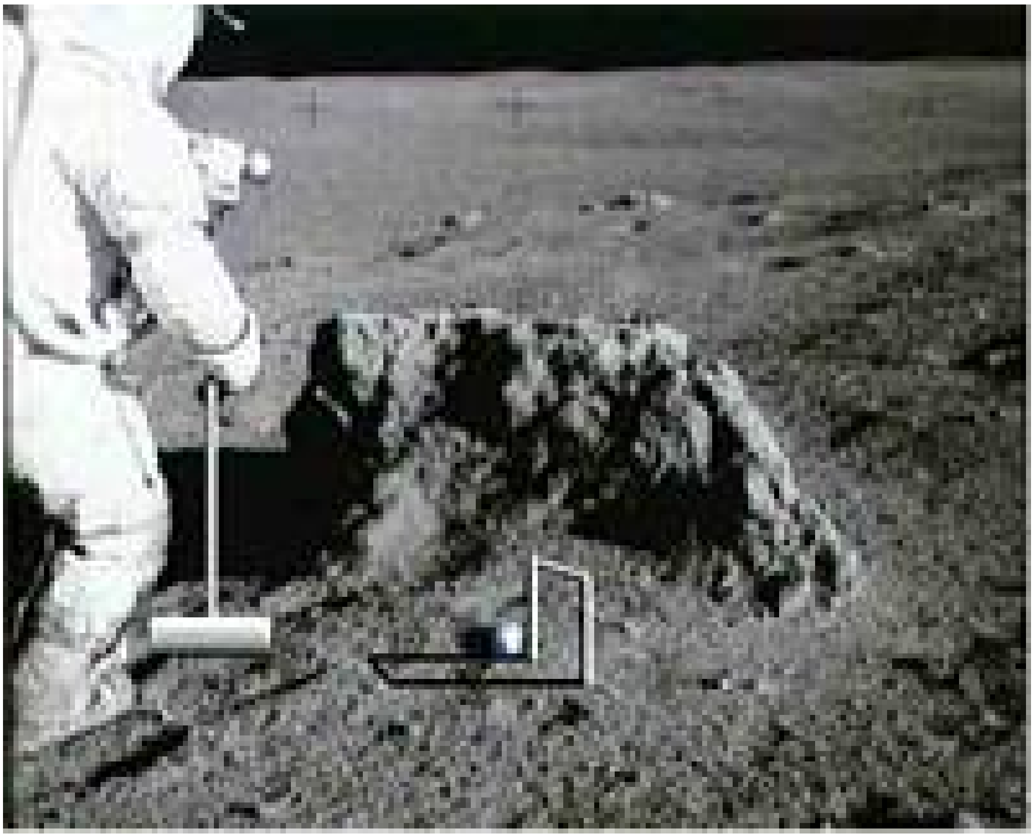
Identificação de um Sistema
Método Frequencial

Prof. Fernando Passold
E-mail: fpassold@upf.br

2 de junho de 2011



Universidade de Passo Fundo
Curso de Engenharia Elétrica
Disciplina de Lab. de Controle Automático II

Atenção:
Este material ainda está sendo editado (esta não é a versão final).

Falta revisar este material!

Aceita-se observações quanto a correções que se façam necessárias!

Sumário

1 Respostas ao Degrau

1.1 Sistema de 1a-ordem com atraso de transporte

Seja um sistema do tipo:

$$G(s) = \frac{K \cdot e^{-t_p s}}{(s + p_1)} \tag{1}$$

A resposta (em malha aberta) para um degrau de amplitude A injetado na entrada deste sistema pode ser calculado como:

Degrau de entrada:

$$r(t) = u(t) = \begin{cases} 0, & t < 0 \\ A, & t > 0 \end{cases}$$

Sua transformada de Laplace resulta em:

$$\mathcal{L}\{u(t)\} = A \cdot \frac{1}{s}$$

Agrupando os termos temos:

$$Y(s) = R(s) \cdot G(s) \tag{2}$$

$$Y(s) = \frac{K \cdot A \cdot e^{-t_p s}}{s(s + p_1)} \tag{3}$$

Realizando a transformada inversa de (3) obtemos:

$$y(t) = \mathcal{L}^{-1}\{Y(s)\} = \left(\mathcal{L}^{-1}\{W(s)\}\right) \cdot u(t - t_p)$$

onde:

$$W(s) = \frac{K\,A}{s(s + p_1)} = \frac{k_1}{s} + \frac{k_2}{s + p_1}$$

$$K\,A = k_1(s + p_1) + k_s\,s$$

$$K\,A = s(k_1 + k_2) + k_1\,p_1$$

$$\begin{array}{ll} K\,A = k_1\,p_1 & \quad 0 = k_1 + k_2 \\ k_1 = \frac{K\,A}{p_1} & \quad k_2 = -\frac{K\,A}{p_1} \end{array}$$

Lembrando da Tabela de Transformadas de Laplace:

$$\mathcal{L}^{-1}\left\{\frac{k_1}{s}\right\} = k_1\,u(t)$$

$$\mathcal{L}^{-1}\left\{\frac{1}{s - a}\right\} = e^{at}$$

$$\mathcal{L}^{-1}\left\{\frac{k_2}{(s + p_1)}\right\} = k_2\,e^{-p_1\,t}$$

Teremos então:

$$y(t) = \frac{KA}{p_1}\left(1 - e^{-p_1 t}\right) \cdot u(t - t_p)$$

A parte esquerda da equação anterior (desconsiderando o atraso no tempo) resulta no gráfico 1, cujo código no MATLAB adotado para gerá-lo aparece à seguir:

```
>> K=5; % ganho estático da planta
>> A=2; % amplitude do degrau
>> p1=2; % posição do pólo
>> fplot(@(t) [(K*A*(1-exp(-p1*t)))/p1],[0 5])
>> grid
```

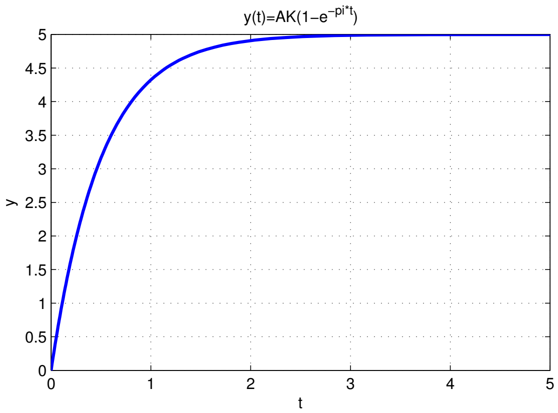


Figura 1: Resposta temporal de sistema de 1a-ordem.

1.2 Sistema de 2a-ordem

Um sistema de 2a-ordem pode ser parametrizado como:

$$G(s) = \frac{k_2 \cdot b}{s^2 + a\, s + b} = \frac{k_2 \cdot w_n^2}{s^2 + 2\,\zeta\, w_n\, s + w_n^2} \tag{4}$$

onde:

$$\begin{aligned} s &= \sigma \pm j\, w_d \\ \sigma &= w_n \cos(\alpha) = w_n\, \zeta \\ w_d &= w_n \sin(\alpha) = w_n \sqrt{1 - \zeta^2} \\ a\zeta &= \cos(\alpha) \\ \sin(\alpha) &= \sqrt{1 - \zeta^2} \end{aligned}$$

1.3 Sistema de 3a-ordem (somente pólos)

Seja uma planta caracterizada como:

$$G(s) = \frac{K}{(s + p_1)(s + p_2)(s + p_3)} \tag{5}$$

Supondo que todos os seus pólos são estáveis (provavelmente um par conjugado complexo e mais uma raiz real). O acréscimo de uma entrada degrau de amplitude A a este sistema resulta na seguinte equação característica:

$$\begin{aligned} C(s) &= R(s)G(s) \\ &= \frac{A\,K}{s(s + p_1)(s + p_2)(s + p_3)} \end{aligned}$$

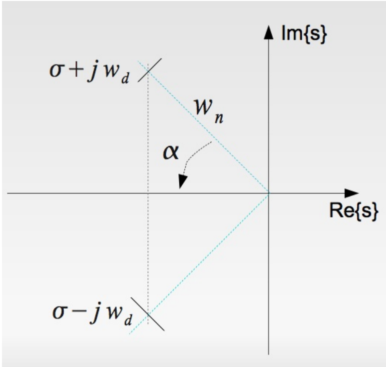


Figura 2: Localização dos pólos complexos num sistema de 2a-ordem.

E a transformada inversa de Laplace (saída do sistema em malha aberta) resulta em:

$$y(t) = \frac{A\,K}{p_1\,p_2\,p_3} - \frac{A\,K}{p_1\,\mathrm{e}^{p_1\,t}\,(p_1 - p_2)\,(p_1 - p_3)} + \frac{A\,K}{p_2\,\mathrm{e}^{p_2\,t}\,(p_1 - p_2)\,(p_2 - p_3)} - \frac{A\,K}{p_3\,\mathrm{e}^{p_3\,t}\,(p_1 - p_3)\,(p_2 - p_3)} \tag{6}$$

Usando MATLAB:

```
>> clear all
>> syms s A K p_1 p_2 p_3;
>> c = (A*K)/(s*(s+p_1)*(s+p_2)*(s+p_3));
>> y = ilaplace(c)
```

Exemplo: Supondo que um sistema possua os pólos localizados em: $p_1 = -1 + 2j$, $p_2 = -1 - 2j$ e $p_3 = -5$, e que o ganho estático da planta seja: $K = 0,5$:

$$Y = \frac{1}{2(s^2 + 2s + 5)(s + 5)} \tag{7}$$

Suponha ainda que este sistema seja submetido a uma entrada degrau de amplitude $A = 2$, a resposta no tempo (malha aberta) resultaria em algo do tipo mostrado na figura 3. A figura 4(b) mostra o lugar das raízes que corresponde a este exemplo.

