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

Avaliação Parcial 2 - Atividade Extra
Controle Discreto

17/08/2024

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

Problema 1 (Adaptado): Dada a função de transferência:

$$V(s) = \frac{0.5}{(1+0.5s)}\left[\frac{4}{(1+0.1s)}U(s) - T(s)\right], \quad (1)$$

sendo $V(s)$ a velocidade do robô no mesmo eixo, $T(s)$ o torque aplicado pela carga (caixa) e $U(s)$ o sinal de acionamento do servomotor. Neste modelo o torque de carga pode ser considerado nulo quando o robô se movimenta sem carga e igual a $T = 3P$ quando carregando uma caixa, sendo P o peso da mesma. Desta forma a perturbação para o sistema de posicionamento z é uma sequência de degraus de amplitude variável (positivos quando aumenta-se a carga e negativos quando a mesma diminui).

1. Considerando que a velocidade $V(s)$ é mensurada, projete um controle de velocidade que rejeite perturbações de T do tipo degrau em regime permanente e siga referências. Considere que o pico ($overshoot$) tem que ser adequado e que o tempo de assentamento desejado é 0.2 segundos. Este projeto deve ser realizado diretamente no domínio z. Defina a taxa de amostragem a ser usada no controlador e justifique. **Analise a solução no domínio do tempo e usando diagramas polo-zero. Analise detalhadamente o efeito das perturbações na saída do sistema.** As respostas estão adequadas?

A função de transferência da velocidade é:

$$V(s) = \frac{0,5}{(1+0,5s)}\left[\frac{4}{(1+0,1s)} U(s) - T(s)\right]$$

Para simplificação a mesma é reescrita:

$$V(s) = G_1(s)\left[G_2(s)\, U(s) - T(s)\right]$$

Para projeto do controlador a ação do distúrbio é ignorada, portanto se obtém a seguinte FTMA:

$$V(s) = G_1(s)\, G_2(s) \cdot U(s)$$

$$\frac{V(s)}{U(s)} = G_1(s)\, G_2(s) = \frac{0,5 \cdot 4}{(1+0,5s)(1+0,1s)}$$

$$\frac{V(s)}{U(s)} = \frac{2/(0,1 \cdot 0,5)}{(2+s)(10+s)} = \frac{40}{(2+s)(10+s)}$$

De posse da função de transferência de malha aberta, a mesma é escrita em um script do matlab para projeto do controlador. Devido o sistema ser de segunda ordem, se optou por um controlador PID, pois com dois zeros se consegue de forma melhor manipular o lugar das raizes, que é o metodo pelo qual será projetado o controlador.

Para discretização do sistema, se define o tempo de amostragem como 1/25 avos do tempo do tempo de assentamento. Em comparação com o recomendado pela literatura de utilizar a frequência de amostragem dentre 15 a 30 vezes a frequência natural do sistema, se obtem um valor bem próximo como observado ao lado, se opta pelo ultimo valor visto que o mesmo é pouco maior.

Em sequência se utiliza o código demostrado abaixo para discretização do sistema pelo metodo de mapeamento de polos e zeros com um Tempo de amostragem (Ts) de 0.0072 segundos.

```
>> Ts = 2*pi/(wn*30)

Ts =

   0.0066

>> Ts = tsd/25

Ts =

   0.0072
```

```
clc; clear all;

S = tf('s');
G1 = 0.5 / (1 + 0.5*S);
G2 = 4 / (1 + 0.1*S);
Gma = zpk( G1*G2 )

tsd = 0.2 * 0.9; % considerar 90% para dar uma margem de erro
Mp = 0.05 * 0.9; % considerar 90% para dar uma margem de erro

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

% resultou em polo bem alto (>0.97)
Ts = tsd/25 %

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

```

Devido o Ts ser um valor tão baixo (frequência de amostragem grande), os polos discretos se encontram bem próximos do círculo unitário, o que dificulta um o controle do sistema pois o mesmo se torna facilmente instável, e tende a necessitar de ganhos grandes para fornecer uma resposta adequada.

Se definiu um overshoot (Mp) de 5% como adequado.

