Capítulo 1

Projeto de Controladores

1.1 Controlador por Tempo Mínimo

Seja o seguinte sistema que se deseja controlar:

$$G(s) = \frac{10}{(s+1)(s+10)}$$

Suponha que pede-se um controlador pelo tempo mínimo...

Solução: A idéia deste controlador é que o mesmo cancele todos os pólos e zeros estáveis da planta e são acrescentados tantos integradores quanto os necessários para zerar o erro em regime permanente para o tipo de entrada especificado.

Neste caso, primeiro temos que obter $BoG(z)$, fazendo no MATLAB:

```
>> clear
>> num=10;
>> den=conv([1 1],[1 10]);
>> printsys(num,den)

num/den =

                    10
    ---------------
    s^2 + 11 s + 10
>> G=tf(num,den)

Transfer function:
           10
---------------
s^2 + 11 s + 10

>> zpk(G)

Zero/pole/gain:
        10
------------
(s+10) (s+1)

>>
```

Imaginando que estamos amostrando o sistema à cada 0,1 segundos teremos:

```
>> T=0.1;
>> [numd,dend]=c2dm(num,den,T)
>> printsys(numd,dend,'z')
```

```
num/den =

              0.035501 z + 0.024654
       ------------------------
       z^2 - 1.2727 z + 0.33287
>> BoG=c2d(G,T)
>> zpk(BoG)

Zero/pole/gain:
 0.035501 (z+0.6945)
---------------------
(z-0.9048) (z-0.3679)

Sampling time: 0.1
>>
```

Note que nossa planta possui apenas 1 zero em $z = -0.6945$ e 2 pólos em $z = 0.9048$ e outro em $z = 0.3679$ (este último é o pólos mais rápido do sistema, está mais próximo do centro do círculo unitário em $\mathcal{Z}$.

Como queremos um controlador pelo tempo mínimo, nosso controlador deve ficar como:

$$C(Z) = K \frac{\Pi_{n=1}^{ZE} \left(z - z_{BoG(z)_n}\right)}{\Pi_{m=1}^{PE} \left(z - p_{BoG(z)_m}\right) \cdot (z-1)^w}$$

onde:

	K	= ganho do controlador;
	ZE	= quantidade de zeros estáveis da planta ($BoG(z)$);
	PE	= quantidade de pólos estáveis da planta;
	$z_{BoG(z)_n}$	= n-ésimo zero da planta;
	$p_{BoG(z)_m}$	= m-ésimo pólo da planta;
	$(z-1)$	= são integradores impostos pelo integrador no sistema em malha fechada;
	w	= número de integradores necessários para tornar nulo o erro em regime permanente – depende do tipo de entrada com a qual se está esperando operar a planta (degrau, rampa, parábola, etc).

Note que o controlador não cancela pólos ou zeros instáveis da planta. Se estes existirem, estes são desconsiderados na equação de $C(z)$.

Para o nosso caso então, $C(z)$ ficaria:

$$C(z) = \frac{K \cdot (z - 0.9048)(z - 0.3679)}{(z + 0.6945)(z-1)}$$

Note que fomos obrigados a introduzir um integrador no sistema (termo $(z-1)$) até para fazer com que o grau do denominador fosse maior ou igual ao do numerador senão estaríamos tentando projetar um controlador antecipativo (irrealizável na prática, por apareceriam termos como $y[k+1]$ na equação de diferenças do controlador). E o fator K se refere ao ganho que deve ser introduzido no sistema para garantir a resposta no tempo desejado – pode ser determinado traçando-se o lugar das raízes para o sistema em malha aberta: $FMTA(Z) = C(z)BoG(z)$.

Este controlador pode ser sintetizado no MATLAB fazendo-se:

```
>> polos_BoG=roots(dend)
polos_BoG =
            0.9048
            0.3679
>> zeros_BoG=roots(numd)
zeros_BoG =
       -0.6945
>> num_c=poly(polos_BoG);
>> den_c_aux=poly(zeros_BoG)
den_c_aux =
```

```
          1.0000    0.6945
>> den_c=conv(den_c_aux,[1 -1]);
>> C=tf(num_c,den_c,T);
>> zpk(C)

Zero/pole/gain:
(z-0.9048) (z-0.3679)
---------------------
    (z-1) (z+0.6945)

Sampling time: 0.1
```

Verificando como fica este sistema em malha-fechada, fazemos:

```
>> [num_ftma,den_ftma]=series(num_c,den_c,numd,dend);
>> FTMA=series(C,BoG);
>> zpk(FTMA)

Zero/pole/gain:
0.035501 (z-0.9048) (z-0.3679) (z+0.6945)
-----------------------------------------
 (z-1) (z-0.9048) (z-0.3679) (z+0.6945)

Sampling time: 0.1
```

Percebemos pela equação acima que $FTMA(z) = 0.035501/(z-1)$.

