

```
% limpa memoria e fecha figuras
clear all
close all
```

Define o sistema

```
% Sistema
num = [0 1 2];
den = [1 1 0.5];
G = tf(num,den)
```

```
G =
```

$$\frac{s + 2}{s^2 + s + 0.5}$$

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

```
numinv = fliplr(num)';
deninv = fliplr(den)';
```

Monta polinômio desejado para malha fechada

Note que os polos são os especificados nos slides, correspondendo a $-0.8 \pm 1.324j$ e -8 .

```
% especificacao de malha fechada
D = conv(conv([1 0.8 + 1.324i], [1 0.8 - 1.324i]), [1 8])
```

```
D = 1x4
    1.0000    9.6000   15.1930   19.1438
```

```
Dinv = fliplr(D)'
```

```
Dinv = 4x1
    19.1438
    15.1930
     9.6000
     1.0000
```

Resolve a equação Diofantina

```
% resolve diofantina
E = [[deninv; 0]; [0; deninv]];
E = [E [numinv; 0] [0; numinv]];
M = E \ Dinv
```

```
M = 4x1
     9.6631
     1.0000
     7.1561
    -1.0631
```

Recupera o controlador

Neste caso, volte no texto de referência para verificar a ordem dos parâmetros.

```
% recupera o controlador:  
C = tf([M(4) M(3)], [M(2) M(1)])
```

```
C =  
  
    -1.063 s + 7.156  
    -----  
         s + 9.663  
  
Continuous-time transfer function.
```

Monta a malha fechada

São construídas duas malhas fechadas com o mesmo controlado. No primeiro caso, o controlador é colocado em série com o processo e no segundo, ele é colocado no ramo de realimentação. Veja a apresentação dos slides para acompanhar a diferença nos zeros da malha fechada.

```
% malhas fechadas  
MF1 = feedback(series(C,G),1)
```

```
MF1 =  
  
    -1.063 s^2 + 5.03 s + 14.31  
    -----  
    s^3 + 9.6 s^2 + 15.19 s + 19.14  
  
Continuous-time transfer function.
```

```
MF2 = feedback(G,C)
```

```
MF2 =  
  
         s^2 + 11.66 s + 19.33  
    -----  
    s^3 + 9.6 s^2 + 15.19 s + 19.14  
  
Continuous-time transfer function.
```

```
% zeros da malha fechada  
zero(MF1)
```

```
ans = 2x1  
    6.7311  
   -2.0000
```

```
zero(MF2)
```

```
ans = 2x1  
   -9.6631  
   -2.0000
```

Projeto de compensações

Essas compensações visam usar o mesmo controlador projetado acima. São dois os objetivos:

1. ajustar o ganho DC da malha fechada, de forma a assegurar o erro nulo no seguimento de sinais de referência que sejam constantes por partes;
2. mitigar o efeito do zero do processo que está localizado em -2 , o que é muito próximo dos polos dominantes especificados para a malha fechada.

```
% çõ compensaes :  
Cp = tf(2,[1 2]) % compensa o zero do processo. Note o ganho áunitrio do ...  
compensador.
```

```
Cp =
```

```
      2  
-----  
s + 2
```

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

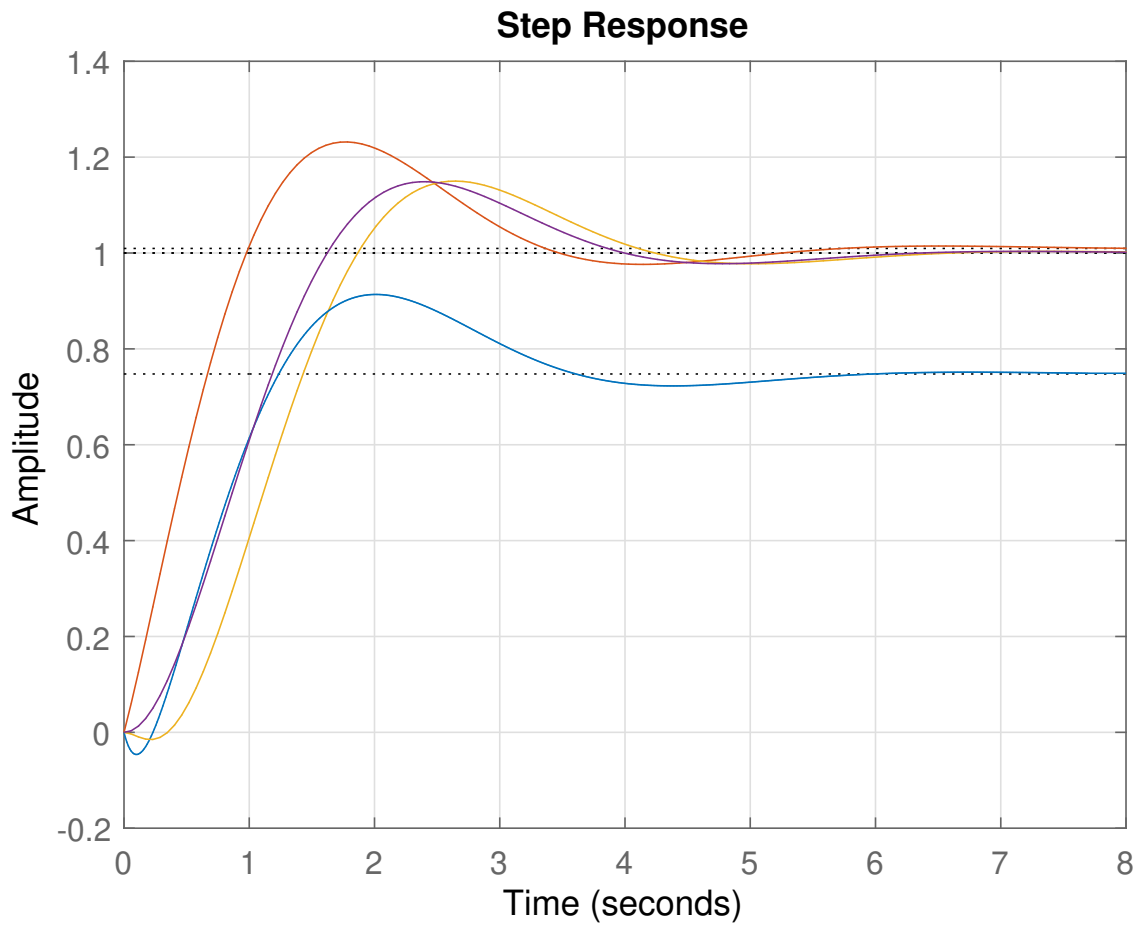
```
P1 = (dcgain(MF1)) ^ (-1) % compensa o ganho DC da malha 1
```

```
P1 = 1.3376
```

```
P2 = (dcgain(MF2)) ^ (-1) % compensa o ganho DC da malha 2
```

```
P2 = 0.9906
```

