Table of Contents

Pendulo de Mola 1
Definição dos parâmetros do experimento 1
Veja a montagem experimental do sistema em estudo 2
Carga dos dados 2
Visualização dos dados 3
Análise 4
Análise de Frequencia 5
Uso da função 'periodogram' (cálculo do periodograma) 5
Cálculo de T e w 6
Filtro 6
Visualização do resultado do filtro 7
Cálculo da constante de tempo 7
Ajuste dos dados para obter a constante de tempo de decaimento 7
Usa o m-file, automaticamente, gerado pelo 'cftool' para extrair os parametros 8
Avalia a qualidade do ajuste 8
Cálculo de outros parâmetros 10
Cálculo do valor de L (comprimento final da mola) 10
Este material pode ser obtido em: 11

Pendulo de Mola

Análise dos dados de um sistema de pêndulo de mola. Os dados foram obtidos no site:

http://nonlinear.sdsu.edu/~carreter/teach/M636_projects/spring_pendulum/index.html

e

http://www.rohan.sdsu.edu/~jmahaffy/courses/f01/math636/lectures/spring_pendulum/springpend.html

Um primeiro experimento foi executado com uma massa de aço sendo puxada para baixo verticalmente. O movimento foi captado utilizando-se um detector de movimento que coletou dados na frequencia de 30 vezes por segundo. Esses dados foram disponibilizados em uma planilha de Excel.

O objetivo do experimento é, a partir desses dados, estimar:

1) o período de oscilação (T)

2) a constante elástica da mola (k)

3) o comprimento final da mola (L)

Inicialização

```
clc; clear all; close all;
```

Definição dos parâmetros do experimento

O projeto físico do pêndulo de mola usou uma massa de aço (de 3.5cm de diâmetro e peso de 233g), presa a uma mola de metal (de 155g). Essa configuração foi montada em uma plataforma a cerca de 1 metro acima do chão. A mola em repouso tinha um comprimento total de 18cm com 14cm de espiral)

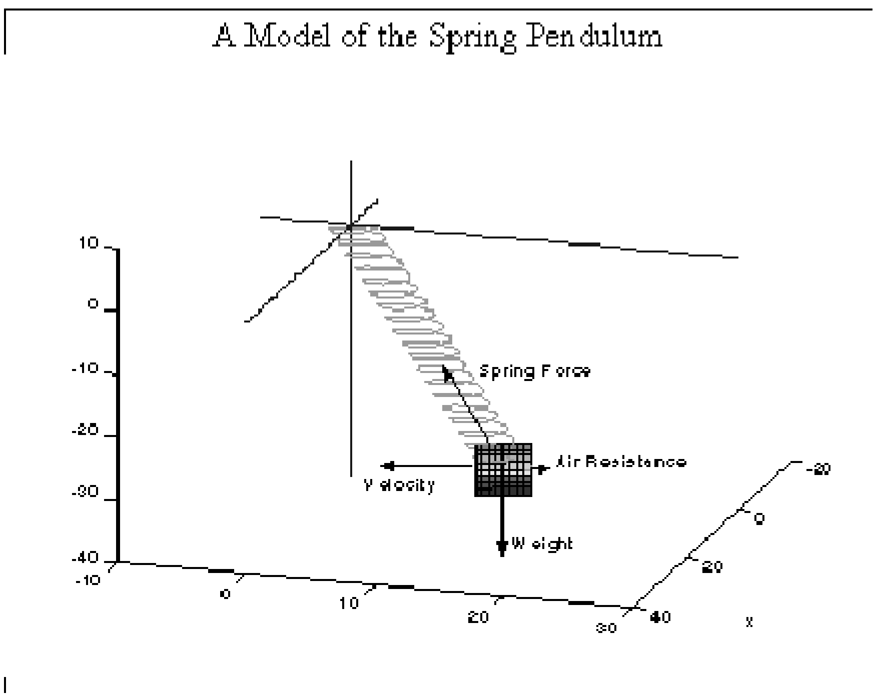
O sistema de medidas utilizado foi kg/m/s.

```
g = 9.8;            % m/s^2
m_ball = 0.233;     % kg
m_spring = 0.155;   % kg
l_spring = 0.18;    % m   Comprimento em repouso (nenhuma força em ação)
f0 = 30;            % Hz  Frequencia de amostragem da aquisição de dados
```