A equação transferência resultaria em:

$$\begin{aligned} Y(s) &= R(s)G(s) = \frac{A\,K}{s(s + p_1)(s + p_2)(s + p_3)} \\ &= \frac{1}{s(s + 1 + 2j)(s + 1 - 2j)(s + 5)} \\ &= \frac{1}{s(s^2 + 2s + 5)(s + 5)} \\ &= \frac{1}{s(s^3 + 7s^2 + 15s + 25)} \\ &= \frac{1}{s^4 + 7s^3 + 15s^2 + 25s} \end{aligned}$$

E sua transformada inversa de Laplace resulta em:

$$y(t) = \frac{1}{25} - \frac{3\left(\cos(2\,t) + \frac{4\,\sin(2\,t)}{3}\right)}{100\,\mathrm{e}^{t}} - \frac{1}{100\,\mathrm{e}^{5\,t}}$$

$$y(t) = \frac{1}{25} - \frac{3}{100}\left[cos(2t) + \frac{4}{3}sin(2t)\right]\,e^{-t} - \frac{e^{-5t}}{100} \tag{8}$$

O termo $cos(2t) + \frac{4}{3}sin(2t)$ corresponde a uma senóide com atraso de fase (ver Apêndice B, pág. 22):

$$cos(2t) + \frac{4}{3}sin(2t) = \sqrt{\left[1^2 + \left(\frac{4}{3}\right)^2\right]}\ \sin(2t + \varphi) = 1,67\ \sin(2t + 36,87^o)$$

onde:

$$\varphi = \arctan\left(\frac{1}{4/3}\right) = \arctan(3/4) = 0,6435(\text{rad}) = 36,87^o$$

ou finalmente:

$$y(t) = \frac{1}{25} - \frac{1}{20}\,e^{-t}\,\sin\left(2t + 36,87^o\right) - \frac{e^{-5t}}{100}$$

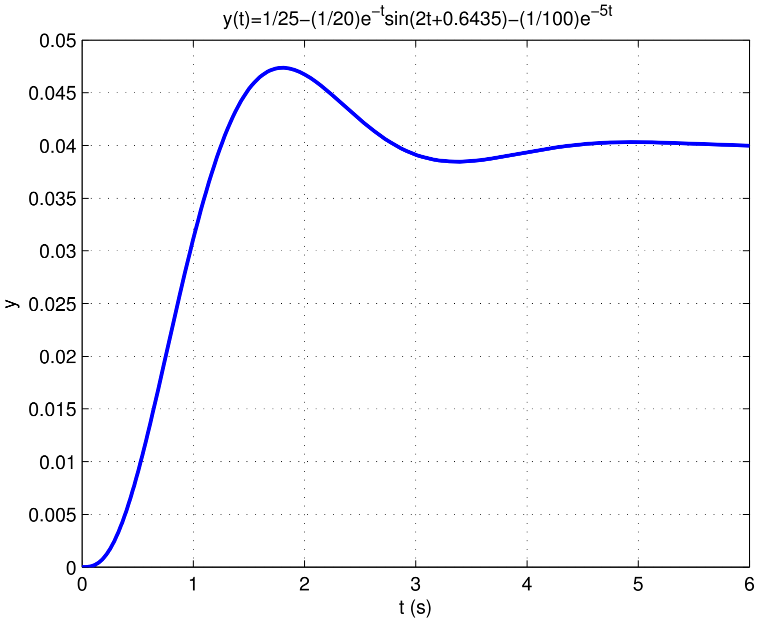


Figura 3: Resposta à entrada degrau de um sistema de 3a-ordem (somente pólos).

A mesma resposta alcançada de maneira literal, pode ser obtida usando-se as funções '**tf**(·)' e '**step**(·)' do MATLAB (ver figura 4(a)):

```
>> num=0.5;
>> den=[1 7 15 25];
>> c=tf(num,den)
 Transfer function:
              0.5
----------------------
s^3 + 7 s^2 + 15 s + 25
>> step(2*c)
>> figure; pzplot(c)
```

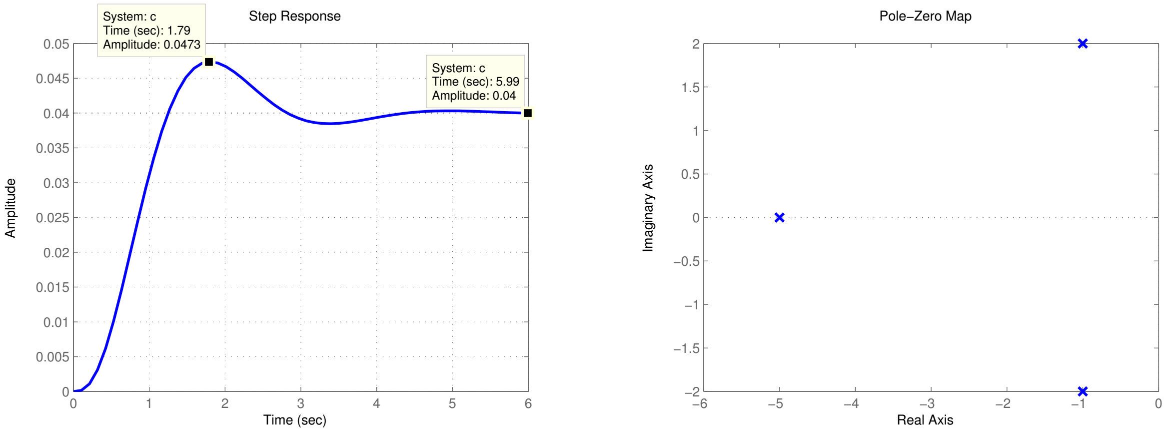
Se uma malha de controle com ganho proporcional for incorporada a este sistema, eq(7), a função transferência de malha fechada fica:

$$\text{FTMF(s)} = \frac{1}{2}\cdot\frac{1}{(s+5,025)(s^2+1,975s+5,075)}$$

Aplicando-se uma entrada degrau de amplitude 2, obteríamos a saída mostrada na figura 5.

Código adotado no MATLAB:

```
>> clear
>> num=0.5;
>> den=poly([-1+2j -1-2j -5])
>> g=tf(num,den)
 Transfer function:
             0.5
----------------------
s^3 + 7 s^2 + 15 s + 25
>> zpk(g)
Zero/pole/gain:
          0.5
```



(a) Usando função '**step**(·)' do MATLAB.

(b) Lugar das raízes.

Figura 4: Sistema com 3 pólos somente.

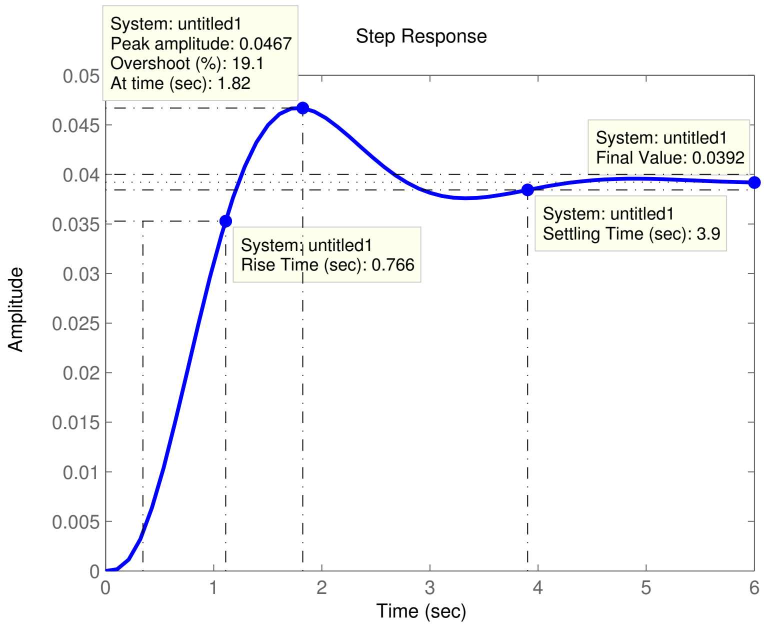


Figura 5: Sistema com 3 pólos somente em malha fechada com ganho proporcional unitário.

```
------------------------
(s+5) (s^2 + 2s + 5)

>> h=tf([1],[1]); % controlador proporcional, ganho unitário
>> FTMF=feedback(g,h)
Transfer function:
          0.5
------------------------
s^3 + 7 s^2 + 15 s + 25.5
>> zpk(FTMF)
Zero/pole/gain:
             0.5
--------------------------------
(s+5.025) (s^2 + 1.975s + 5.075)
>> step(2*FTMF)
>>
```

1.4 Sistema de 3a-ordem com 1 zero

Seja uma planta semelhante à do caso anterior, contendo 3 pólos em: $s = -5$ e $s = -1 \pm j2$ mas com um zero cuja localização varia conforme os casos abaixo:

a) Zero em $s = -1/2$;

b) Zero em $s = -3$;

c) Zero em $s = -10$.

A figura 6 mostra o plano-s com a localização das raízes para os 3 casos.

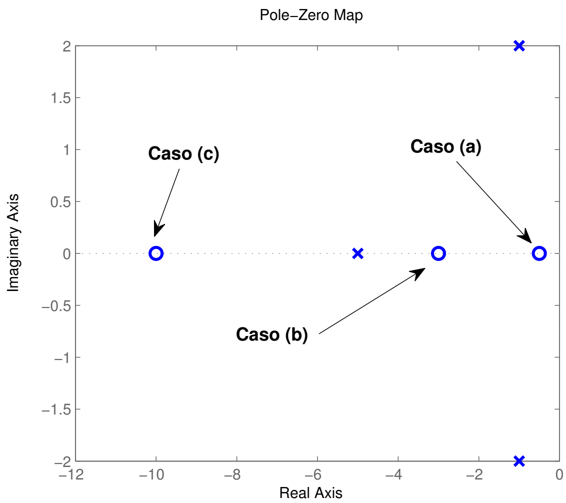


Figura 6: Os 3 casos do sistema de 3a-ordem com um zero.