Resultados obtidos abaixo:

```
Gma =

     40
  ------------
  (s+10) (s+2)

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

```
zeta =

   0.7025


wn =

   31.6328


Ts =

   0.0072
```

```
Gma_z =

   0.00099319 (z+1)
  ---------------------
  (z-0.9305) (z-0.9857)

Sample time: 0.0072 seconds
Discrete-time zero/pole/gain model.
```

A partir desses valores se calcula então os polos desejados, a partir dos quais serão definidos os parametros do controlador, se observa a reposta dos polos desejados para se certificar que é a resposta deseja:

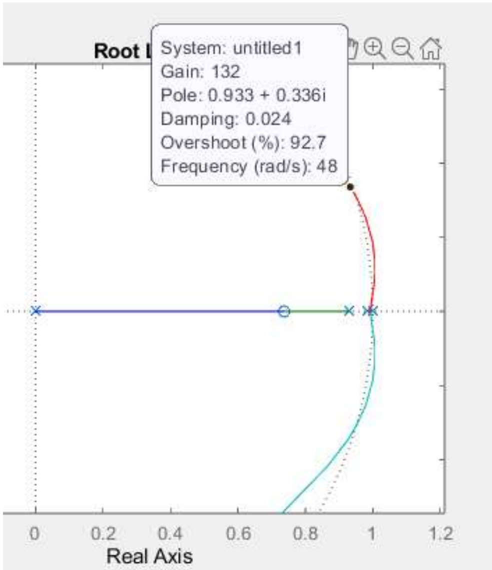
```
%% Calculo dos polos desejados
j = sqrt(-1);
real_pd = wn*zeta;
cmpl_pd = j*wn*sqrt(1-zeta^2);

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) );

model_controler = stepinfo(model_controler)
model_controler_z = stepinfo(model_controler_z)

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

```
Pd1 =

  22.2222 +22.5124i


Pd2 =

  22.2222 -22.5124i


Pdz1 =

   0.8410 + 0.1375i


Pdz2 =

   0.8410 - 0.1375i
```

```
model_controler =

  struct with fields:

        RiseTime: 0.0675
    SettlingTime: 0.1888
     SettlingMin: 9.0943e-04
     SettlingMax: 0.0010
       Overshoot: 4.4989
      Undershoot: 0
            Peak: 0.0010
        PeakTime: 0.1388


model_controler_z =

  struct with fields:

        RiseTime: 0.0720
    SettlingTime: 0.1944
     SettlingMin: 21.2589
     SettlingMax: 23.6482
       Overshoot: 4.5273
      Undershoot: 0
            Peak: 23.6482
        PeakTime: 0.1440
```

Em sequência, se utiliza a condição de fase para definir o lugar das raizes, porém, quando observado o lugar das raizes se nota que o valor obtido da posição dos zero pela condição de fases não é suficiente para se aproximar o lugar das raizes dos polos desejados. Se recorre a dividir a contribuição de fase de maneira assimétrica entre os dois zeros do controlador, porém, ainda com essa modificação não se atinge o Lugar das Raizes desejado.

Ainda mais, seria necessário um ganho grande para que o sistema se torne estável, e mesmo assim o mesmo não se aproxima dos polos desejados como mencionado previamente.

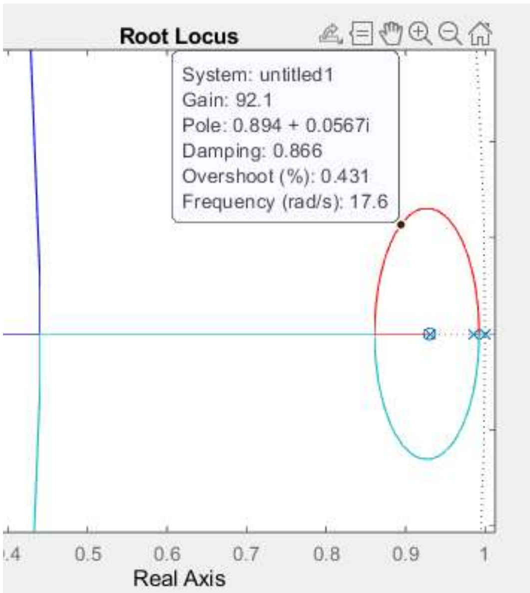


Abaixo o código utilizado para cálculo da contribuição de fase:

```
%% 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
```

Após alguns testes para posição dos zeros, se constatou que os dois zeros do controlador em cima do polo não dominantes da planta resulta em uma resposta perto da desejada, o tempo de assentamento está abaixo do 0.02 segundos mas o Mp está além do aceitável. Tal situação pode ser remediada com um filtro de referência, dado que o mesmo tende a reduzir o Mp e atrasar um a dinâmica da planta. Abaixo o lugar das raízes obtido que com os zeros do controlador definidos empiricamente:

```
%% Rlocus - verificação posicionamento polos/zeros:
% perc = 0.5;
% z1 = real(Pdz1) - (imag(Pdz1)/tan(teta_pd*perc))
% z2 = real(Pdz1) - (imag(Pdz1)/tan(teta_pd*(1-perc)))

%0.93
pos = 0.93; z1 = pos; z2 = pos;
C_z = ((Z - z1)*(Z - z2))/(Z*(Z-1))

rlocus(Gma_z*C_z)
```



