir dos cálculos abaixo:

#Filtro de referência
*zero que se deseja atenuar: 0,92

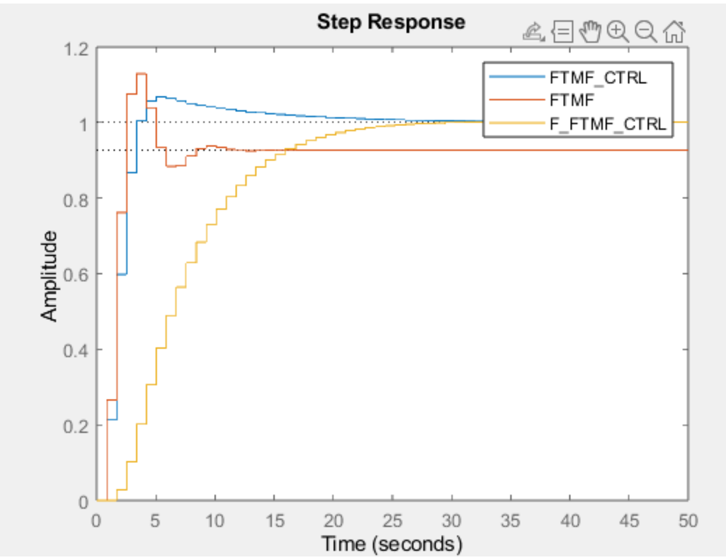
$\therefore F_R(z) = \frac{a}{z-0,92}$; $\lim_{z\to 1} F(z) = 1 \Rightarrow \lim_{z\to 1} \frac{a}{z-0,92} = 1 \Rightarrow a = 0,08$

$\therefore F_R(z) = \frac{0,08}{z-0,92}$ //

Teste da resposta do sistema com a implementação do filtro:

```
%% Calculado a FTMF resultante
GC_mf_z = feedback(GC_z, 1)
G_mf_z = feedback(Gma_z, 1);

F_z = 0.08 / (z - 0.92);
GCF_mf_z = F_z*feedback(GC_z, 1)

step(GC_mf_z, G_mf_z, GCF_mf_z)
legend("FTMF_CTRL", "FTMF", "F_FTMF_CTRL")
controled_step_result = stepinfo(GC_mf_z)
filtered_controled_step_result = stepinfo(GCF_mf_z)
```

```
filtered_controled_step_result =

  struct with fields:

        RiseTime: 20.1934
    SettlingTime: 38.7039
     SettlingMin: 0.9012
     SettlingMax: 1.0000
       Overshoot: 0
      Undershoot: 0
            Peak: 1.0000
        PeakTime: 100.1254
```

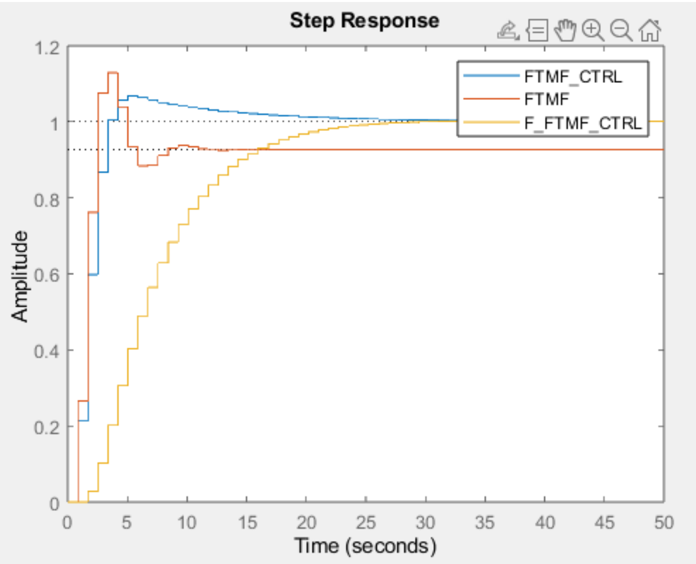


- O filtro projetado acima do zero acabou atrasando demais o sistema, então empiricamente se testou outros filtros até se encontrar um que melhor atendesse o desempenho desejado.

O melhor filtro encontrado foi:

```
%% Calculado a FTMF resultante
GC_mf_z = feedback(GC_z, 1)
G_mf_z = feedback(Gma_z, 1);

pos_filt = 0.87;
F_z = (1-pos_filt) / (z - pos_filt)
GCF_mf_z = F_z*feedback(GC_z, 1)

step(GC_mf_z, G_mf_z, GCF_mf_z)
legend("FTMF_CTRL", "FTMF", "F_FTMF_CTRL")
controled_step_result = stepinfo(GC_mf_z)
filtered_controled_step_result = stepinfo(GCF_mf_z)
```

```
F_z =

  0.13
 --------
 z - 0.87

Sample time: 0.84139 seconds
Discrete-time transfer function.
```




```
filtered_controled_step_result =

  struct with fields:

        RiseTime: 11.7795
    SettlingTime: 21.8761
     SettlingMin: 0.9010
     SettlingMax: 1.0021
       Overshoot: 0.2126
      Undershoot: 0
            Peak: 1.0021
        PeakTime: 37.0212
```

Apesar de não ter anulado o overshoot, chegou bem próximo, em menos em 1%, e no tempo de assentamento objetivado.

c) Utilizando o diagrama polo-zero ($Y(z)/R(z)$ e $Y(z)/Q(z)$) e as respostas temporais, compare os dois controladores. O sistema segue referências? Por que? O que houve com o distúrbio? A resposta ao distúrbio tem a mesma dinâmica que a resposta a mudanças de referência? Alguma resposta apresenta pico? É possível, com estes ajustes, a resposta à referência não ter pico e a velocidade da rejeição de distúrbio ser 3 vezes mais rápida que a resposta em MA?

As funções de transferência de malha fechada em relação a referência foram obtidas através dos códigos demonstrados previamente, para obtenção das FTs em relação ao disturbio, se fez os calculos apresentados abaixo:

$\boxed{15}$

#Planta:

$$G(z) = \frac{0{,}26547(z+1)}{(z-0{,}1859)(z-0{,}9488)}$$

#Controlador PI

$$C_1(z) = \frac{0{,}208(z-0{,}9488)}{(z-1)}$$

#Controlador PID

$$C_2(z) = \frac{0{,}8(z-0{,}92)(z-0{,}1859)}{z(z-1)}$$

#FTMF $y(z)/R(z)$ $\left[\frac{y(z)}{R(z)} = \frac{C(z)G(z)}{1+C(z)G(z)}\right] \Rightarrow \left[\frac{y(z)}{R(z)} = \frac{N_c . N_G}{D_c D_G + N_c N_G}\right]$

* Controlador 1 [PI]

$$\frac{y_1(z)}{R(z)} = \frac{0{,}0552(z+1)}{(z-0{,}2851)(z-0{,}8455)}$$

* Controlador 2 [PID]

$$\frac{y_2(z)}{R(z)} = \frac{0{,}2123(z-0{,}92)(z+1)(z-0{,}1859)}{(z-0{,}913)(z-0{,}186)(z^2-0{,}8233z+0{,}2139)}$$

#FTMF $y(z)/Q(z)$ $\left[\frac{y(z)}{Q(z)} = \frac{G(z)}{1+C(z)G(z)}\right] \Rightarrow \left[\frac{y(z)}{R(z)} = \frac{N_G D_c}{D_G D_c + N_c N_G}\right]$

* Controlador 1 [PI]

$$\frac{y_1(z)}{Q(z)} = \frac{0{,}26547(z+1).(z-1)}{(z-0{,}1859)(z-0{,}9488)(z-1) + 0{,}208(z-0{,}9488).0{,}26547(z+1)}$$