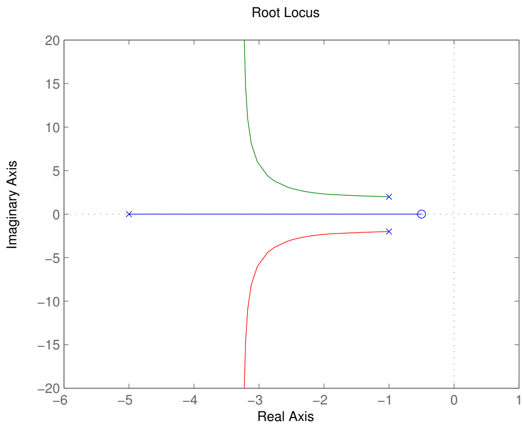
1.4.1 Caso a) Zero em $s = -1/2$

A função transferência deste sistema fica:

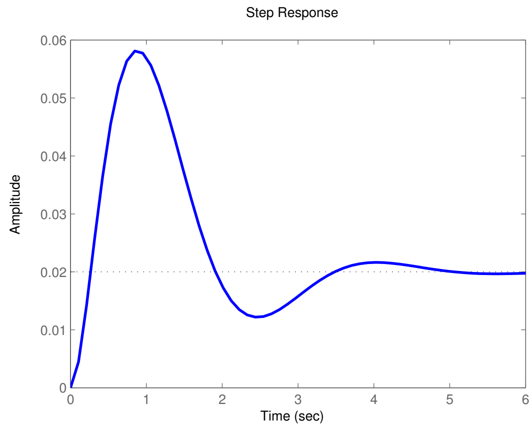
$$G(s) = \frac{0.5(s + 1/2)}{s^3 + 7s^2 + 15s + 25}$$

A posição dos pólos e zero deste sistema, justamente com o diagrama do local das raízes (resultando deste sistema quando submetido a um controlador proporcional) é mostrado na figura 7(a). Quando aplicado um degrau de amplitude 2 (em malha aberta) neste sistema, resulta no gráfico mostrado na figura 7(b).

Note que este sistema não se torna estável para malha fechada com ganho proporcional (figura 7(a)).



(a) Lugar das raízes.



(b) Resposta ao degrau (malha aberta).

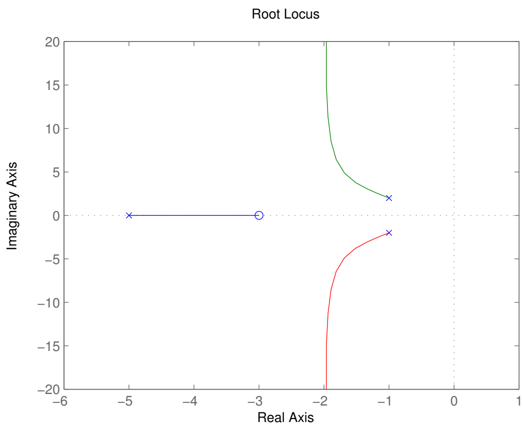
Figura 7: Caso a) Sistema com 3 pólos e 1 zero em $s = -1/2$.

1.4.2 Caso b) Zero em $s = -3$

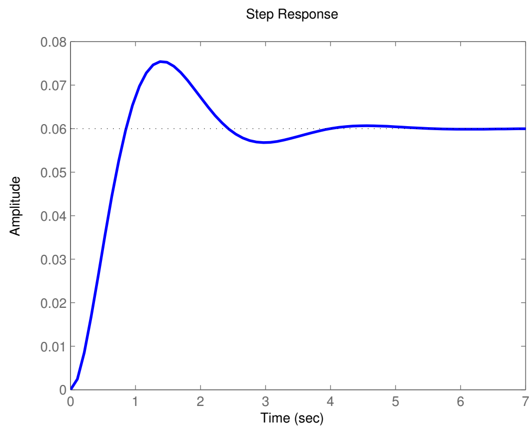
A função transferência deste sistema fica:

$$G(s) = \frac{0.5(s+5)}{s^3 + 7s^2 + 15s + 25} = \frac{1/2s + 3/2}{s^3 + 7s^2 + 15s + 25}$$

A posição dos pólos e zero deste sistema, justamente com o diagrama do local das raízes (resultando deste sistema quando submetido a um controlador proporcional) é mostrado na figura 8(a). Quando aplicado um degrau de amplitude 2 (em malha aberta) neste sistema, resulta no gráfico mostrado na figura 8(b).



(a) Lugar das raízes.



(b) Resposta ao degrau (malha aberta).

Figura 8: Caso a) Sistema com 3 pólos e 1 zero em $s = -3$.

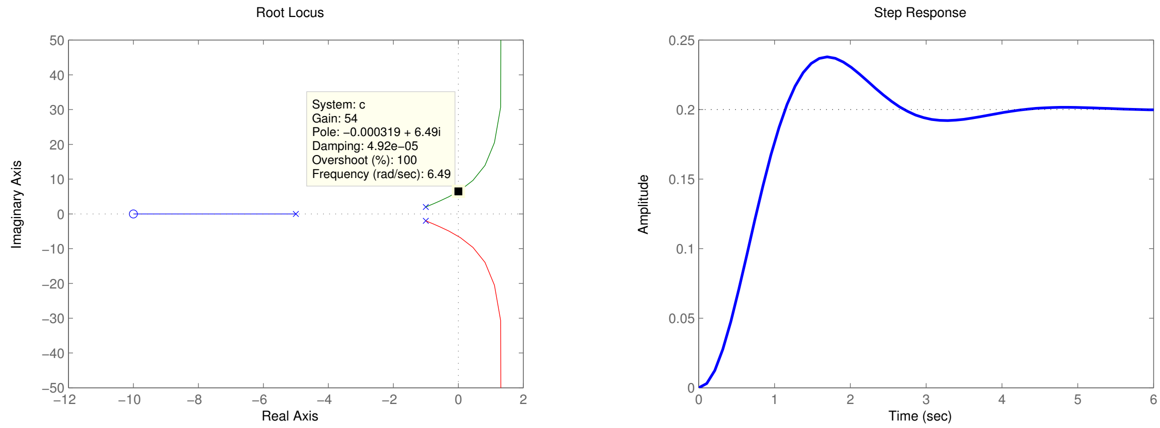
Notar que este sistema também não se torna instável mesmo para ganhos elevados (K_p) quando em malha fechada (ver figura 8(a)).

1.4.3 Caso c) Zero em $s = -10$

A função transferência deste sistema seria:

$$G(s) = \frac{0.5(s+10)}{s^3 + 7s^2 + 15s + 25} = \frac{1/2s + 5}{s^3 + 7s^2 + 15s + 25} \tag{9}$$

A posição dos pólos e zero deste sistema, justamente com o diagrama do local das raízes (resultando deste sistema quando submetido a um controlador proporcional) é mostrado na figura 9(a). Quando aplicado um degrau de amplitude 2 (em malha aberta) neste sistema, resulta no gráfico mostrado na figura 9(b).



(a) Lugar das raízes.

(b) Resposta ao degrau (malha aberta).

Figura 9: Caso a) Sistema com 3 pólos e 1 zero em $s = -10$.

Neste caso, observando-se a figura 9(a) se percebe que a partir de certo valor, a malha fechada usando apenas ganho proporcional se torna instável, o caso do processo da bola-e-tubo.

Aplicando uma entrada degrau de amplitude 5 a este sistema, resulta em:

$$Y(s) = R(s) \cdot G(s) = \frac{5 \cdot 0.5(s+10)}{s(s^2+1 \pm j2)(s+5)} = \frac{2.5s+25}{s^4+7s^3+15s^2+25s} \tag{10}$$

A transformada inversa de Laplace sobre a equação 10 resulta em:

$$y(t) = 1 - \frac{7\left(\cos(2\,t) + \frac{6\,\sin(2\,t)}{7}\right)}{8\,e^{t}} - \frac{1}{8\,e^{5\,t}} \tag{11}$$

$$y(t) = 1 - \frac{7}{8}e^{-t}\left[\cos(2\,t) + \frac{6}{7}\sin(2\,t)\right] - \frac{1}{8}e^{-5\,t} \tag{12}$$

Note que em comparação com um sistema de 3a-ordem constituído apenas por 3 pólos, o que mudou agora, foi o valor atingido em regime permamente (efeito de "offset" no ganho DC provocado pelo zero introduzido no sistema).