Para definição do ganho se calcula o mesmo pela condição de modulo através do código abaixo, dado o ganho da planta em MA ser muito baixo, o ganho do controlador resultou em um valor alto:

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

```
Kc =

  119.2880
```

Controlador:

```
ans =

  (z-0.93)^2
  ----------
   z (z-1)
```

FTMF:

```
GC_mf_z =

         0.11848 (z+1) (z-0.93)^2
  ---------------------------------------------
  (z-0.1493) (z-0.8433) (z-0.8747) (z-0.9305)
```

E se obtém a resposta do sistema de malha fechada em relação a referência:

```
%% 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")

si_GC_mf_z = stepinfo(GC_mf_z)
```

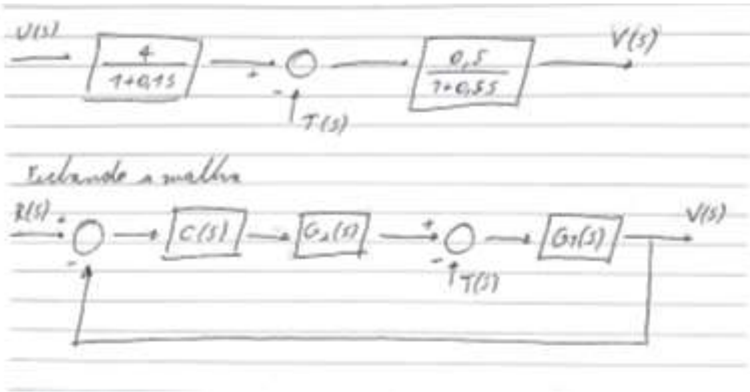
```
si_GC_mf_z =

  struct with fields:

        RiseTime: 0.0360
    SettlingTime: 0.2736
     SettlingMin: 0.9273
     SettlingMax: 1.1473
       Overshoot: 14.7282
      Undershoot: 0
            Peak: 1.1473
        PeakTime: 0.1008
```

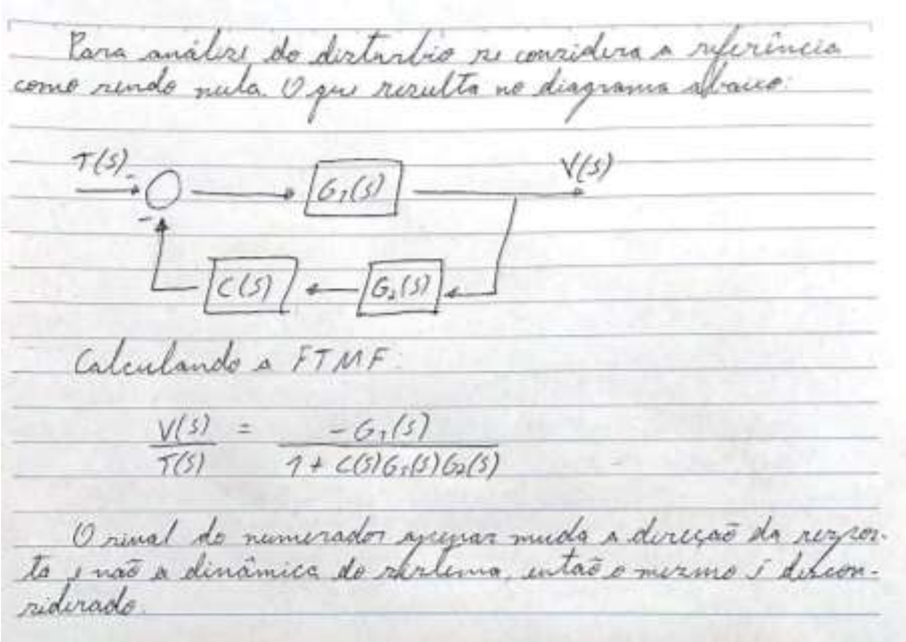
Se aplica então um filtro de referência, o mesmo foi obtido de maneira empírica, e através deste se obtém uma dinâmica satisfatória:

```
%% Teste resposta com filtro de referência
%pzmap(GC_mf_z)
pos_pf1 = 0.905;
z = tf('z', Ts);
F_z = (1 - pos_pf1)/(z - pos_pf1)

f_GC_mf_z = F_z*GC_mf_z;
resp_f_GC_mf_z = stepinfo(f_GC_mf_z)
```

```
resp_f_GC_mf_z =

  struct with fields:

        RiseTime: 0.1152
    SettlingTime: 0.1872
     SettlingMin: 0.9140
     SettlingMax: 1.0186
       Overshoot: 1.8605
      Undershoot: 0
            Peak: 1.0186
        PeakTime: 0.2736
```