Note que antes de fecharmos a malha, necessitamos estabelecer o ganho K do nosso controlador. Para tanto traçamos o lugar das raízes para o nosso sistema em malha aberta:

```
>> rlocus(FTMA)
>> zgrid(1,1)
>> axis equal
```

Que resulta no seguinte gráfico do lugar das raízes:

Note que os pólos de malha fechada do sistema se tornaram pólos reais simples (andam apenas sobre a parte $\mathbb{R}\Game\lessdot$ do eixo $\mathcal{Z}$ – linhas verde e vermelha da figura. Analisando o gráfico obtido, concluí-se que podemos localizar um pólo de malha fechada justamente na origem do plano $\mathcal{Z}$. Para tanto, temos que fazer:

```
>> hold on
>> [K,polos_mf]=rlocfind(FTMA)
Select a point in the graphics window
selected_point =
    -0.0039 - 0.0039i
K =
     28.2794
polos_mf =
          0.9048
     -0.6945
          0.3679
     -0.0039
>>
```

Acompanhe pelo gráfico à seguir:

Adotando o ganho de $K = 28.2794$, fechando a malha vamos obter:

```
>> FTMF=feedback(K*FTMA,1,-1);
>> [num_mf,den_mf]=tfdata(FTMF,'v');
>> figure;dstep(num_mf,den_mf)
```

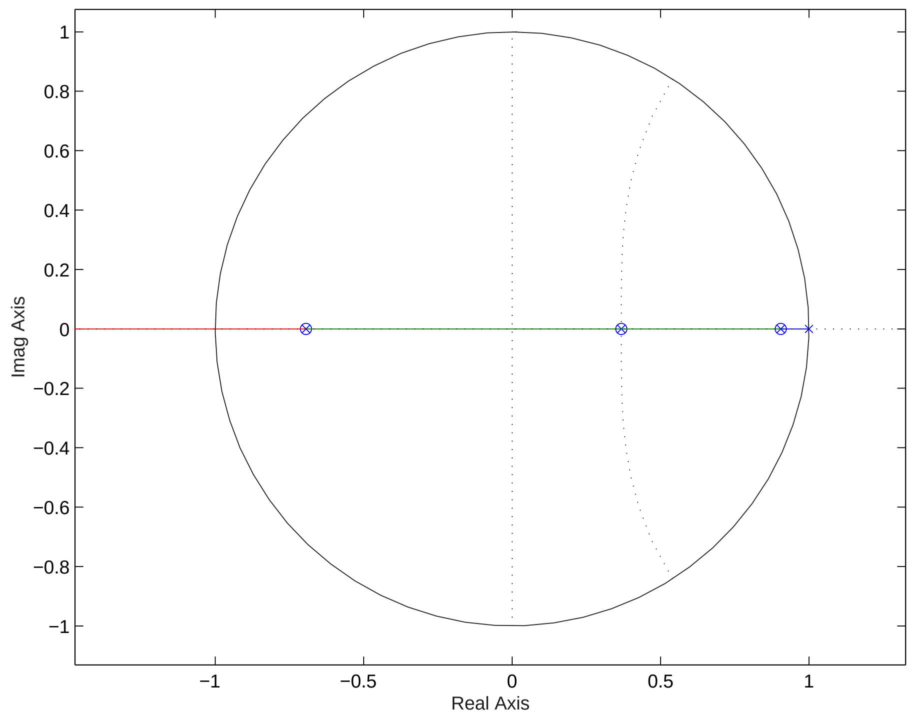

Figura 1.1: Lugar das raízes para $FTMA(z)$.