1.5 Sistema de 5a-ordem

Seja um sistema de 5a-ordem com os pólos localizados em:

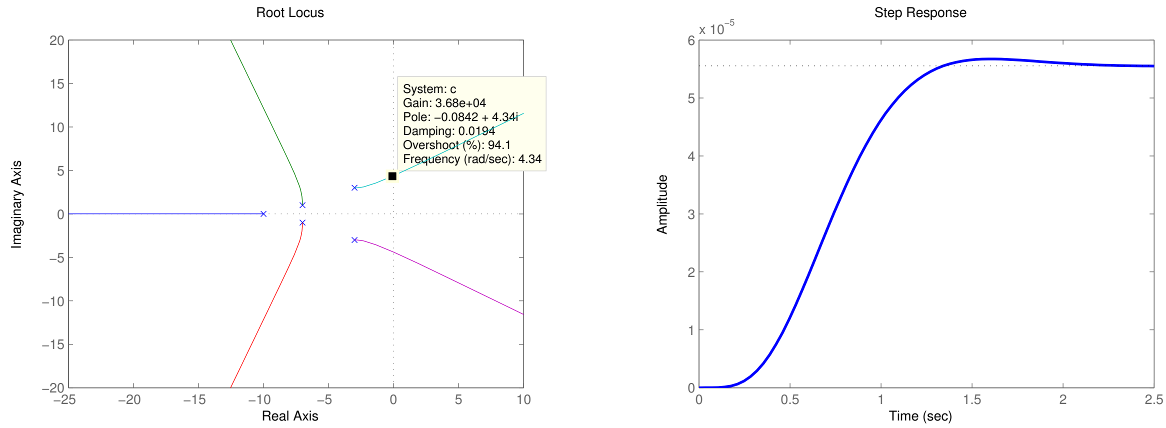
$$\begin{aligned} p_1 &= -3 + j3 \\ p_2 &= -3 - j3 \\ p_3 &= -7 + j \\ p_4 &= -7 + j \\ p_5 &= -10 \end{aligned}$$

Considerando ainda um ganho estático de 1/2, a função transferência deste sistema fica:

$$\begin{aligned} G(s) &= \frac{1}{2\left[(s+10)(s^2+14s+50)(s^2+6s+18)\right]} \\ G(s) &= \frac{1}{2\left(s^5+30s^4+352s^3+2072s^2+6420s+9000\right)} \end{aligned}$$

Quando submetido a uma entrada degrau de amplitude 2 (em malha aberta), resulta na resposta ilustrada na figura 10(b).

Note que os pólos dominantes neste sistema são: $p_{\{1,2\}} = -3 \pm j3$ (resulta na equação: $(s^2 + 6s + 18)$). Se aplicarmos uma entrada degrau de amplitude 2 neste sistema de 2a-ordem (malha aberta), obtemos uma resposta bastante próxima a não ser pelas amplitudes envolvidas como pode ser visto na figura.



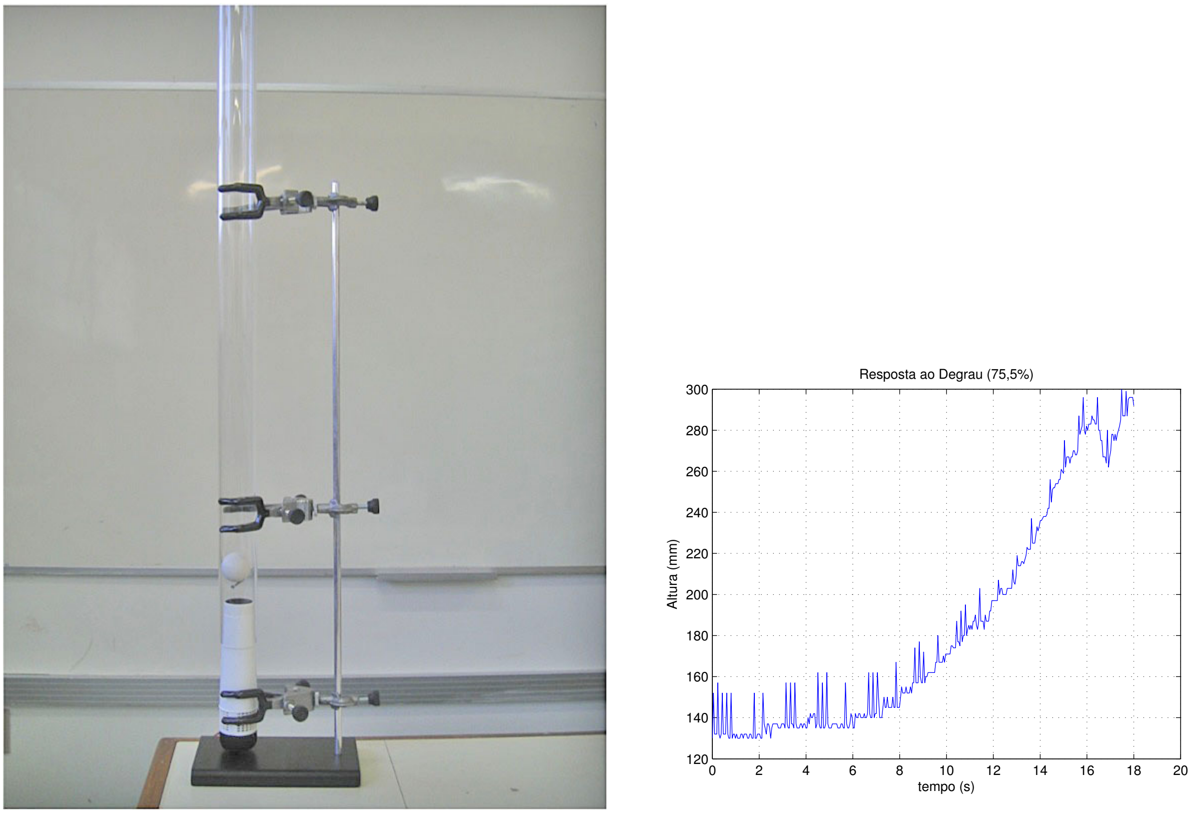
(a) Lugar das raízes.

(b) Resposta ao degrau (malha aberta).

Figura 10: Sistema com 5 pólos (somente).

2 Um exemplo real

Seja um exemplo real do processo da Bola-e-Tubo (figura 11(a)). Quando aplicado uma entrada degrau de 75,5% de amplitude (saída do controlador PWM), ele reage da forma mostrada na figura 11(b).



(a) Foto do processo.

(b) Resposta ao Degrau.

Figura 11: Processo da Bola e Tubo e sua resposta ao degrau.

Uma saída filtrada do processo pode ser obtida, se for adotado um filtro passa-baixa (PB) sobre a resposta original do sistema (ver Apêndice D, pág. 22).

Aplicando-se o filtro PB sobre o sinal ruidoso do processo da Bola-e-Tubo com freqüência de corte de $f_c = 0,6$ [Hz] obtemos o resultado apresentado na figura 12.

```
>> RC=1/(2*pi*0.6)
RC =
      0.2653
```

```
>> y=lowpass(dados(:,1),Ts,RC);
alpha =
       0.1505
RC/Ts= 5.6438
>> plot(t,dados(:,1),'m:', t,y,'b-')
```

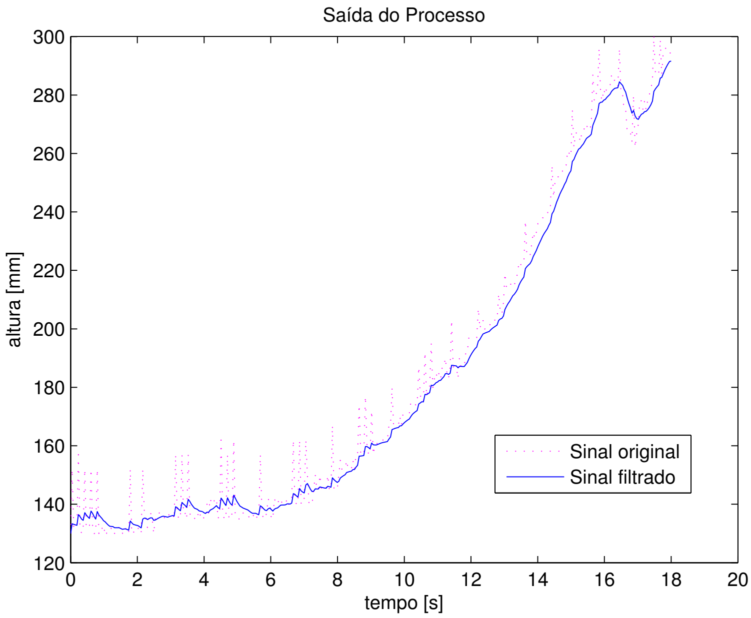


Figura 12: Sinal de saída filtrado do processo da Bola-e-Tubo.

2.1 Indentificando como de 2a-ordem, subamortecido

Seja um sistema de 2a-ordem parametrizado como:

$$G(s) = K \cdot \frac{w_n^2}{s^2 + 2\zeta w_n s + w_n^2} \tag{13}$$

onde: w_n corresponde a frequencia de oscilação natural do sistema; ζ corresponde ao fator de amortecimento, neste caso específico, de um sistema subamortecido: $0 < \zeta < 1$; e K corresponde ao valor atingido em regime permanente (malha aberta, entrada degrau).

Quando submetido à uma entrada degrau, de amplitude K_u o sistema fica:

$$C(s) = K_u \cdot K \cdot \frac{w_n^2}{s\left(s^2 + 2\zeta w_n s + w_n^2\right)} = \frac{K_1}{s} + \frac{K_2 s + K_3}{s^2 + 2\zeta w_n s + w_n^2} \tag{14}$$

A transformada de Laplace inversa da equação 14 resulta em:

$$c(t) = K\,K_u\left\{1 - \exp\left(-\zeta\,w_n\,t\right) \cdot \left[\cos\left(w_n\sqrt{1-\zeta^2}\,t\right) + \frac{\zeta}{\sqrt{1-\zeta^2}}\sin\left(w_n\,\sqrt{1-\zeta^2}\,t\right)\right]\right\} \tag{15}$$

onde: $w_n\sqrt{1-\zeta^2} = w_d$ e corresponde a frequencia da oscilação amortecida. Notar que corresponde também a parte imaginária do par conjugado complexo: $\pm j\,w_d = \pm j\,w_n\sqrt{1-\zeta^2}$ e que $-\zeta\,w_n = -\sigma_d$ corresponde a parte real do par conjugado complexo. Desta forma, podemos reescrever a eq. (15) no formato:

$$c(t) = K\,K_u\,\left\{1 - \exp(-\sigma\,t) \cdot [\cos(w_d\,t) + b\sin(w_d\,t))\right\} \tag{16}$$

onde: $b = \frac{\zeta}{\sqrt{1-\zeta^2}}$.