Para verificação da resposta ao distúrbio, se analisa o diagrama de blocos da FT da velocidade:

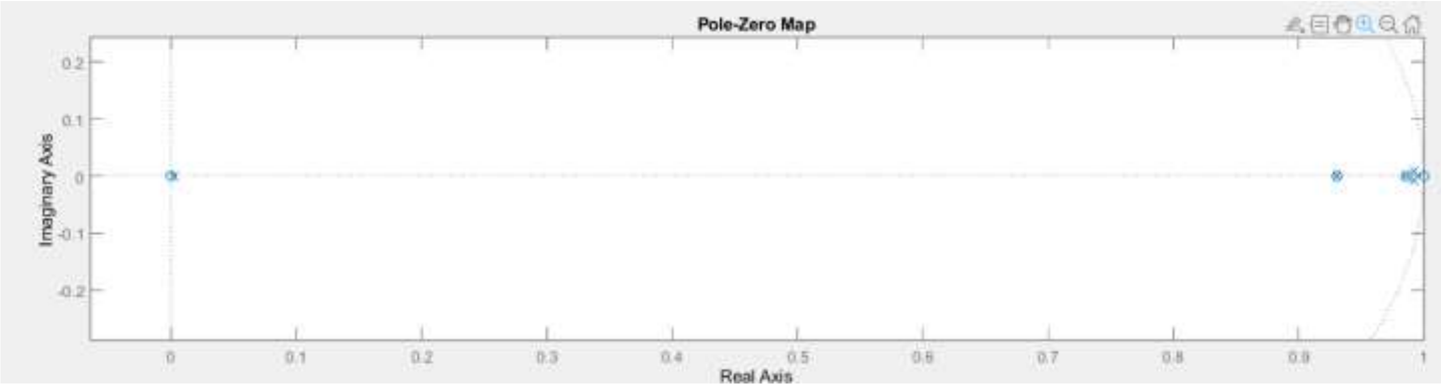




É calculada a FT da saída em relação ao distúrbio através do matlab, e se analisa o diagrama de polos e zeros:

```
%% verificação resposta ao disturbio

G1_z = c2d(G1, Ts, 'matched')
G2_z = c2d(G2, Ts, 'matched')

Gmf_z_dist = 1*G1_z / (1 + C_z*G1_z*G2_z)

pzmap(Gmf_z_dist)
resp_dist = stepinfo(Gmf_z_dist)
```

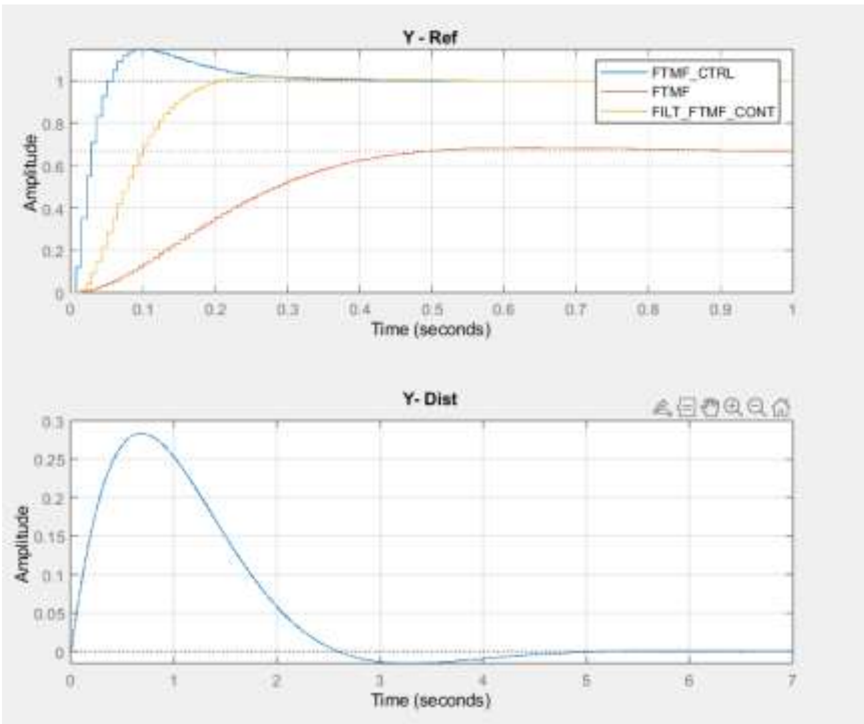
```
>> zpk(Gmf_z_dist)

ans =

        0.0071484 z (z-1) (z-0.9857) (z-0.9305)
  ----------------------------------------------------------
  (z-0.9857) (z-0.9305) (z-0.001876) (z^2 - 1.984z + 0.984)

Sample time: 0.0072 seconds
Discrete-time zero/pole/gain model.
```