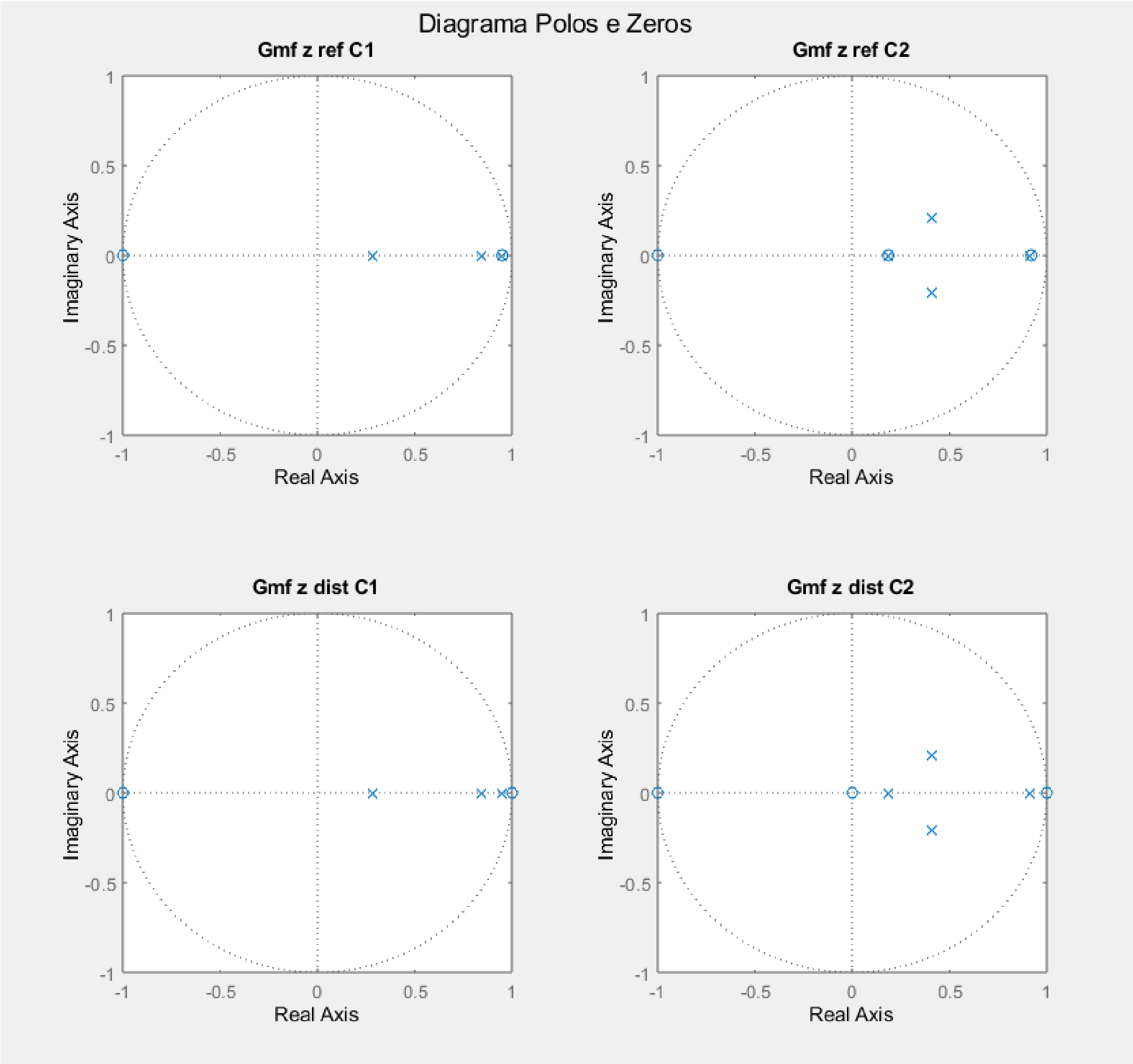
* Controlador 2 [PID]

$$\frac{y_2(z)}{Q(z)} = \frac{0{,}26547(z+1).z(z-1)}{(z-0{,}1859)(z-0{,}9488).z(z-1) + 0{,}26547(z+1)0{,}8(z-0{,}92)(z-0{,}1859)}$$

Então se utilizou o código abaixo para verificar os diagramas de polos e zeros:

```
clc; clear all;

%% FT Planta
s=tf('s');
Gma = zpk((0.95*13.4) / ((0.5*s + 1)*(16*s + 1)))


%% Ts
step_result = stepinfo(Gma);
tsd = step_result.SettlingTime/3
Ts = tsd/25 %0.76s


%% discretização com casamento de polos e zeros:
Gma_z = c2d(Gma, Ts, 'matched')


%% Controladores
z = tf('z', Ts);

% controlador 1 [PI]
C_z_1 = zpk(0.208*(z - 0.9488) / (z - 1))

% controlador 2 [PID]
C_z_2 = zpk(0.8*( (z - 0.92)*(z - 0.1859) ) / ( z*(z - 1) ))


%% FTMFs / ref
Gmf_z_ref_C1 = feedback(minreal(Gma_z*C_z_1), 1);
Gmf_z_ref_C2 = feedback(minreal(Gma_z*C_z_2), 1);

%% FTMFs / Dist.
Gmf_z_dist_C1 = (0.26547*(z+1)*(z-1)) / ((z-1)*(z-0.9488)*(z-0.1859) + 0.05521*(z-0.9488)*(z+1));
zpk(Gmf_z_dist_C1)

Gmf_z_dist_C2 = (0.26547*z*(z+1)*(z-1)) / (z*(z-1)*(z-0.9488)*(z-0.1859) + 0.2123*(z-0.1859)*(z-0.92)*(z+1));
zpk(Gmf_z_dist_C2)

figure(1)
subplot(2,2,1)
pzmap(Gmf_z_ref_C1)
title('Gmf z ref C1')
%figure;
subplot(2,2,2)
pzmap(Gmf_z_ref_C2)
title('Gmf z ref C2')
%figure;
subplot(2,2,3)
pzmap(Gmf_z_dist_C1)
title('Gmf z dist C1')
%figure;
subplot(2,2,4)
pzmap(Gmf_z_dist_C2)
title('Gmf z dist C2')

sgtitle('Diagrama Polos e Zeros')
```

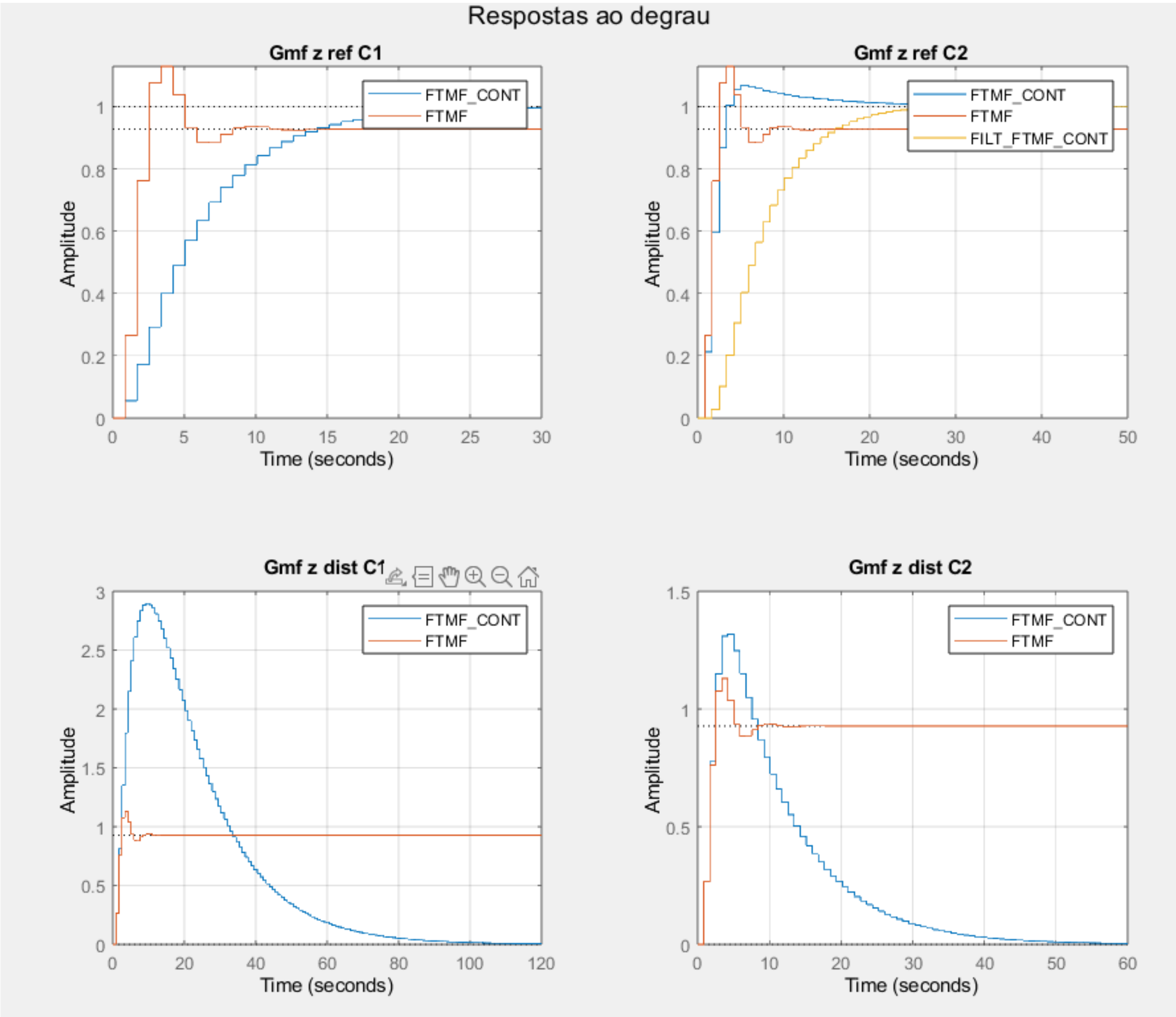
E se obteve o diagrama de polos e zeros:



O resultado dos polos e zeros estão dentro do observado nas questões anteriores, o controlador PID não consegue anular o overshoot sem a utilização do filtro, como previsto um dos pares de polos acabam possuindo parte complexa e consequentemente um pequeno overshoot. Quanto ao mapa do disturbio, se observa que para ambos os controladores, polos que eram cancelados por zeros passam a não mais ser cancelados voltando a ter influência na dinâmica do sistema, não apenas isto como os zeros passam a se localizar em cima do circulo únitário, acelerando a dinâmica do sistema em relação a resposta ao distrubio, refletindo no alto overshoot.