A função **step2poles.m** sintetiza no MATLAB a resposta temporal descrita pela eq. (15).

```
function y = step2poles(p,t)
% esta função retorna um vetor de valor de tamanho size(t) que
% corresponde a resposta temporal de um sistema de 2a-ordem subamortecido
% com par de polos complexos, e 0 < fator_amortecimento < 1
% y=K*{1-exp(-zet*wn*t)*[cos(wn*sqrt(1-zeta^2)*t)+(zeta/sqrt(1-zeta^2))*sin(wn*sqrt(1-
    zeta^2)*t)]}
% K --> p(1)
% zeta --> p(2)
% wn --> p(3)

K=p(1);
zeta=p(2);
wn=p(3);
sigma=zeta*wn;           % parte real do par complexo conjugado
wd=wn*sqrt(1-zeta*zeta); % parte imaginária do par complexo conjugado
b=zeta/(sqrt(1-zeta*zeta));
y= K*(1 - exp(-sigma.*t).*( cos(wd.*t)+b*sin(wd.*t) ) );
```

step2poles.m

Testando a função criada anteriormente para o sistema:

$$G(s) = \frac{9}{s^2+2s+9} = \frac{b}{s^2+as+b}$$

com pólos localizados em $s = -1.0000 \pm j\sqrt{8}$. Temos neste caso: $b = w_n^2$, $w_n = \sqrt{9} = 3$ [rad/s], $a = 2\zeta w_n$, $\zeta = \frac{a/2}{w_n} = \frac{2/2}{3} = 1/3 = 0.3333$. Assim:

```
>> wn=3; p(3)=wn;
>> zeta=1/3; p(2)=zeta;
>> K=1; p(1)=K;
>> t=1:0.01:5;
>> y=step2poles(p,t);
>> % montando função transferência no MATLAB:
>> num=9;
>> g2=tf(num,den)
 Transfer function:
        9
-------------------
 s^2 + 2 s + 9
>> y2=step(g2,t);
>> figure; plot(t,y2,'m+', t,y,'b-')
```

O gráfico resultante aparece na figura 13.

Pode-se usar o método dos mínimos quadrados para identificar os parâmetros K, ζ e w_n do processo da bola-e-tubo, trabalhando sobre o sinal de saída filtrado (figura 12). Neste caso usamos a função pronta do MATLAB: 'lsqcurvefit($\cdot$)'. Protótipo da função lsqcurvefit($\cdot$):

x = lsqcurvefit (FUN, x_0, x_{DATA}, y_{DATA}) inicia com a estimativa x_0 (de parâmetros de entrada) para encontrar os coeficientes (parâmetros) x que melhor se ajustem a função não linear FUN, dado o conjunto de pontos (dados): $[x_{DATA}, y_{DATA}]$, usando método dos mínimos quadrados.

onde FUN pode ser especificada usando '@':

```
xdata = [5;4;6];          % example xdata
ydata = 3*sin([5;4;6])+6; % example ydata
x = lsqcurvefit(@myfun, [2 7], xdata, ydata)
```

Neste caso 'mfun($\cdot$)' é uma função do MATLAB especificada como:

```
function F = myfun(x,xdata)
  F = x(1)*sin(xdata)+x(2);
```

Outro exemplo de função usando parâmetros fixos (c) e outros que dependem do problema ($x(i)$) pode ser adotada:

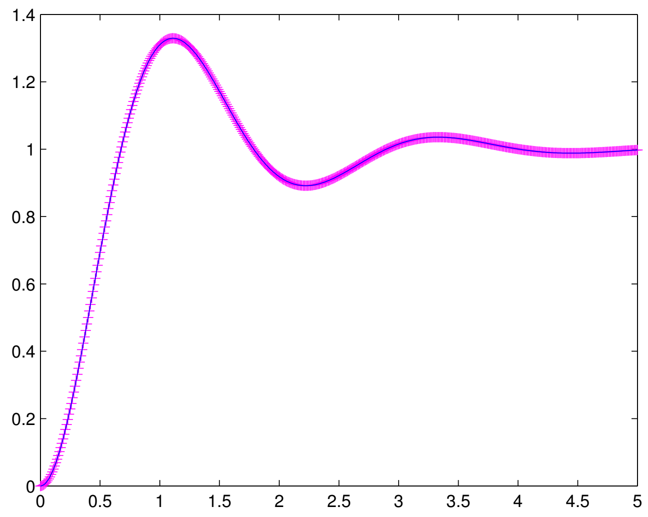


Figura 13: Teste da função step2poles.m.

```
function F = myfun(x,xdata,c)
  F = x(1)*exp(c*xdata)+x(2);
```

Neste último caso, antes desta função ser chamada é necessário já estar atribuído um valor para o parâmetro c. O código abaixo mostra um exemplo de uso:

```
xdata = [3; 1; 4];              % example xdata
ydata = 6*exp(-1.5*xdata)+3;  % example ydata
c = -1.5;                       % define parameter
x = lsqcurvefit(@(x,xdata) myfun(x,xdata,c),[5;1],xdata,ydata)
```

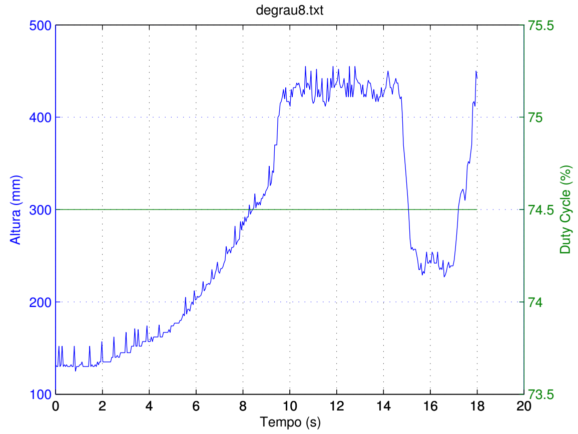
Continuando com os comandos no MATLAB usados para tentar encontrar os parâmetros para o processo da bola-e-tubo:

```
>> show_data
Nome do arquivo (com a extensão) ? degrau8.txt
>> who
Your variables are:
AX        H2        colunas   filename  t
H1        ans       dados     linhas
>> % dados(:,1) = altura da bola
>> % dados(:,2) = duty cycle adotado
>> % projetando FPB
>> RC=1/(2*pi*0.6);
>> Ts=0.047*2; % degrau8.txt saltava 1 amostragem de cada vez
>> yf=lowpass(dados(:,1),Ts,RC);
alpha =
    0.2617
RC/Ts= 2.8219 (esperado >= 5)
Warning: the filter may behave quite differently from the original continuous-time filter
>> t=0:0.047*2:0.047*2*383; % redifinindo vetor tempo
>> figure; plot(t,dados(:,1),'b-', t,yf,'m-')
```

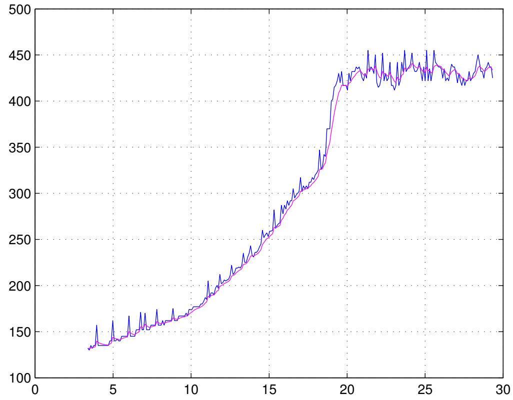
É gerada a figura 14(a). Por inspeção da figura, se percebe que algumas amostras não servem para indentificar o sistema. Assi, trabalhamos com as amostras válidas entre: $3,5 < t < 29,5$ segundos ou as amostras de número 37 até 313 (figura 14(b)).

Usando estes dados para tentar obter os parâmetros do processo usando as funções lsqcurvefit($\cdot$) e step2poles($\cdot$):

```
>> figure; plot(t(37:313),dados(37:313,1),'b-', t(37:313),yf(37:313),'m-')
>> par=lsqcurvefit('step2poles', [420 0.4 0.4], t(37:313)',dados(37:313,1))
Solver stopped prematurely.
lsqcurvefit stopped because it exceeded the function evaluation limit,
```



(a) Visualização dos dados reais amostrados: 'degrau8.txt'..



(b) Região de interesse.

Figura 14: Resposta ao degrau, em malha aberta para processo da bola-e-tubo.

```
options.MaxFunEvals = 300 (the default value).
par =
  416.3118    3.3785    0.9851
>>
```

Testando os parâmetros encontrados: $K = 416.3118$, $\zeta = 3,3785$ e $w_n = 0,9$ com a função **step2poles**$(\cdot)$:

```
>> % fabricando novo vetor tempo: t_teste
>> t(37)
ans =
    3.3840
>> % t(37) = offset no tempo...
>> t_teste=t(37): 0.047*2: t(37)+0.047*2*((313-37)-1);
>> size(t_teste)
ans =
     1   276
>> 313-37
ans =
   276
>> y_teste=step2poles(par,t_teste);
>> figure; plot(t(37:313),dados(37:313,1),'b-', t_teste,y_teste,'m-')
```

O resultado gráfico pode ser visto na figura 15.

Obs.: Note pela figura 15 se percebe que não foram levados em conta 2 parâmetros: um "offset" para a altura da bola e outro "offset" para a dimensão tempo.