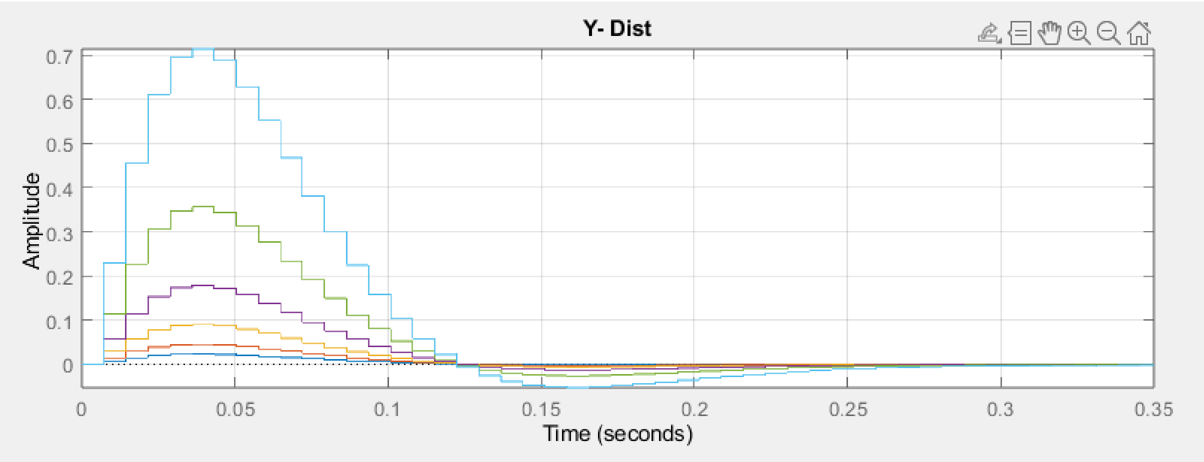
Se observa que há um polo bastante lento que não é completamento sobreposto por um zero, logo o mesmo atrasa a resposta do sistema.

Abaixo se observa a reposta do sistema em relação a referência e em relação ao distúrbio, assim como o código utilizado para se obter a mesma:

```
%% Plotagem
figure(1)

subplot(2,1,1)
step(GC_mf_z, G_mf_z, f_GC_mf_z);
grid on; box on;
legend("FTMF_CTRL","FTMF","FILT_FTMF_CONT")
title('Y - Ref')

subplot(2,1,2)
for ind = 0:5
    %step_config = RespConfig('Amplitude',2^ind); %não existe na minha versão
    step_config = stepDataOptions('StepAmplitude',2^ind);
    step(Gmf_z_dist, step_config);
    hold on;
    grid on; box on;
    title('Y- Dist')
end
 hold off;
```



A resposta do distúrbio está inadequada, tanto o Mp quanto o tempo de assentamento do mesmo estão ordens de magnitude acima do aceitável. Ao se mover os zeros do controlador um pouco de 0.9305 para 0.905 se obtém uma reposta completamente diferente como observável abaixo:



Tanto o overshoot quanto o tempo de assentamento estão adequados. Se atribui tal mudança tão brusca ao não cancelamento de um dos zeros do controlador com o polo da planta que ocorria no posicionamento anterior. Observando o novo diagrama de polos e zeros abaixo se nota que antes os polos dominantes do sistema estavam bastante próximos do círculo unitário, causando uma dinâmica lenta, no novo diagrama os mesmos se distanciaram. Tal mudança tão brusca no posicionamento dos polos e zeros pode ser atribuída ao alto ganho do controlador, que influencia diretamente no cálculo da FTMF de V(S)/T(S).