Veja a montagem experimental do sistema em estudo

O diagrama mostra o sistema em estudo e as forças agindo no pêndulo

```
pen = imread('spring_pen.jpg');
imagesc(pen), axis off
set(gcf,'Colormap',gray);
```



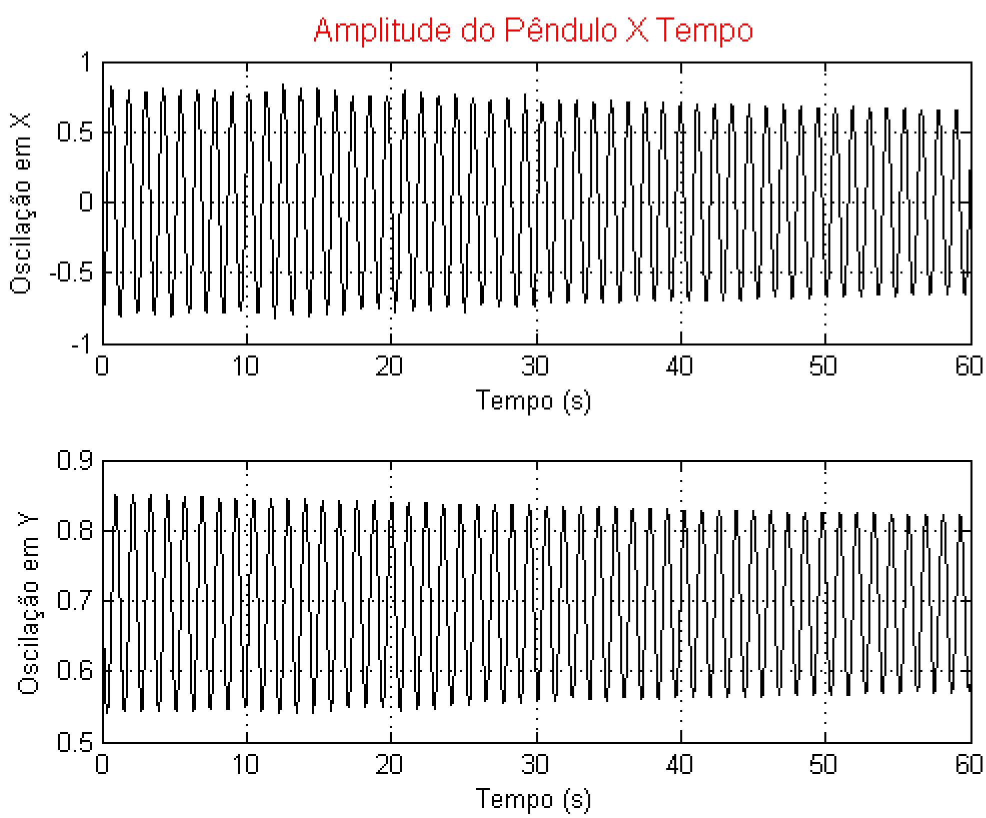
Carga dos dados

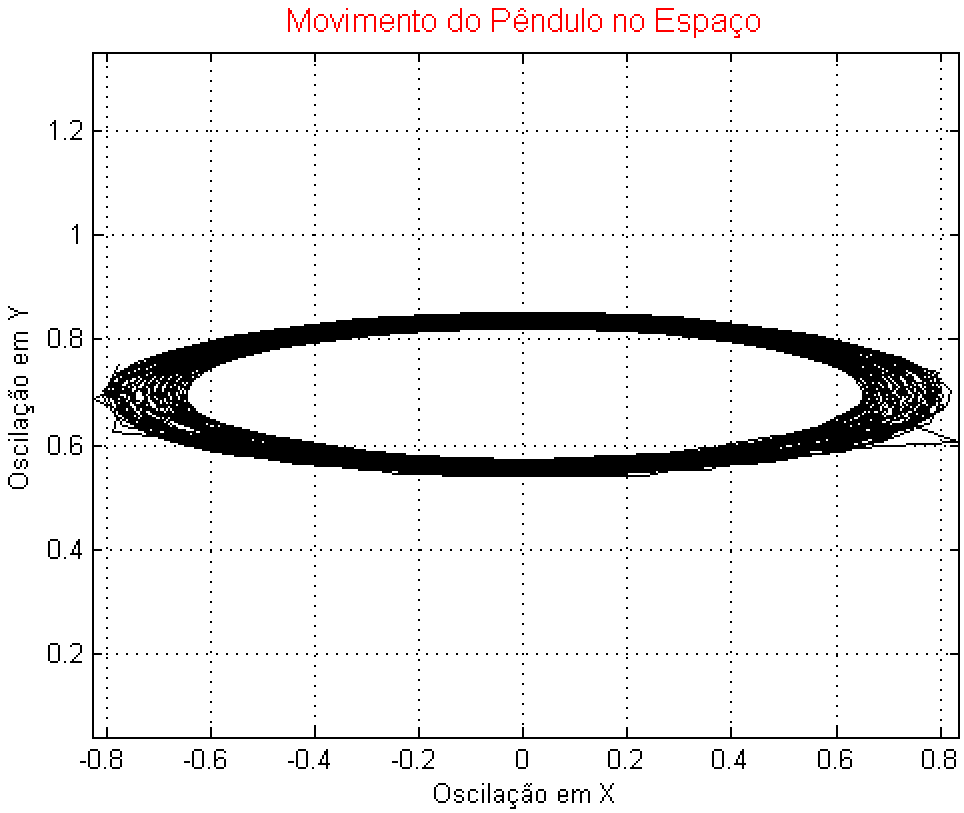
Importação dos dados na planilha de Excel

```
pendulum = xlsread('pendulo_dados.xls');
time = pendulum(:,1);                % Tempo de amostragem
y_oscillations = pendulum(:,2);      % Oscilação em Y
x_oscillations = pendulum(:,3);      % Oscilação em X
```

Visualização dos dados

```
figure(2)
subplot(2,1,1)
plot(time,x_oscillations,'k')
xlabel ('Tempo (s)')
ylabel ('Oscilação em X')
title('Amplitude do Pêndulo X Tempo','color','r','FontSize',12)
grid on, zoom on
subplot(2,1,2)
plot(time,y_oscillations,'k')
xlabel ('Tempo (s)')
ylabel ('Oscilação em Y')
grid on, zoom on
hold off

figure(3)
plot(x_oscillations,y_oscillations,'k')
xlabel ('Oscilação em X')
ylabel ('Oscilação em Y')
title('Movimento do Pêndulo no Espaço','Color','r','FontSize',12)
grid on, zoom on, axis equal
```





Análise

O modelo linear matemático para o movimento vertical, encontrado na maioria dos livros de Física, assume que o peso da mola é desprezível e que o peso da massa está todo concentrado em um único ponto. Nessas condições, a equação de amortecimento da mola é dada por:

```
my" + ay' + ky = 0,
```

Onde: - y(t) é o deslocamento da massa de aço a partir da posição de equilíbrio. - m é o peso da massa (variável m_ball)

A solução geral para essa equação é dada por:

```
y(t) = exp(c1*t)*(c2*cos(wt) + c3*sin(wt)) =
A*exp(c1*t)*cos(wt + z)
```