De todas as formas os parâmetros encontrados: $K = 416.3118$, $\zeta = 3,3785$ e $w_n = 0,9$, levam à:

$$G(s) \cdot U(s) = \frac{K\, w_n^2}{s(s^2 + 2\zeta w_n s + w_n^2)} \tag{17}$$

$$= \frac{403,9981}{s(s^2 + 6,6563\, s + 0,9704)} \tag{18}$$

$$= \frac{403,9981}{s(s + 6,5071)(s + 0,1491)} \tag{19}$$

considerando que a amplitude da entrada degrau adotada em '**degrau8.txt**' foi $u = 74,5\%$, então:

$$G(s) = \frac{5,5881}{(s + 6,5071)(s + 0,1491)}$$

2.2 Outra tentativa de identificação

Trabalhando com os dados: **dregrau8.txt**, amostrado à $T_s = 0,047 \times 2$ segundos e com base nas equações da tabela 1:

Usando o MATLAB para trabalhar sobre os dados reais:

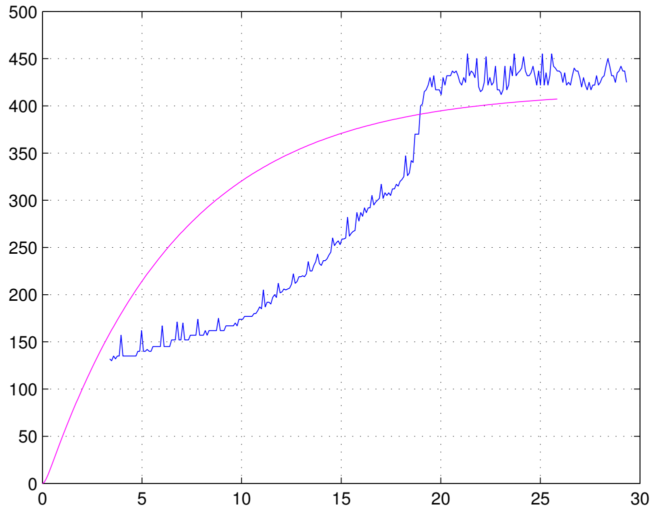


Figura 15: Teste da identificação realizada.

$$\%OS = \frac{y_{max} - y_{final}}{y_{final}} \times 100 \quad (20)$$

$$\zeta = \frac{-\ln(\%OS/100)}{\sqrt{\pi^2 + \ln^2(\%OS/100)}} \quad (21)$$

$$T_s = \frac{4}{\zeta\, w_n} \quad (22)$$

$$w_n = \frac{4}{\zeta\, T_s} \quad (23)$$

$$T_p = \frac{\pi}{w_n\sqrt{1-\zeta^2}} = \frac{\pi}{w_d} \quad (24)$$

$$w_n = \frac{\pi}{T_p\sqrt{1-\zeta^2}} \quad (25)$$

Tabela 1: Equações relacionadas com resposta típica de sistema subamortecido de 2a-ordem

```
>> dados=load('degrau8.txt');
>> size(dados)
ans =
   384     2
>> % fabricando vetor tempo:
>> Ts=0.047*2
>> t=0: Ts: Ts*383;
>> size(t)
ans =
     1   384
>> y1=dados(37:313,1); % separando região de interesse
>> t1=t(37:313);
>> plot(t1,y1)
>> y_min=min(y1) % separando menor valor de y
y_min =
   130
>> y_max=max(y1) % separando maior valor de y
y_max =
   455
>> hold on;
>> % mostrando pontos de interesse num único gráfico
>> aux=num2str(y_max)
```

```
aux =
   455
>> text(t1(1),y_max,aux) % mostra na tela gráfica, valor
>> aux=num2str(y_min);
>> text(t1(1),y_min,aux)
>> aux=num2str(t1(1));
>> text(t1(1),y_min,aux)
```

Até este ponto obtemos o gráfico mostrado na figura 16.

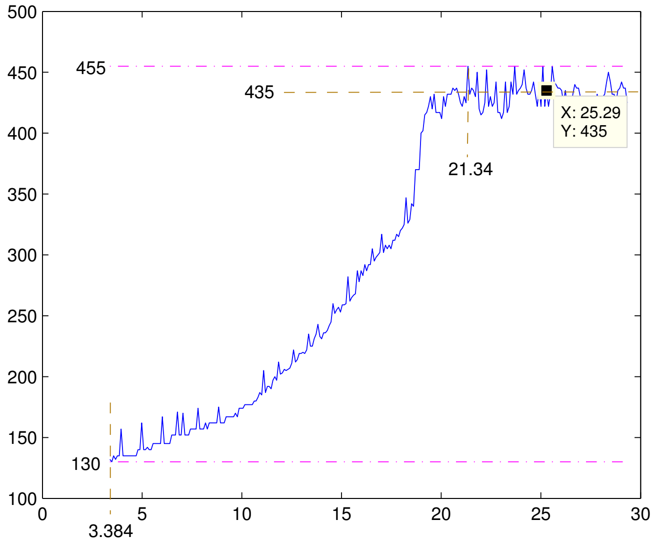


Figura 16: Pontos de interesse na resposta ao degrau do processo da Bola e Tubo.

A função 'lsqcurvefit(myfun)(·),...)' apresenta a possibilidade de se trabalhar com a função **myfun** que aceita constantes como **segundo parâmetro de entrada**.

Observe o seguinte exemplo:

```
function F = myfun(x,xdata,c)
  F = x(1)*exp(c*xdata)+x(2);
```

usada da seguinte forma com a função lsqcurvefit(·):

```
  xdata = [3; 1; 4];            % example xdata
  ydata = 6*exp(-1.5*xdata)+3;  % example ydata
  c = -1.5;                     % define parameter
  x = lsqcurvefit(@(x,xdata) myfun(x,xdata,c),[5;1],xdata,ydata)
```

onde: c seria um segundo argumento de entrada, no caso, uma constante, cujo valor já seria conhecido de antemão. Seria para o nosso caso, o valor do "offset" que é necessário adicionar à saída do processo identificado como um sistema de 2a-ordem. Isto implica em modificar a função 'step2poles(·)' para a nova função: 'step2polesplus(·)':

```
function y = step2polesplus(p,t, constant)
% esta função retorna um vetor de valor de tamanho size(t) que
% corresponde a resposta temporal de um sistema de 2a-ordem subamortecido
% com par de polos complexos, e 0 < fator_amortecimento < 1
% y=K*{1-exp(-zet*wn*t)*[cos(wn*sqrt(1-zeta^2)*t)+(zeta/sqrt(1-zeta^2))*sin(wn*sqrt(1-
   zeta^2)*t)]}
% K --> p(1)
% zeta --> p(2)
% wn --> p(3)
% offset em y --> p(4)
% atraso no tempo --> p(5)
```

```
K=p(1);
zeta=p(2);
wn=p(3);
sigma=zeta*wn;              % parte real do par complexo conjugado
wd=wn*sqrt(1-zeta*zeta); % parte imaginária do par complexo conjugado
b=zeta/(sqrt(1-zeta*zeta));
% y= K*(1 - exp(-sigma.*t).*( cos(wd.*t)+b*sin(wd.*t) ) );
  y= constant + K*(1 - exp(-sigma.*t).*( cos(wd.*t)+b*sin(wd.*t) ) );
```

step2polesplus.m

Outro detalhe antes do momento de voltar a usar a função '**lsqcurvefit**(·)' é que a função '**step2polesplus**(·)' não é capaz de lidar com atraso de transporte, isto é, seu segundo parâmetro de entrada, t, não contempla o caso de uma resposta de sistema de 2a-ordem com atraso no tempo. Então se faz necessário realizar um "shift left"(deslocamente para esquerda) do vetor de dados $y1$ e $t1$ usado anteriormente, como se não existisse o atraso no tempo. Voltando ao MATLAB:

2.3 Identificando como de 3a-ordem, com 1 zero

Comparando-se este processo com a resposta esperada para um processo de 3a-ordem composto apenas de 3 pólos e um atraso no tempo:

$$Y(s) = \frac{K}{(s+p_1)(s+p_2)(s+p_3)} \cdot e^{-s\,t_p}$$

– A EQUAÇÃO ACIMA ESTÁ CONFORME COM O QUE SEGUE ?
ERRO: DEVERIA SER UM ZERO + 3 PÓLOS = SISTEMA COM Ku

– REFAZER PARA RESPOSTA À DEGRAU MALHA ABERTA PARA SISTEMAS DE 2A-ORDEM ***

onde se aplica um sinal degrau de amplitude A, obteremos como resposta em malha aberta:

$$R(s)\,Y(s) = \frac{A\,K}{s(s+p_1)(s+p_2)(s+p_3)} \cdot e^{-s\,t_p}$$

desconsiderando o atraso no tempo, a transformada inversa de Laplace da equação acima é dada pela equação (8):

$$y(t) = k_1 + k_2 \cdot e^{Re\{p_1\}} \cdot [\cos(\text{Im}\{p_1\} \cdot t) + k_3 \cdot \sin(\text{Im}\{p1\} \cdot t)] + k_4 e^{p_3\,t} \tag{26}$$