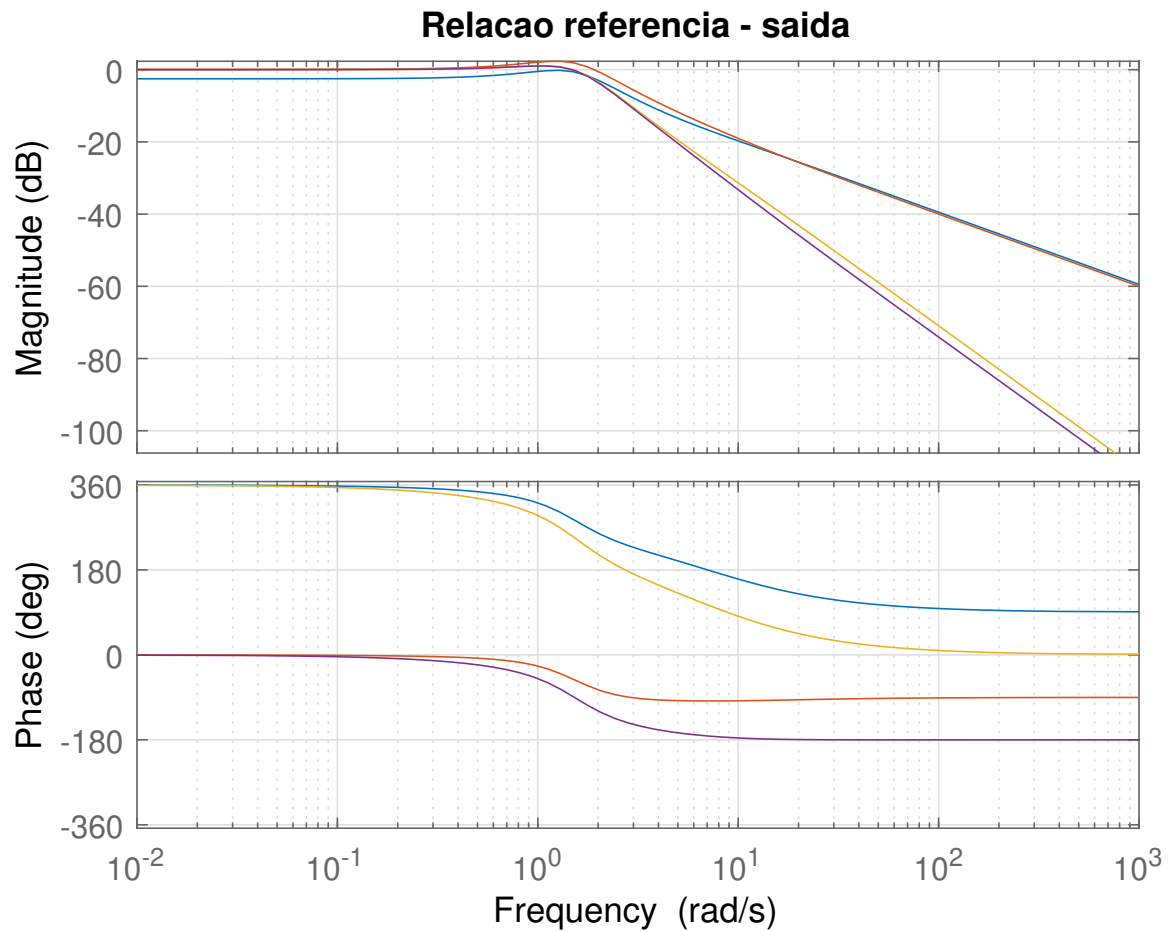
```
% respostas finais  
figure(1);  
step(MF1)  
hold on  
step(MF2)  
step(P1*Cp*MF1)  
step(P2*Cp*MF2)  
grid
```



Resposta em frequência da malha fechada

Relação entre a saída e a referência:

```
figure(2);
bode(MF1);
hold on
bode(MF2);
bode(P1*Cp*MF1)
bode(P2*Cp*MF2)
title('Relacao referencia - saida');
grid;
```



Relação entre o sinal de controle e a referência

```
MF1u = feedback(C,G);
figure(3);
bode(P1*Cp*MF1u);
hold on;
MF2u = feedback(1,series(C,G));
bode(P2*Cp*MF2u);
grid;
title('Relação entre o sinal de controle e a referência');
```