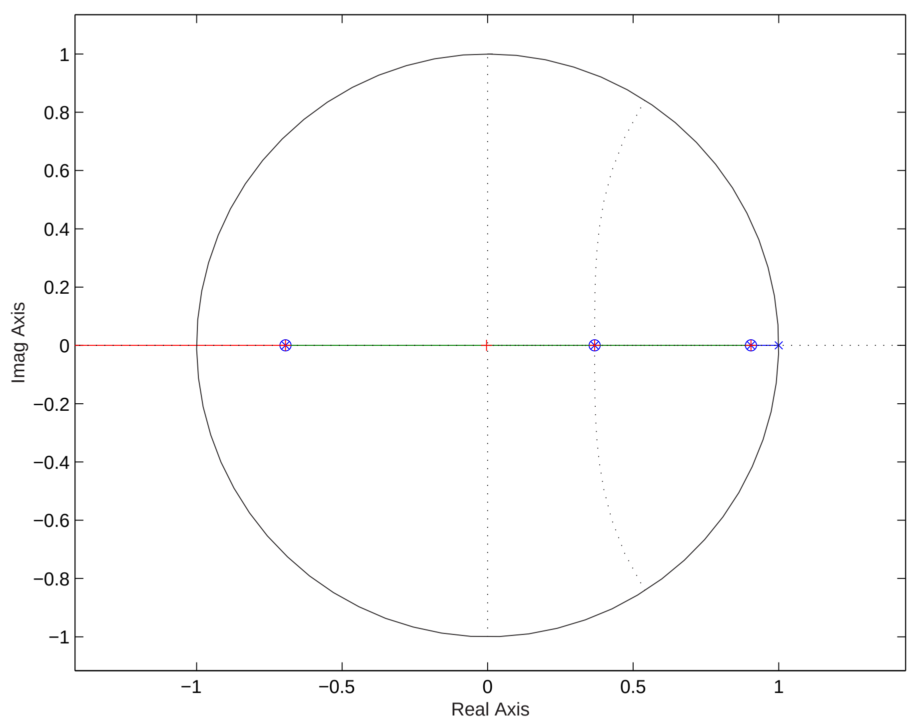

Figura 1.2: Lugar das raízes para $FTMA(z)$ – Descobrindo K de $C(z)$.

Que gera a seguinte figura:

Note que este sistema estabilizou em $y(\infty) \cong 1$. O valor exato podemos descobrir no MATLAB aplicando o teorema do valor final em $Y(z) = FTMF(z) \cdot R(Z)$, para quando $R(z) = U(z)$ (referência é uma entrada degrau), e então temos:

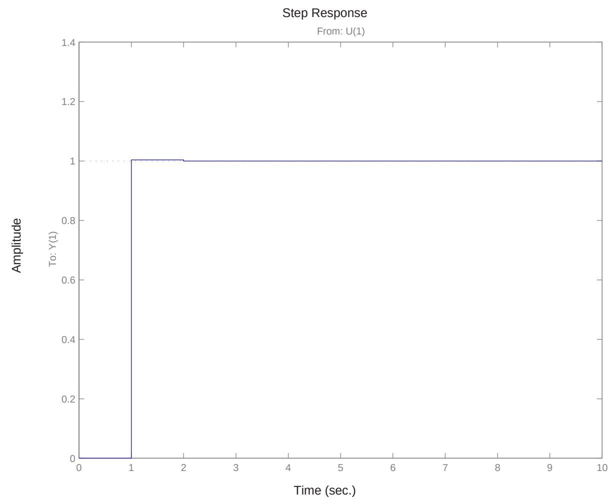

Figura 1.3: Lugar das raízes para $FTMA(z)$ – Descobrindo K de $C(z)$.

$$y(\infty) = \lim_{k\to\infty} FTMF(z) \cdot R(z) = \lim_{z\to 1}(1 - z^{-1}) \cdot FTMF(z) \cdot \frac{1}{(1 - z^{-1})}$$

No MATLAB:

```
>> ks=polyval(num_mf,1)/polyval(den_mf,1)
ks =
          1
>>
```

Confirmamos que $y(\infty) = 1$. Se fosse desejado que a saída estabilizasse em $y = 5$, teríamos que introduzir uma entrada degrau de amplitude igual à 5 na entrada do sistema. No MATLAB ficaria:

```
>> figure;dstep(5*num_mf,den_mf)
>> print -depsc2 -f3 dstep5.eps
```

Com a conseqüente figura gerada:

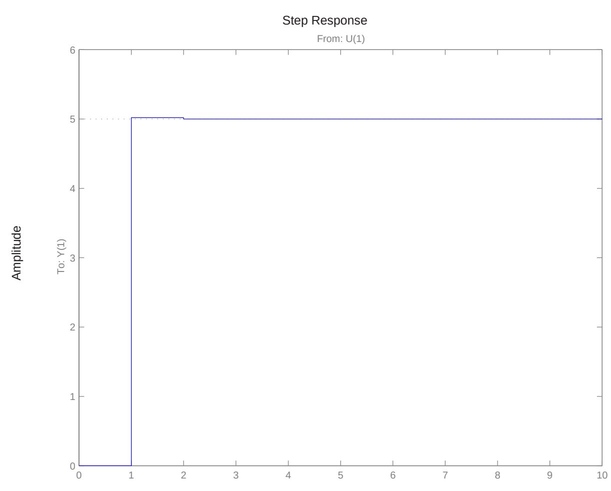

Figura 1.4: Resposta do Sistema em malha fechada para degrau de amplitude 5.