Aplicando-se no caso, podemos aproximar o atraso no tempo à 6 segundos, o que equivale a trabalhar a partir da amostra de número $k = 128$ $(6/0,047 = 127,6596)$. Podemos usar o método numérico dos mínimos quadrados para tentar obter os 7 parâmetros: k_1, k_2, k_3, k_4, $\text{Re}\{p_1\}$, $\text{Im}\{p_1\}$ e p_3 da equação (26) para o processo em questão. Para tanto, necessitamos codificar esta função no MATLAB – função 'step3poles', realizando uma pequena adaptação na equação (26):

$$y(t) = p(1) + p(2) \cdot e^{p(5)\,t} \cdot \{\cos[p(6)\,t] + p(3)\,\sin[p(6)\,t]\} + p(4)\,e^{p(7)\,t} \tag{27}$$

```
% Função que devolte y(t) para entrada degrau de
% sistema com 3 pólos somente
% y(t)=k1-k2*exp(Re{p1}*t)*[cos(|Im{p2}|*t+k3*sin(|Im{p2}|*t]-k4*exp(p3*t)
% parâmetros de entrada: 7 + t
%                         1  2  3  4   5     6    7
% function resp = step3poles(k1,k2,k3,k4,Re_p1,Im_p1,p3,t)
function resp = step3poles(p, t)
  % resp= k1 + k2.*exp(Re_p1.*t).*(cos(Im_p1.*t)+k3*sin(Im_p1.*t)) + k4.*exp(p3.*t);
    resp= p(1)+p(2).*exp(p(5).*t).*(cos(p(6).*t)+p(3)*sin(p(6).*t))+ p(4).*exp(p(7).*t);
```

step3poles.m

E neste momento podemos fazer uso da função 'lsqcurvefit(·)' do MATLAB:

```
>> size(dados)
ans =
   384     2
>> par=lsqcurvefit('step3poles', [1/5 -1/10 1/2 -1/50 -1 -2 -3], dados(128:384,1), dados(128:384,2))
Local minimum found.
Optimization completed because the size of the gradient is less than
```

```
the default value of the function tolerance.
<stopping criteria details>
par =
  755.0000   -0.1000    0.5000   -0.0200   -1.0000   -2.0000   -3.0000

Optimization completed: The first-order optimality measure, 0.000000e+00,
is less than options.TolFun = 1.000000e-06.
Optimization Metric                                        Options
relative first-order optimality =   0.00e+00     TolFun =   1e-06 (default)
>>
```

Neste caso, com o resultado obtido pela função lsqcurvefit() obtemos a seguinte função $y(t)$:

$$y(t) = 755 - 1 \cdot e^{-1\,t} \cdot \{\cos[-2\,t] + (1/2) \cdot \sin[-2\,t]\} - (2/100) \cdot e^{-3\,t} \tag{28}$$

ou seja, não parece que função tenha convergido (encontrado valores) adequadamente para os parâmetros: p_5 ($\text{Re}\{polo_1\}$), p_6 ($\text{Im}\{polo_1\}$) e p_7 (posição do pólo real). Executando novamente a função, mas desta vez como novos valores iniciais:

*** REFAZER códigos no MATLAB, SEPARANDO as variáveis utilizadas para evitar enventuais confusões (erros) ****

Observações:

1. Local Minimum Found: The solver located a point that seems to be a local minimum of the sum of squares, since the first-order optimality measure is close to 0. A local minimum of a function is a point where the function value is smaller than at nearby points, but possibly greater than at a distant point or global minimum is a point where the function value is smaller than at all other feasible points – veja figura 17.

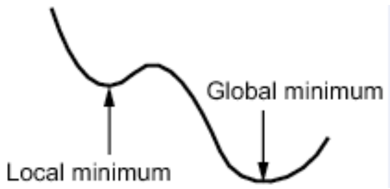


Figura 17: O problema do mínimo local.

2. Function Tolerance: The function tolerance called TolFun relates to both:

   - The size of the latest change in objective function value.
   - The value of the first-order optimality measure

   Ver figura 18

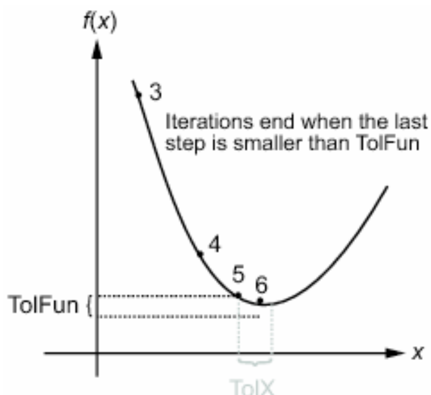


Figura 18: Estabelecendo uma tolerância para a aproximação realizada.

Referências

[1] Nise, Norman S.; Control Systems Engineering, 4th Edition (Hardcover), Wiley; 4 edition (September 2, 2003), http://bcs.wiley.com/he-bcs/Books?action=index&itemId=0471445770&itemTypeId=BKS&bcsId=1758 by

[2] 2011 Spring - EE381 Control System Engineering (English Lecture), Professor : Ju-Jang Lee, http://iliad.kaist.ac.kr/english/frameset/frameset.htm (Disponivel em 12/04/2011).

[3] Interactive Mathematics, Table of Laplace Transformations, http://www.intmath.com/Laplace-transformation/Table-laplace-transforms.php (Disponível em 12/04/2011).

[4] Laplace Transform, Wolfram MathWorld (the web's most extensive mathematics resource, http://mathworld.wolfram.com/LaplaceTransform.html (Disponivel em 12/04/2011).

[5] R2011a Documentation – Symbolic Math Toolbox, MathWorks, ilaplace (Inverse Laplace transform, http://www.mathworks.com/help/toolbox/symbolic/ilaplace.html (Disponivel em 12/04/2011).

[6] R2011a Documentation – Control System Toolbox, MathWorks, sgrid, http://www.mathworks.com/help/toolbox/control/ref/sgrid.html (Disponível em 12/04/2011).

[7] EE 6319 Control Theory, Computer Assignment 01, Prof. Ariyadasa L.P.N.I., Dept. Electrical and Informatics Engineering, Faculty of Engineering, University of Ruhuna, 05/16/2009, http://www.scribd.com/doc/15492580/Matlab-for-Control-Theory (Disponivel em 12/04/2011).

A Transformada Inversa de Laplace, no MATLAB

Com o MATLAB é possível de determninar a transformada inversa de Laplace de algumas funções. Para tanto se faz necessário o uso do toolbox 'Symbolic Math Toolbox'.
Exemplos:

Ex. 1: Seja:

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

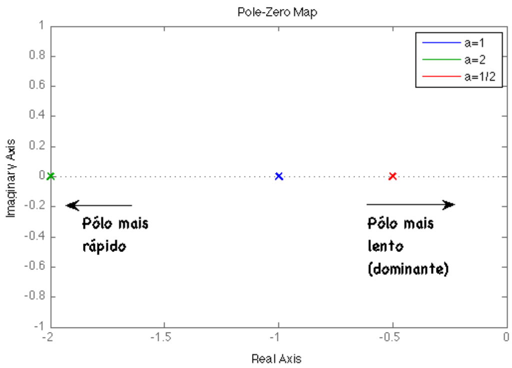
Sua inversa no MATLAB (ou seja, a resposta no tempo para uma entrada impulso unitário) ficaria:

```
>> syms s a;
>> g=1/(s+a);
>> ilaplace(g)
ans =
1/exp(a*t)
>>
```

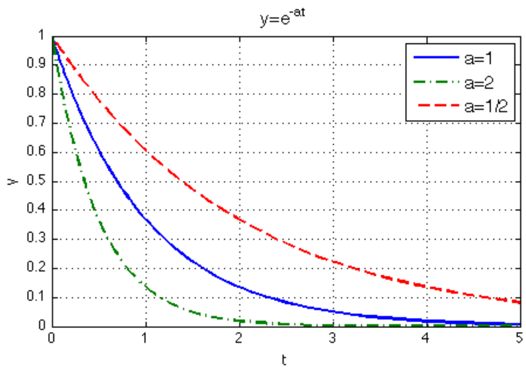
Resposta:

$$\mathcal{L}^{-1}\{G(s)\} = e^{-a\,t}$$

A figura 19 d? uma id»ia gr?fica do resultado.



(a) P?los no plano-s.



(b) Resposta no tempo.

Figura 19: Resposta ao impulso unit?rio para sistema de 1a-ordem (1 p?lo).

Seguem abaixo os comandos usados no MATLAB para criar a figura 19 acima

```
>> % comandos para criar plano-s
>> den1=[1 1];
>> den2=[1 2];
>> den3=[1 0.5];
>> [z1 p1 k1]=tf2zp(num,den1);
>> [z2 p2 k2]=tf2zp(num,den2);
>> [z3 p3 k3]=tf2zp(num,den3);
>> pzmap(p1,z1);
>> hold on; pzmap(p2,z2)
>> hold on; pzmap(p3,z3);
>> legend('a=1','a=2','a=1/2')
>> % comandos para criar resposta no tempo
>> [x1,y1]=fplot(@(t) exp(-t),[0 5]);
>> [x2,y2]=fplot(@(t) exp(-2*t),[0 5]);
>> [x3,y3]=fplot(@(t) exp(-0.5*t),[0 5]);
>> plot(x1,y1, x2,y2, x3,y3); grid
>> legend('a=1','a=2','a=1/2')
>> title('y=e^{-at}')
>> xlabel('t')
>> ylabel('y')
```

Ex. 2: Mesmo sistema do exemplo anterior, mas submetido a uma entrada degrau unitário (malha aberta). A função transferência se modifica agora para

Ex. 3: Seja:

$$
\begin{aligned}
G(s) &= Y(s)\cdot U(s) = \frac{1}{(s+a)}\cdot\frac{1}{s} \\
&= \frac{1}{s(s+a)}
\end{aligned}
$$

A inversa de $G(s)$, usando o MATLAB, fica:

```
>> clear all
>> syms s a;
>> g=1/(s*(s+a));
>> ilaplace(g)
ans =
1/a - 1/(a*exp(a*t))
>>
```

Ou seja:

$$
\begin{aligned}
\mathcal{L}^{-1}\{G(s)\} &= \frac{1}{a} - \frac{1}{a\,\mathrm{e}^{a\,t}} \\
&= \frac{1}{a}\left(1 - e^{-a\,t}\right)
\end{aligned}
$$