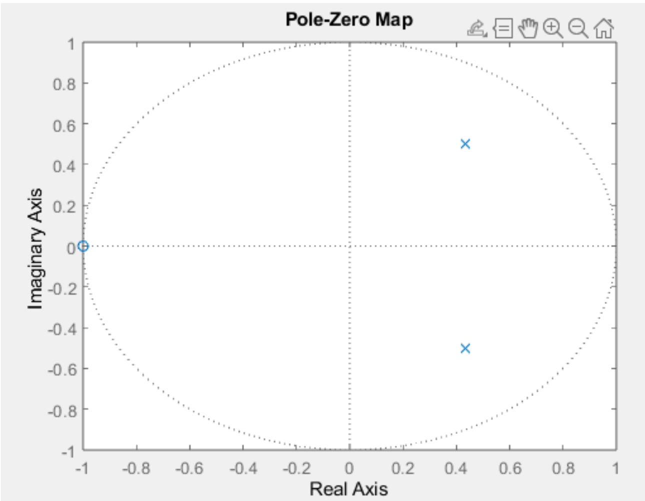
Em seguida, se utilizou o código abaixo para verificação das respostas do disturbio:

```
%% Reposta ao degrau com filtro de referência
pos_pf2 = 0.87;
F_C2 = (1 - pos_pf2) / (z - pos_pf2)

f_Gmf_z_ref_C2 = F_C2*Gmf_z_ref_C2;

figure(1)
subplot(2,2,1)
step(Gmf_z_ref_C1, feedback(Gma_z,1))
legend('FTMF_CONT', 'FTMF')
title('Gmf z ref C1')
grid on; box on;
si_c1 = stepinfo(Gmf_z_ref_C1)

subplot(2,2,2)
step(Gmf_z_ref_C2, feedback(Gma_z,1), f_Gmf_z_ref_C2)
legend('FTMF_CONT', 'FTMF', 'FILT_FTMF_CONT')
title('Gmf z ref C2')
grid on; box on;
si_c2 = stepinfo(Gmf_z_ref_C2)
si_fc2 = stepinfo(f_Gmf_z_ref_C2)

subplot(2,2,3)
step(Gmf_z_dist_C1, feedback(Gma_z,1))
legend('FTMF_CONT', 'FTMF')
title('Gmf z dist C1')
grid on; box on;

subplot(2,2,4)
step(Gmf_z_dist_C2, feedback(Gma_z,1))
legend('FTMF_CONT', 'FTMF')
title('Gmf z dist C2')
grid on; box on;

sgtitle('Respostas ao degrau')
```



De modo geral, os controladores tenderam a atrasar o fim da resposta transiente dos sistemas, porém permitiram as mesmas seguirem a referência proposta, e, no caso do controlador PI, remover o overshoot. Em relação ao distrubio, ambos os controladores conseguiram rejeitar o efeito do disturbio, isso se deve aos polos próximo do circulo únitario que agem como um integrador (zerando o erro em estado estacionario), observaveis no diagrama de polos e zeros da resposta ao disturbio, ditos polos deixam de serem cancelados pelos zeros na reposta ao disturbio.

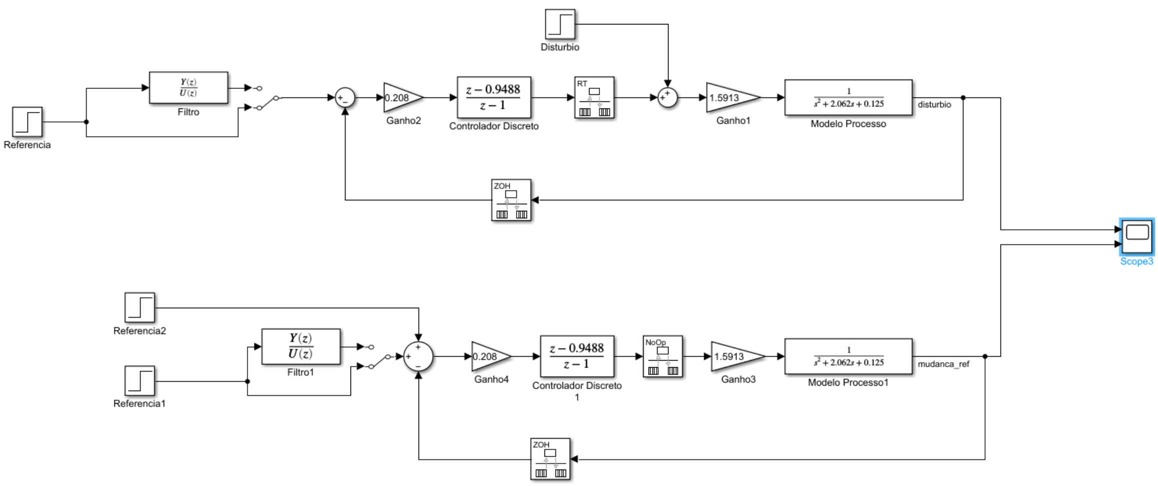
Toda via, se observa também que o controlador amplifica o pico das resposta ao disturbio, se análisa então o diagrama de polos e zeros do sistema de malha fechada sem controlador, que vem a ser o mesmo diagrama de polos e zeros do sistema de malha fechada em relação ao disturbio:



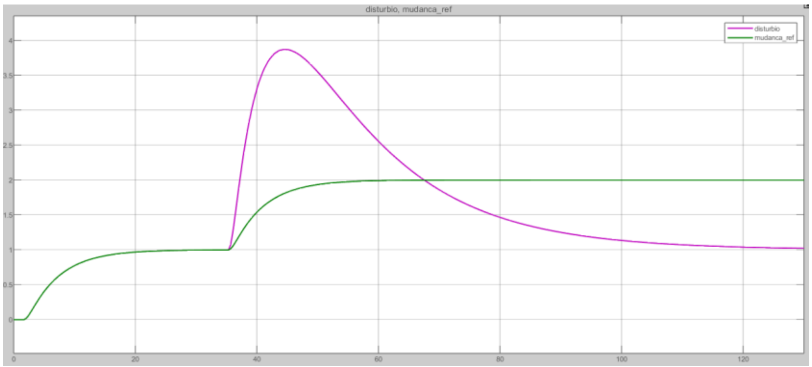
A ausencia de polos mais lentos (próximos ao circulo unitário) explica a rapidez do sistema para sair da reposta transitória, porém a ausencia de polos próximos do circulo únitario impede que o mesmo siga referências. O alto pico do disturbio dos sistemas controlados pode ser atribuido a presença de multiplos zeros dominantes que aceleram a reposta do sistema.

“A resposta ao distúrbio tem a mesma dinâmica que a resposta a mudanças de referência?” Não, diferente da resposta a mudança de referência, o disturbio apenas é afetado pela ação do controlador depois de o sistema ser realimentado pelo erro. Enquanto a resposta a mudança de referência é tratada imediatamente pela ação do controlador.