A figura

Ex. 4: Seja um sistema de 2a-ordem composto de apenas 2 pólos (estáveis):

$$
G(s) = \frac{1}{(s+p_1)(s+p_2)}
$$

No MATLAB sua inversa ficaria:

```
>> clear
>> syms s p1 p2;
>> g=1/((s+p1)*(s+p2));
>> ilaplace(g)
ans =
1/(exp(p2*t)*(p1 - p2)) - 1/(exp(p1*t)*(p1 - p2))
>>
```

Ou seja, a resposta ≫:

$$
\begin{aligned}
\mathcal{L}^{-1}\{G(s)\} &= \frac{1}{\mathrm{e}^{\mathrm{p2}\,t}\,(\mathrm{p1}-\mathrm{p2})} - \frac{1}{\mathrm{e}^{\mathrm{p1}\,t}\,(\mathrm{p1}-\mathrm{p2})} \\
&= \frac{e^{-p_2\,t}}{(p_1-p_2)} + \frac{e^{-p_1\,t}}{(p_1-p_2)}
\end{aligned}
$$

B Relações trigonométricas

$$
\begin{aligned}
\sin(-\theta) &= -\sin(\theta) \\
\sin(\theta+\pi) &= -\sin(\theta) \\
\sin(\theta+\pi/2) &= +\cos(\theta) \\
\cos(-\theta) &= +\cos(\theta) \\
\cos(\theta+\pi) &= -\cos(\theta) \\
\cos(\theta+\pi/2) &= -\sin(\theta)
\end{aligned}
$$

Combinação linear de senos e cossenos:

$$x\sin(\alpha) + y\cos(\alpha) = R\sin(\alpha+\beta) \quad (29)$$
$$x\cos(\alpha) - y\sin(\alpha) = R\cos(\alpha+\beta) \quad (30)$$
$$R = \sqrt{x^2+y^2} \quad (31)$$
$$\beta = \arctan(y/x) \quad (32)$$
$$(33)$$

Exemplo:

$$f(t) = [4 - 4\mathrm{e}^{-2t}\cos(t) + 2\mathrm{e}^{-2t}\sin(t)]u(t)$$

resulta em:

$$f(t) = [4 + 4,47\mathrm{e}^{-2t}\cos(t - 153,4^o)]u(t)$$

Ou, de outra forma:

$$a\sin(x) + b\cos(x) = \sqrt{a^2+b^2}\cdot\sin(x+\varphi)$$

onde:

$$\varphi = \begin{cases} \arcsin\left(\frac{b}{\sqrt{a^2+b^2}}\right) & \text{se } a \geq 0, \\ \pi - \arcsin\left(\frac{b}{\sqrt{a^2+b^2}}\right) & \text{se } a < 0. \end{cases}$$

ou:

$$\varphi = \arctan\left(\frac{b}{a}\right) + \begin{cases} 0 & \text{se } a \geq 0, \\ \pi & \text{se } a < 0. \end{cases}$$

C Relações de Euler

Relações de Euler e funções exponenciais:

$$\cos(x) = \mathrm{Re}\{e^{ix}\} = \frac{e^{ix}+e^{-ix}}{2}$$
$$\sin(x) = \mathrm{Im}\{e^{ix}\} = \frac{e^{ix}-e^{-ix}}{2i}$$

ou

$$e^{ix} = \cos(x) + i\sin(x)$$
$$e^{-ix} = \cos(-x) + i\sin(-x) = \cos(x) - i\sin(x)$$

D Filtro Passa Baixa Digital

Um filtro PB de 1a-ordem no formato digital é dado por:

$$y[k] = \alpha\, x[k] + (1-\alpha)\, y[k-1] \quad (34)$$

onde: $y[k]$=saída atual do filtro; $y[k-1]$=amostra atrasada da saída do filtro (de um período de amostragem), $x[k]$=sinal de entrada (sem atraso); α=fator de amortecimento ($0 \leq \alpha \leq 1$) e onde α é calculado como:

$$\alpha = \frac{T_s}{(RC+T_s)}$$

onde: T_s=valor do período de amostragem adotado e RC=constante de tempo do filtro (a mesma que seria usada num filtro PB analógico formado por um circuito TC). Este tipo de filtro é dito também de média móvel de ponderação exponencial.

Este filtro (eq. 34) no formato de um algoritmo para MATLAB (**lowpass.m**) fica:

```
% Filtro passa-baixa de 1a-ordem
% Média móvel de ponderação exponencial (EWMA)
%
% function y=lowpass(x,Ts,RC)
%
% Parâmetros de entrada:
%   RC=1/(2*pi*fc)=constante de tempo do filtro
```

```
%   Ts=período de amostragem adotado (em s)
% onde:
%   fc=freq. de corte do filtro (em Hz)
% Internamente será calculado 'alpha':
% alpha = Ts/(RC+Ts) = parâmetro do filtro
% É importante que Ts<= RC/5, senão este filtro apresenta
% um comportamento muito diferente do esperado.
function y=lowpass(x,Ts,RC)
    alpha=Ts/(RC+Ts)
    fprintf('RC/Ts=%7.4f (esperado >= 5)\n',RC/Ts);
    if (Ts>(RC/5))
        fprintf('Warning: the filter may behave quite differently from the original
            continuous-time filter\n');
    end
    y(1)=x(1);
    amostras=length(x);
    for i=2:amostras
        % y(i)= alpha*x(i) + (1-alpha)*y(i-1); % eq. (1)
        y(i)= y(i-1) + alpha*(x(i) - y(i-1)); % eq. (2)
    end
% return y
```

lowpass.m

A equação (34) pode ser re-escrita no formato:

$$y[k] = y[k-1] + \alpha \cdot (x[k] - y[k-1]) \tag{35}$$

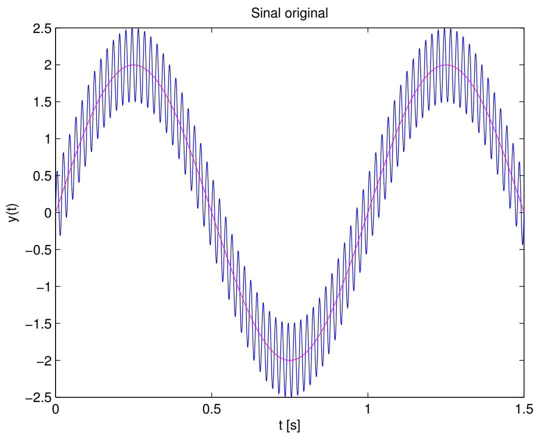
Realizando um teste: suponha que o sinal a ser filtrado seja dado por: $y(t) = 2\sin(2\pi 1) + 1/4\sin(2\pi 50)$ – conforme mostra a figura 20(a), ou seja, o sinal principal oscila a 1 Hz enquanto sobreposto a este sinal há um ruído de25% de amplitude e freqüência de 50 Hz. Se for aplicado um FPB com $f_c = 5$ [Hz], no formato digital trabalhando com período de amostragem de $T_s = 1$[ms] ($f_s = 1$ KHz), obteremos algo como mostrado na figura 20(b). Aplicando esta informação sobre a rotina de teste mostrada abaixo (sinal.m), obtêm-se o resultado mostrado na figura 20(b).

```
% sinal.m
% montando sinal x(:,1) no tempo para teste do filtro PB
% entradas:
%   Ts= periodo de amostragem
%    f = frequencia base da senoide
clear x
i=0; % contador do numero da amostra
f=1;
T=1/f;
for k=0:Ts:(1.5*T) % gera 2,5 períodos
    i=i+1;
    x(i,1)=k; % variável tempo
    x(i,2)=2*sin(2*pi*1*k)+0.5*sin(2*pi*50*k); % sinal base + "ruído"
    x(i,3)=2*sin(2*pi*1*k); % sinal base
end
figure; plot(x(:,1),x(:,2),'b-', x(:,1),x(:,3),'m-')
```

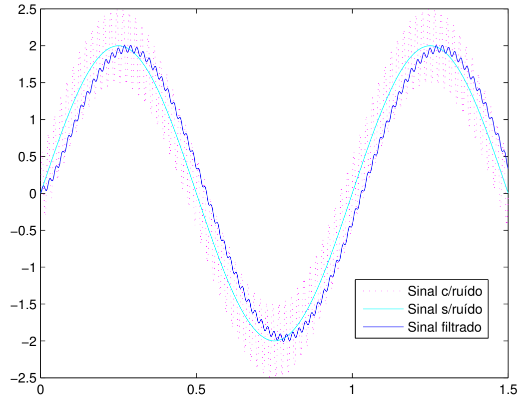
sinal.m

Comandos usados no MATLAB:

```
>> Ts=1e-3;
>> sinal
>> RC=1/(2*pi*5)
RC =
     0.0318
>> y=lowpass(x(:,2),Ts,RC);
alpha =
     0.0305
RC/Ts=31.8310
>> figure; plot(x(:,1),x(:,2),'m:', x(:,1),x(:,3),'c-', x(:,1),y,'b-')
>> legend('Sinal c/ruído','Sinal s/ruído', 'Sinal filtrado')
```



(a) Sinal original.



(b) Sinal filtrado.

Figura 20: Teste do filtro digital FPB.

Note pela figura 20(b) que como a freqüência do ruído está uma década acima da freqüência de corte do filtro, sua amplitude decaiu de 20 dB ou seja de 1/10.

Material criando usando LaTeX

Referências:

- Introdução ao LaTeX: **http://en.wikibooks.org/wiki/LaTeX/Introduction**
- Tutorial básico sobre LaTeX: **http://en.wikipedia.org/wiki/LaTeX**.
- Outro tutorial sobre LaTeX: **http://www.andy-roberts.net/misc/latex/index.html**.
- Tutorial não tão curto sobre LaTeX: **http://mirror.softwarelivre.ufsc.br/pub/ctan/info/lshort/english/lshort.pdf** – PDF de 157 páginas.
- Material avançado: **http://nitens.org/taraborelli/tools**