Para verificação dessas duas situações se fez o seguinte modelo de simulação do primeito controlador PI na ferramenta simulink:

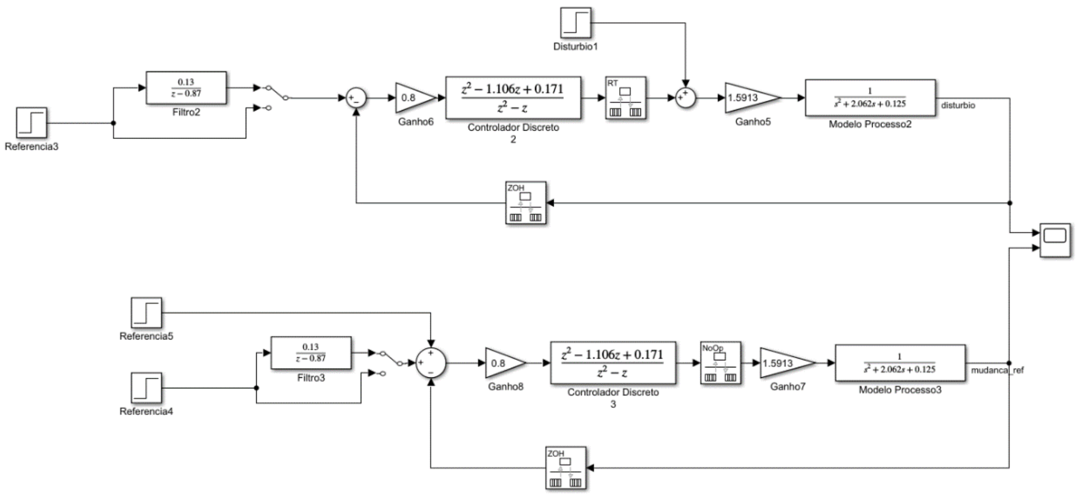


A partir do qual se obteve as respostas:

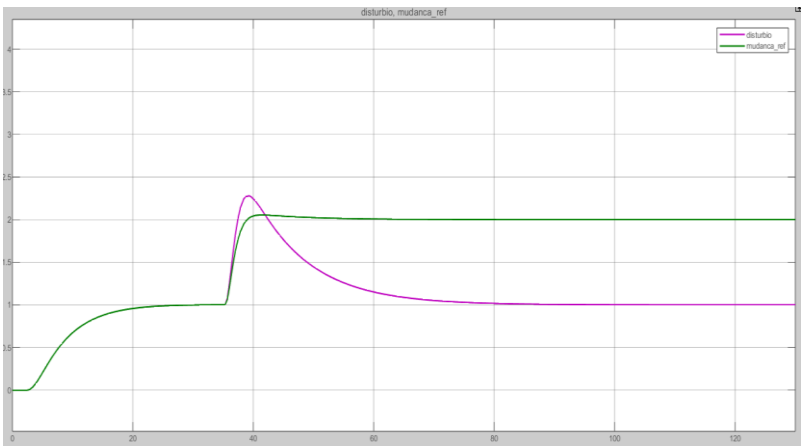


Se observa que o disturbio segue referência, mas possui um enorme sobressinale e tem um tempo de assentamento ainda maior que o sistema de malha aberta.

Para o controlador PID se utilizou o seguinte modelo de simulação:



A partir do qual se obteve a resposta:



Como observado anteriormente, o controlador PID não apenas segue referência como possui um sobressinal consideravelmente menor, além de retornar a referência mais rapido. Não se análisa, em ambos os controladores, a ação do filtro de referência porque este não tem efeito sobre o disturbio.

3. **(Projeto L.G.R.)**

a) Ajuste um controlador PID discreto por L.G.R tal que a resposta do sistema para referência e de rejeição de distúrbio, em malha-fechada (MF), tenha tempo de assentamento menor que 19 segundos e sobressinal menor que 20%.

b) Mostre o projeto e o Lugar das Raízes e especifique os detalhes e todas as etapas de projeto. Mostre os somatórios de fase e justifique o posicionamento dos zeros e dos polos do controlador projetado.

Inicialmente se calculou o tempo de amostragem e então se discretizou a planta:

```
clc; clear all;

S = tf('s');
Gma = zpk((1.59) / ((S + 2)*(S + 0.0625)) )

tsd = 19;
Mp = 0.2;

zeta = sqrt( (log(Mp)^2) / (pi^2 + log(Mp)^2))
wn = 4 / (zeta*tsd)

%Ts = 2*pi/(wn*30) %0.4536
% resultou em polo bem alto (>0.97)
Ts = tsd/25 %0.76

Gma_z = c2d(Gma, Ts, 'matched')

```

```
Gma =

      1.59
  ----------------
  (s+2) (s+0.0625)

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

```
Ts =

    0.7600


Gma_z =

     0.23051 (z+1)
  ---------------------
  (z-0.2187) (z-0.9536)

Sample time: 0.76 seconds
Discrete-time zero/pole/gain model.
```

```
zeta =

    0.4559


wn =

    0.4617
```

Foram calculadas também as constantes da EC desejada a partir das especificações de desempenhos passadas, estas serão utilizadas para calcular os polos dominantes desejados, estes serão a base para o projeto do controlador. Poderia se calcular o tempo de amostragem a partir da frequência natural do sistema obtida, porém, a mesma resultou em um polo discreto muito próximo do circulo unitário, indicando uma frequência de amostragem muito alto.

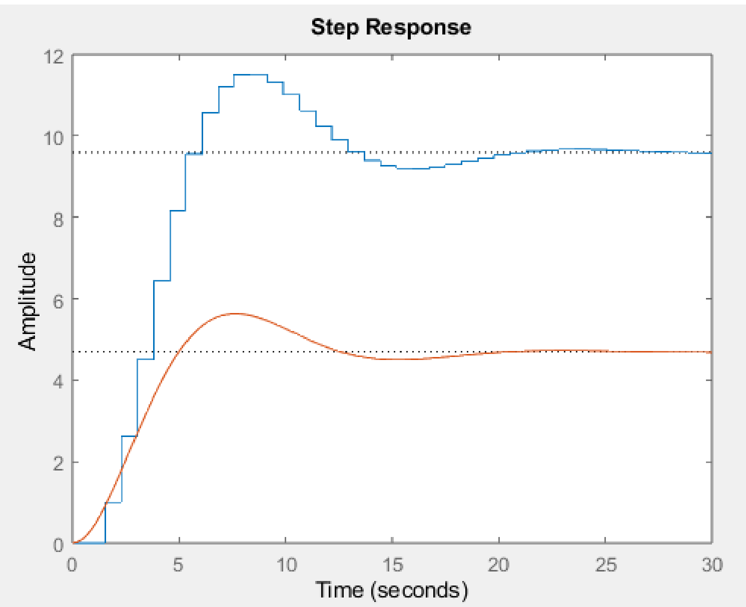
Para cálculo dos polos dominantes desejados, se utilizou o seguinte código abaixo:

```

%% Calculo dos polos desejados
j = sqrt(-1);
real_pd = wn*zeta;
cmpl_pd = j*wn*sqrt(1-zeta^2);

Pd1 = real_pd + cmpl_pd
Pd2 = real_pd -1*cmpl_pd

Pdz1 = exp(Ts*(-real_pd + cmpl_pd))
Pdz2 = exp(Ts*(-real_pd - cmpl_pd))

Z = tf('z', Ts)
model_controler = 1 / ( (S + Pd1)*(S + Pd2) );
model_controler_z = 1 / ( (Z - Pdz1)*(Z - Pdz2) );

stepinfo(model_controler)
stepinfo(model_controler_z)

step(model_controler_z)
hold on
step(model_controler)
hold off
```

De onde se obteve:

```
Pd1 =

  0.2105 + 0.4109i


Pd2 =

  0.2105 - 0.4109i


Pdz1 =

  0.8109 + 0.2618i


Pdz2 =

  0.8109 - 0.2618i
```

Resposta da equação característica desejada:



Desempenho da EC desejada:

```
ans =

  struct with fields:

        RiseTime: 3.8000
    SettlingTime: 19
     SettlingMin: 9.2012
     SettlingMax: 11.5107
       Overshoot: 20.0658
      Undershoot: 0
            Peak: 11.5107
        PeakTime: 8.3600
```

Sabendo os polos dominantes desejados, será utilizada a condição de fase para cálculo da posição dos zeros do controlador. Esta diz que a contribuição de fase de cada polo e zero da planta e do controlador deve resultar em -180º, tendo dois zeros para determinar a posição, se assume que cada um tem a mesma contribuição de fase para possibilitar a resolução da equação. Sabendo a contribuição de fase de cada zero do controlador em relação aos polos desejados, é possível, a partir de trigonometria, cálcular a posição destes na reta real. O código abaixo foi utilizado para cálculo da contribuição de fase de cada polo/zero:

```

%% Função para calculo da contribuição de fase
function phase = phase(sys,x,units)
  if nargin < 3 || isempty(units)
    units = 'deg';
  end

  phase = atan2d(imag(evalfr(sys,x)),real(evalfr(sys,x)));

  if strcmpi(units,'rad')
    phase = phase*(pi/180);
  end
end
```

```

%% Cálculo contribuições de fase

teta_pd = -pi; %-180° da condição de fase

% é dada a entrada manual nos zeros do sistema G*C
zeros = [-1];
cont_zeros = [];
if ~ isempty(zeros)
  for ind = 1:length(zeros)
    zero_phase= phase(zpk([zeros(ind)],[],1), Pdz1, 'rad');
    cont_zeros = [cont_zeros, rad2deg(zero_phase)];
    teta_pd = teta_pd - zero_phase;
  end
end

% é dada a entrada manual nos polos do sistema G*C
poles = [0.2187 0.9536 0 1]; % Ts = 0.76
cont_polos = [];
if ~ isempty(poles)
  for ind = 1:length(poles)
    pole_phase = phase(zpk([],[poles(ind)],1), Pdz1, 'rad');
    cont_polos = [cont_polos, rad2deg(pole_phase)];
    teta_pd = teta_pd - pole_phase;
  end
end
```

Além do código foi feito os cálculos manualmente para validar os resultados obtidos:

10

* Verificação contribuição de fase polos e zeros.

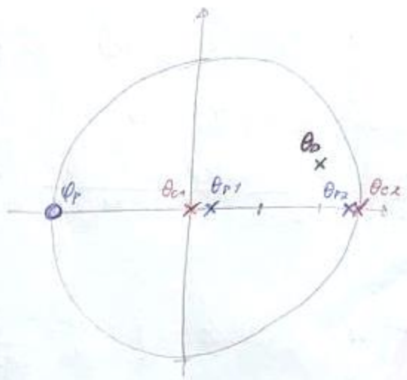
$G(z) = \dfrac{0{,}23051(z+1)}{(z-0{,}2187)(z-0{,}9536)}$ | $C(z) = \dfrac{K_c(z-z_1)(z-z_2)}{z(z-1)}$ | $P_{dz} = 0{,}8109 \pm 0{,}2618j$

lista zeros: $-1; z_1; z_2$

lista polos: $0{,}2187; 0{,}9536; 0; +1$

C.F.: $\varphi_P + \varphi_{c1} + \varphi_{c2} - \theta_{P1} - \theta_{P2} - \theta_{c1} - \theta_{c2} = -180°$;

$\varphi_{c1} + \varphi_{c2} = -180° - \varphi_P + \theta_{P1} + \theta_{P2} + \theta_{c1} + \theta_{c2}$



$$\varphi_P = tg^{-1}\left(\frac{0{,}2618}{1{,}8109}\right) = 8{,}2262°$$

$$\theta_{P1} = tg^{-1}\left(\frac{0{,}2618}{0{,}8109-0{,}2187}\right) = 23{,}8492°$$

$$\theta_{P2} = 180° - tg^{-1}\left(\frac{0{,}2618}{0{,}9536-0{,}8109}\right) = 118{,}5936°$$

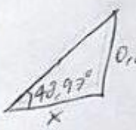
$$\theta_{c1} = tg^{-1}\left(\frac{0{,}2618}{0{,}8109}\right) = 17{,}8927°$$

$$\theta_{c2} = 180° - tg^{-1}\left(\frac{0{,}2618}{1-0{,}8109}\right) = 125{,}8408°$$

$$\varphi_{c1} + \varphi_{c2} = -180° - 8{,}2262° + 23{,}8492° + 118{,}5936° + 17{,}8927° + 125{,}8408°$$

$$\varphi_{c1} + \varphi_{c2} = 97{,}9501$$

* se $\varphi_{c1} = \varphi_{c2} \Rightarrow 2\varphi_c = 97{,}9501 \Rightarrow \varphi_c = 97{,}9501/2 = 48{,}975°$



$\Rightarrow x = \dfrac{0{,}2618}{tg(48{,}975°)} = 0{,}2277$ $\therefore z_c = 0{,}8109 - 0{,}2277 = 0{,}5839$

* Se $\varphi_{c1} = 2\varphi_{c2} \Rightarrow 2\varphi_{c2} + \varphi_{c2} = 97{,}9501$

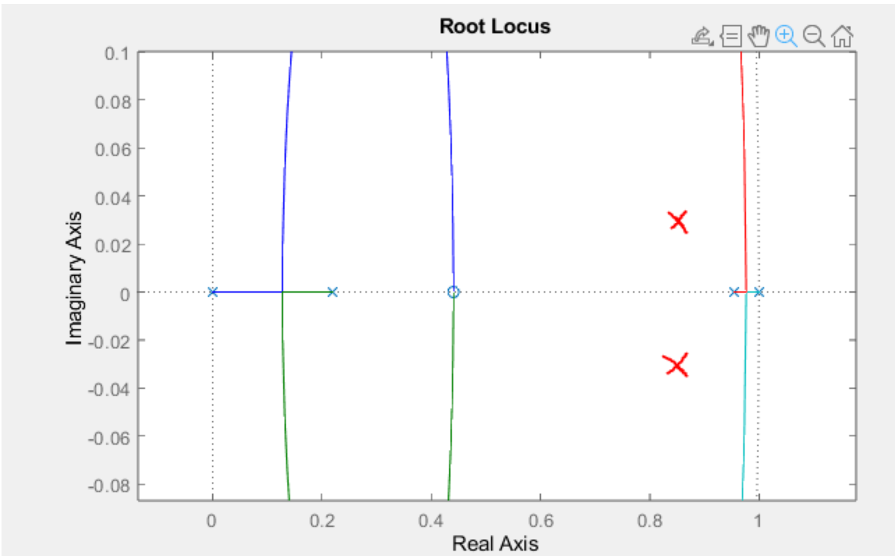
$\varphi_{c2} = 97{,}9501/3 = 32{,}65°$ | $\varphi_{c1} + 32{,}65 = 97{,}9501$; $\varphi_{c1} = 97{,}9501 - 32{,}65$; $\varphi_c = 65{,}3°$

$\Rightarrow z_{c1} = \dfrac{0{,}2618}{tg(32{,}65°)} = -0{,}4085 + 0{,}8109$; $z_{c1} = 0{,}4024$ | $z_{c2} = 0{,}8109 - \dfrac{0{,}2618}{tg(65{,}3°)} = 0{,}6904$

Os resultados obitidos via matlab correspondem aos obitidos pelos cálculos manuais.

```
cont_polos =

  -23.8512 -118.5870  -17.8944 -125.8344

>> cont_zeros

cont_zeros =

    8.2271

>> rad2deg(teta_pd)

ans =

   97.9400
```

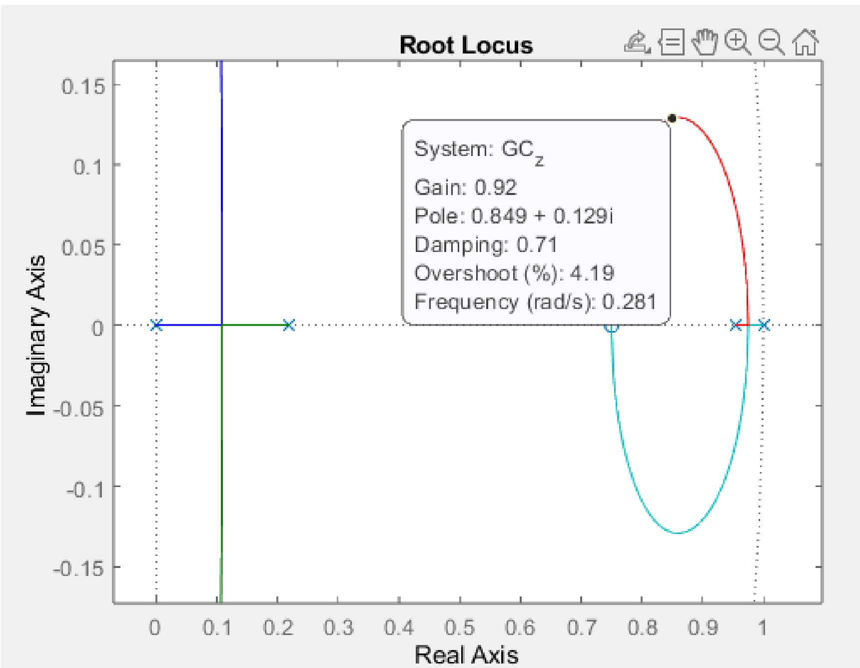
Toadvia, a posição dos zeros obtida pela condição de fase não foi satisfatória, pelo lugar das rapizes se observa que o zero cálculado se encontra muito distante dos polos desejados, fazendo que o lugar das raizes não passem próximas do mesmo. Desta forma, nenhum ganho fará com que se obtenha a resposta desejada.



Os x's desenhados representam aporoximadamente a posição desejada dos polos dominantes desejados. Portanto, para que o sistema atinja as especificações desejadas tentou-se alterar a distribuição da contribuição de fase de cada zero do controlador, de forma que um deles se aproxime mais dos polos desejados e, portanto, 'puxe' o lugar das raizes para mais próximo destes. Porém, como observavel abaixo, isto não foi o suficiente pois quanto mais se aproximava um zero do Polos dominantes desejados, o outro zero era afastado atrapalhando a dinâmica do sistema:

```
%% Dividindo igualmente a contribuição de fase de ambos os zeros do controlador:
% teta_pd = teta_pd/2;
% z1 = real(Pdz1) - (imag(Pdz1)/tan(teta_pd))
% z2 = real(Pdz1) - (imag(Pdz1)/tan(teta_pd));

%% Dividindo assimétricamente contribuição de fase:
%perc = 0.9; % 0.8024 ; -0.70 => ts = 28.2 | Mp = 46.3
%perc = 0.8; % 0.7569 ; 0.1436 => ts = 17.4 | Mp = 40.9
%perc = 0.7; % 0.7081 ; 0.3459 => ts = 16.7 | Mp = 37.2
%perc = 0.6; % 0.6521 ; 0.4896 => ts = 16.7 | Mp = 34.4
%perc = 0.5; % 0.5831 ; 0.5831 => ts = 16.7 | Mp = 33.5 % o mesmo que igualar os zeros
% abaixo de 0.5 irá repetir os casos anteriores
% perc = 0.4;
% z1 = real(Pdz1) - (imag(Pdz1)/tan(teta_pd*perc))
% z2 = real(Pdz1) - (imag(Pdz1)/tan(teta_pd*(1-perc)))
```

Apesar de facilmente se obter um tempo de assentamento satisfatório, nenhuma distribuição da contribuição de fase atendeu o sobressinal solicitado. Então, para que se atendessem as especificações de desempenho se ignorou a condição de fase e foi alterada a posição de ambos os zeros para 0.75, pois dessa forma o lugar das raizes se proxima muito mais dos polos desejados, como observado pelo LGR abaixo:



De posse então da posição dos zeros desejados, se cálcula qual o ganho do controlador Kc que irá levar os polos dominantes para próximo dos polos desejados através da condição do modulo, isto é, o ganho do controlador não deve alterar a resposta do sistema, ou seja, o modulo do controlador vezes a planta em relação ao polo desejado deve ser igual a um. Havendo apenas o ganho do controlador como variavel, é possível cálcular o modulo de cada zero e polo da planta e do controlador e resolver a equação isolando Kc. No matlab é possível fazer este cálculo invertendo a planta e o controlador, e calculando o modulo da multiplicação de ambos, como no código apresentado abaixo:

```
%% Cálcuclo condição de ganho
z1 = 0.75; z2 = 0.75;
C_z = ((Z - z1)*(Z - z2))/(Z*(Z-1))
INV = C_z^(-1) * Gma_z^(-1);
Kc = norm(evalfr(INV,Pdz1))
C_z = zpk(Kc*C_z)
```

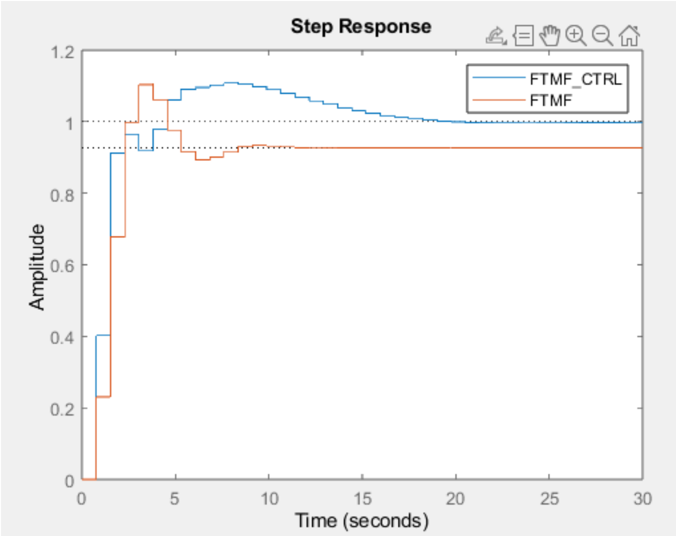
```
Kc =

   1.7433
```

De onde se obtem Kc igual a 1.7433. E, com o ganho do controlador e a posição dos zeros, o controlador está completo, basta então verificar o desempenho do mesmo, apresentado a seguir:

```
%% Verificação desempenho do controlador
GC_z = zpk(C_z*Gma_z)
GC_mf_z = feedback(GC_z, 1)
G_mf_z = feedback(Gma_z, 1);

%figure(1)
%subplot(1,2,1)
%rlocus(GC_z)

%subplot(1,2,2)
step(GC_mf_z, G_mf_z)
legend("FTMF_CTRL", "FTMF")

stepinfo(GC_mf_z)
```

```
ans =

  struct with fields:

        RiseTime: 0.7600
    SettlingTime: 15.9600
     SettlingMin: 0.9124
     SettlingMax: 1.1081
       Overshoot: 10.8056
      Undershoot: 0
            Peak: 1.1081
        PeakTime: 7.6000
```

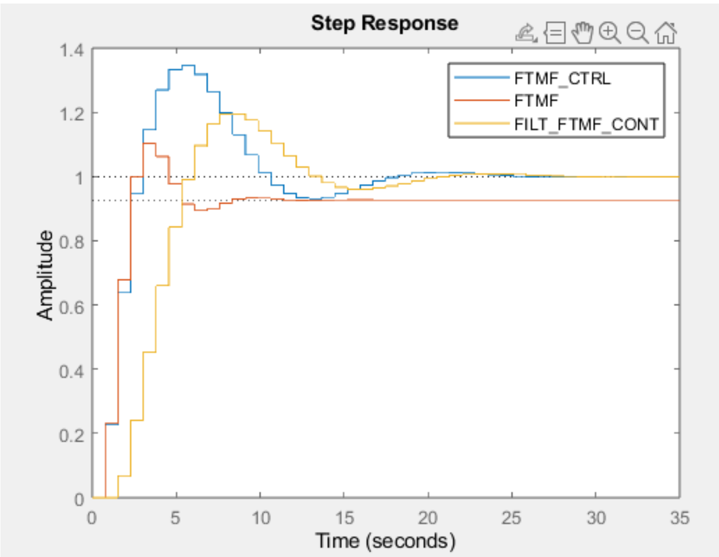
Gráfico da resposta ao degrau:



As especificações de desempenho pedidas foram atendidas com o ajuste necessario dos zeros.

Uma outra forma de melhorar a resposta do sistema é com um filtro de referência, invés de fazer a alteração dos zeros ignorando as contribuições de fase. Utilizando a distribuição da contribuição da fase de 6:4, e aplicando um filtro que acrescenta mais um polo dominantes para atraso do sistema, se obtem a seguinte dinâmica:

```
%% Teste resposta com filtro de referência
%pzmap(GC_mf_z)
%pos_pf1 = 0.652; %zero dominante da MF
  % ts = 19s | Mp = 23.26
pos_pf1 = 0.7;
  % ts = 19s | Mp = 19.5
z = tf('z', Ts);
F_z = (1 - pos_pf1)/(z - pos_pf1)

f_GC_mf_z = F_z*GC_mf_z;
step(GC_mf_z, G_mf_z, f_GC_mf_z);
legend("FTMF_CTRL", "FTMF", "FILT_FTMF_CONT")

si_f_GC_mf_z = stepinfo(f_GC_mf_z)
```




```
si_f_GC_mf_z =

  struct with fields:

        RiseTime: 3.0400
    SettlingTime: 19
     SettlingMin: 0.9602
     SettlingMax: 1.1952
       Overshoot: 19.5159
      Undershoot: 0
            Peak: 1.1952
        PeakTime: 8.3600
```

c) Utilizando o diagrama polo-zero ($Y(z)/R(z)$ e $Y(z)/Q(z)$), discuta e conclua (previsão) do comportamento do domínio do tempo do sistema. Está aceitável ? Como poderíamos melhorá-la?

Abaixo foi feito o calculo da respoata ao disturbio:

$$C(z) = \frac{1{,}7433(z-0{,}75)^2}{z(z-1)} \qquad G(z) = \frac{0{,}23051(z+1)}{(z-0{,}2187)(z-0{,}9536)}$$

#FTMF $y(z)/R(z)$ $\left[\frac{y(z)}{R(z)} = \frac{N_C N_G}{D_C D_G + N_C N_G}\right]$

$$\frac{y(z)}{R(z)} = \frac{0{,}4018(z-0{,}75)^2(z+1)}{z(z-1)(z-0{,}2187)(z-0{,}9536) + 0{,}4018(z-0{,}75)^2(z+1)}$$

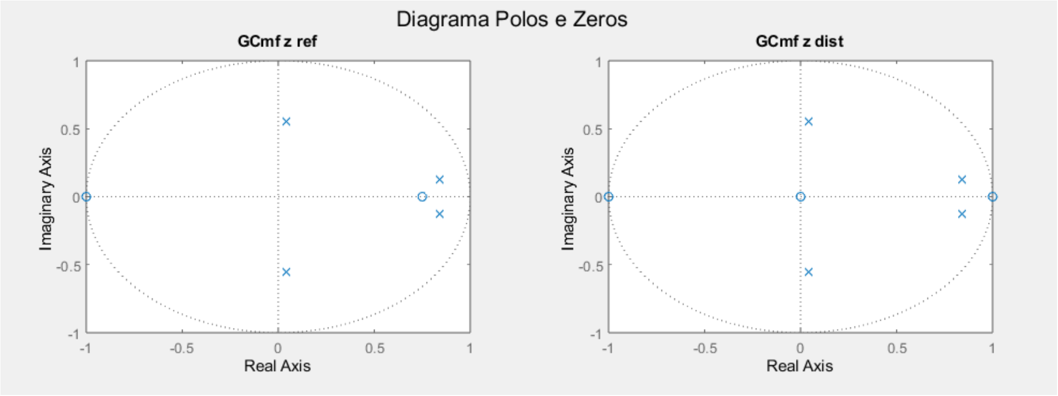
#FTMF $y(z)/Q(z)$ $\left[\frac{y(z)}{Q(z)} = \frac{N_G D_C}{D_C D_G + N_C N_G}\right]$

$$\frac{y(z)}{Q(z)} = \frac{0{,}23051(z+1)z(z-1)}{z(z-1)(z-0{,}2187)(z-0{,}9536) + 0{,}4018(z-0{,}75)^2(z+1)}$$

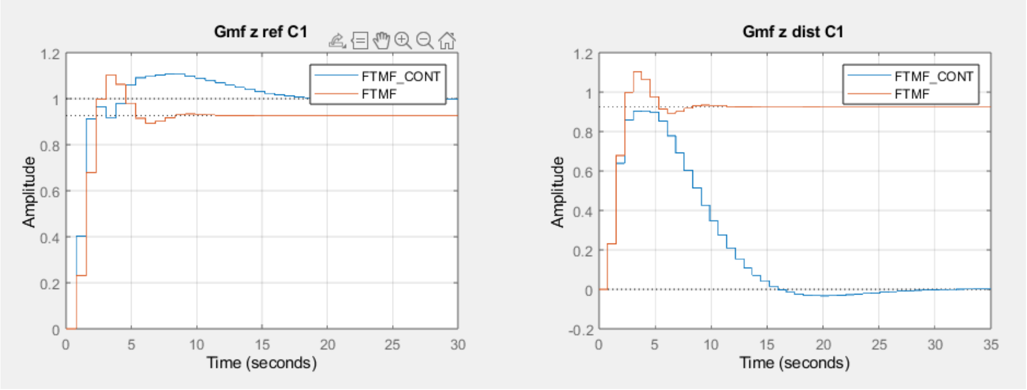
E então se utilizou o código abaixo para analisar os diagramas de polos e zeros das FTs.

```

%% Controlador
z = tf('z', Ts)
Kc = 1.7433;
z1 = 0.75; z2 = 0.75;
%z1 = 0.6521; z2 = 0.4896;
C_z = Kc*((z - z1)*(z - z2))/(z*(z-1))


%% Resposta do sistema da referência
GC_mf_z_ref = feedback(Gma_z*C_z, 1)

%% Resposta do sistema ao disturbio
G_mf_z_dist = (0.23051*z*(z+1)*(z-1)) / (z*(z-1)*(z-0.2187)*(z-0.9536) + Kc*0.23051*(z-z1)*(z-z1)*(z+1));

%% Diagrama de polos e zeros e Reposta ao degrau
figure(1)
subplot(2,2,1)
pzmap(GC_mf_z_ref)
title('GCmf z ref')

subplot(2,2,2)
pzmap(G_mf_z_dist)
title('GCmf z dist')

subplot(2,2,3)
step(GC_mf_z_ref, feedback(Gma_z,1))
legend('FTMF_CONT', 'FTMF')
title('Gmf z ref C1')
grid on; box on;

subplot(2,2,4)
step(G_mf_z_dist, feedback(Gma_z,1))
legend('FTMF_CONT', 'FTMF')
title('Gmf z dist C1')
grid on; box on;
```

E se obteve os diagramas:



A diferença entre o diagrama da resposta ao disturbio e o diagrama da planta é a presença de um zero na origem, um zero puramente de avanço, e o zero dominante que passa a estar em cima do circulo unitário, se tornando ainda mais dominante no sistema o que aceleraria mais ainda a resposta do ao disturbio. E, devido os polos dominantes serem polos complexos conjugados, e por estarem próximo da reta real, é esperrado uma oscilação pequena.



Ao analisar a resposta o disturbio se nota que a mesma não concorda exaramente com o previsto quanto a velocidade da resposta ao disturbio. Uma possivel explicação é que, o zero em 1 vem do controlador, e este passa a atuar na correção do disturbio apenas quando se computa a realimentação da malha fechada, o que causa um pequeno atraso no tempo de assentamento do disturbio em comparação com a resposta a referência.

Através do código desenvolvido, se testou a reposta do sistema para ganhos diferentes do controlador e aumentando o ganho para 2.1 se obteve pequeno aprimoramento no desempenho da resposta ao disturbio, o tempo de assentamento que era de 24.32 segundo passou a ser 22.04 segundos, enquanto a reposta Y(Z)/R(Z) se manteve dentro das especificações desejadas. Isso se deve aos polos dominantes de Y(Z)/Q(Z) terem sua parte real levemente diminuida, torando-os menos dominantes/atarsando menos a reposta do sistema. Não se pode toda via aumentar ainda o ganho, pois os polos não dominantes de Y(Z)/Q(Z) tornariam instaveis ao cruzar para ao lado esquerda do circulo unitário.

4. **(Projeto Controlador DeadBeat)**

a) Ajuste um controlador Deadbeat que garanta seguimento de referência com erro nulo. Mostre as etapas de projeto claramente e com destaque para resultados intermediários importantes. O controle rejeita distúrbios?

Para o controle deadbeat é presumido que já se possui a planta discretizada e, portanto, o tempo de assentamento não é um fator a ser considerado. Toda via, este tem uma influência na resposta do controle deadbeat, um tempo de amostragem muito abaixo resultará em uma ação de controle muito grande, enquanto um tempo de amostragem muito alto torna o sistema vulnerável a ruídos do disturbio. Como orientado previamente pelo professor, será desenvolvido o controle dead beat para diferentes tempos de amostragem para exemplificação do que foi dito anteriormente. Inicialmente, será utlizado o mesmo Ts utilizando nas questões anteriores de 0.84 segundos, em sequência serão refeitos os calculos para 2 e 3 vezes este Ts.

Para projeto do deadbeat é necessario calcular qual será a malha fechada teórica do sistema, para isso, incialmente, se faz uma divisão longa da FT da planta para se descobrir a partir de qual termo se terá uma resposta, os termos previos deste serão desconsiderados para os calculos.

* Verificando atraso do sistema:

$$G_p(z) = \frac{0{,}26526(z+1)}{(z-0{,}1859)(z-0{,}9488)} = \frac{0{,}2865z + 0{,}2865}{z^2 - 1{,}1347z + 0{,}1763} = \frac{0{,}2865z^{-1} + 0{,}2865z^{-2}}{1 - 1{,}1347z^{-1} + 0{,}1763z^{-2}}$$

* Divisão longa

$$\begin{array}{l|l} \cancel{0{,}2865}z^{-1} + 0{,}2865z^{-2} & 1 - 1{,}1347z^{-1} + 0{,}1763z^{-2} \\ \hline -(\cancel{0{,}2865z^{-1}} - 0{,}325z^{-2} + 0{,}05z^{-3}) & 0{,}2865z^{-1} - 0{,}0385z^{-2} + \dots \\ \hline -0{,}0385z^{-2} + 0{,}05z^{-3} & \end{array}$$

→ sistema começa a agir a partir do 1º tempo de amostragem.

Escolhendo p = n:

* $G_m(z) = \cancel{g_0}^{0} + g_1 z^{-1} + g_2 z^{-2}$, onde $g_0 = 0$ porque não existe resposta instantanea

Cálculo das constantes de Gmf(Z):

* Para se calcular os termos de Gmf(z), deve ser analizadas as condições que a planta deve atender. A mesma deve ter erro nulo, portanto $G_{mf}(1)=1$. E a mesma possui um zero sobre o círculo unitário advindo da discretização, o qual não pode ser cancelado, logo: $G_{mf}(-1)=0$.

* $G_{mf}(z) = g_1 z^{-1} + g_2 z^{-2}$

* $G_{mf}(1) = 1$

$$g_1(1)^{-1} + g_2(1)^{-2} = 1$$

$$g_1 + g_2 = 1$$

* $G_{mf}(-1) = 0$

$$g_1(-1)^{-1} + g_2(-1)^{-2} = 0$$

$$-g_1 + g_2 = 0$$

* resolvendo o sistema:

$\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ se obtem $g_1 = 0,5$ e $g_2 = 0,5$

De posse das constantes de Gmf basta apenas calcular o controlador.

Calculando o controlador.

$$G_c(z) = \frac{1}{G_p(z)} \cdot \frac{G_{mf}(z)}{[1 - G_{mf}(z)]} = \frac{z^2 - 1,7347z + 0,7763)}{0,2865z + 0,2865} \cdot \frac{(0,5z^{-1} + 0,5z^{-2})}{[1 - 0,5z^{-1} + 0,5z^{-2}]}$$

$$G_c(z) = \frac{1,885(z-0,7859)(z-0,9488)}{(z-1)(z+0,5)}$$

Dado que o valor das constantes de Gmf(Z) independem do tempo de amostragem Ts, o cáluclo do controlador deadbeat para os outros tempos de amostragem será o mesmo, mudando apenas o valor da planta Gp(Z), para simplificação dos cálculos se realizou estes pelo matlab.

O código escrito abaixo foi utilizado para cálculo dos controladores, com os resultados obtidos a direita:

```
%% 1*TS
Ts1 = tsd/25;
Gma_z1 = c2d(Gma, Ts1, 'matched')

z = tf('z', Ts1)
Gmf1 = 0.5*z^(-1) + 0.5*z^(-2)
Gc1 = minreal(( Gma_z1^(-1) )*( (Gmf1)/(1 - Gmf1) ))


%% 2*TS
Ts2 = Ts1*2;
Gma_z2 = c2d(Gma, Ts2, 'matched')

z = tf('z', Ts2)
Gmf2 = 0.5*z^(-1) + 0.5*z^(-2)
Gc2 = minreal(( Gma_z2^(-1) )*( (Gmf2)/(1 - Gmf2) ))


%% 3*TS
Ts3 = Ts1*3;
Gma_z3 = c2d(Gma, Ts3, 'matched')

z = tf('z', Ts3)
Gmf3 = 0.5*z^(-1) + 0.5*z^(-2)
Gc3 = minreal(( Gma_z3^(-1) )*( (Gmf3)/(1 - Gmf3) ))

Gmf1 = minreal(feedback(Gc1*Gma_z1, 1))
Gmf2 = minreal(feedback(Gc2*Gma_z2, 1))
Gmf3 = minreal(feedback(Gc3*Gma_z3, 1))
step(Gmf1, Gmf2, Gmf3);
legend("1*Ts","2*Ts","3*Ts")
grid on; box on;
```

```
Gc1 =

  1.885 (z-0.1859) (z-0.9488)
  ---------------------------
       (z-1) (z+0.5)
```

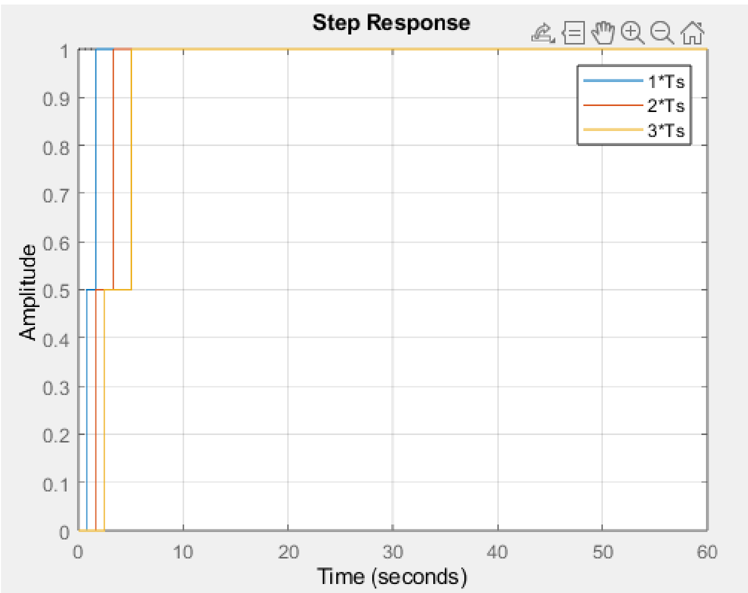
```
Gmf1 =

  0.5 (z+1)
  ---------
     z^2

Sample time: 0.84139 seconds
Discrete-time zero/pole/gain model.
```

```
Gc2 =

  0.81566 (z-0.03454) (z-0.9002)
  ------------------------------
       (z-1) (z+0.5)
```

```
Gmf2 =

  0.5 (z+1)
  ---------
     z^2

Sample time: 1.6828 seconds
Discrete-time zero/pole/gain model.
```

```
Gc3 =

  0.54215 (z-0.00642) (z-0.8541)
  ------------------------------
       (z-1) (z+0.5)
```

```
Gmf3 =

  0.5 (z+1)
  ---------
     z^2

Sample time: 2.5242 seconds
Discrete-time zero/pole/gain model.
```

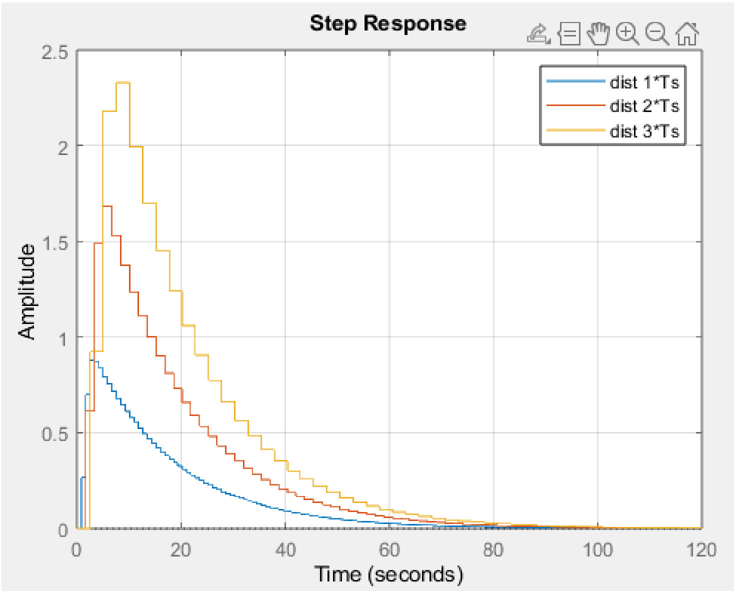


Pela resposta ao degrau observada a esquerda vê-se que o controlador Deadbeat cumpre com sua proposta, ele cancela a dinâmica da planta e alcança com tempo minimo (3 tempos de amostrafem nesse caso) a referência. O mesmo pode ser observado nas funções de transferência de malha fechada acima, a planta é cancelada (exceto o zero em -1), e sobra apenas a ação do controlador. Na prática tal controlador é impossível de ser implementado, dado que sua propria definição parte de uma inverdade matemática.

Para análise da resposta ao disturbio, se desenvolveu o seguinte código:

```
%% resposta ao disturbio
Gmf1_dist = Gma_z1 / (1 + Gma_z1*Gc1);
Gmf2_dist = Gma_z2 / (1 + Gma_z2*Gc2);
Gmf3_dist = Gma_z3 / (1 + Gma_z3*Gc3);
step(Gmf1_dist, Gmf2_dist, Gmf3_dist);
legend("dist 1*Ts","dist 2*Ts","dist 3*Ts")
grid on; box on;
```

De onde se obteve o gráfico:



O sistema rejeita disturbios, se observa que quanto menor o tempo de amostragem, mais rapido o controlador foi capaz de responder ao disturbio, por sua vez atenuando este. O controlador projetado para 3 vezes o Ts original obteve um pico de reposta maior do que o dobro do pico da resposta do sistema com o Ts original, pois o mesmo demora mais para começar a atuar no disturbio.