onde

```
c1 = a/2m

w^2 = (4km - a^2)/4m^2

tan(z) = -c3/c2
```

e

```
A^2 = c2^2 + c3^2
```

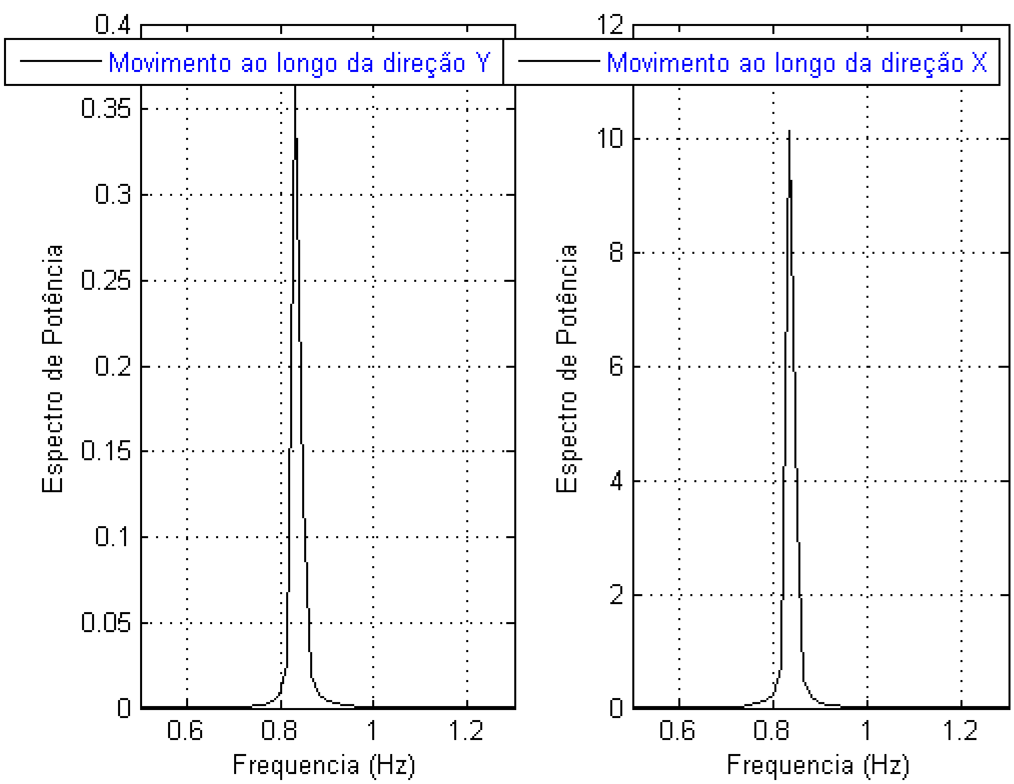
Como as oscilações diminuem (são amortecidas), esperamos um valor negativo para c1.

Análise de Frequencia

w pode ser calculado a partir da análise de frequencia dos dados Podemos usar a função 'periodogram' do Signal Processing Toolbox para obter o espectro de potência dos dados, e as funções 'max' e 'find' do MATLAB para extrapolar as frequencias de oscilação (a longo das direções x e y).

Uso da função 'periodogram' (cálculo do periodograma)

```
ndata = size(time,1);
nfft = ndata/2;

freq = (0:nfft-1)*(f0/ndata);
PSy = periodogram(y_oscillations,[],ndata-1,f0,'onesided');
PSx = periodogram(x_oscillations,[],ndata-1,f0,'onesided');

grafico_espectro_de_potencia(freq,PSy,PSx)
```



Cálculo de T e w

```
MaxPSy = max(PSy(10:nfft));
ny_max = find(PSy == MaxPSy);
% Frequencia máxima ao movimento ao longo de Y
fy_max = freq(ny_max)
w_y = 2*pi*fy_max;
% Período de oscilação para o movimento ao longo de Y
Ty = 1/fy_max


MaxPSx = max(PSx(10:nfft));
nx_max = find(PSx == MaxPSx);
% Frequencia máxima ao movimento ao longo de X
fx_max = freq(nx_max)
w_x = 2*pi*fx_max;
% Período de oscilação para o movimento ao longo de X
Tx = 1/fx_max
```

```
fy_max =

   0.832408435072142


Ty =

   1.201333333333333


fx_max =

   0.832408435072142


Tx =

   1.201333333333333
```

Filtro

Como os dados podem apresentam um pouco de ruído, antes de ajustar as curvas (para extrapolar a constante de tempo), aplicamos um filtro. Utilizamos o comando interativo do Signal Processing Toolbox, 'fdatool', que permite definir o filtro a ser aplicado(banda baixa, banda alta, etc), o método (IIR - Infinite Impulse Response ou FIR - Finite Impulse Response) e o algoritmo a ser utilizado. O mesmo comando permite visualizar, imediatamente, a resposta do filtro em amplitude e em fase, bem como outras características.

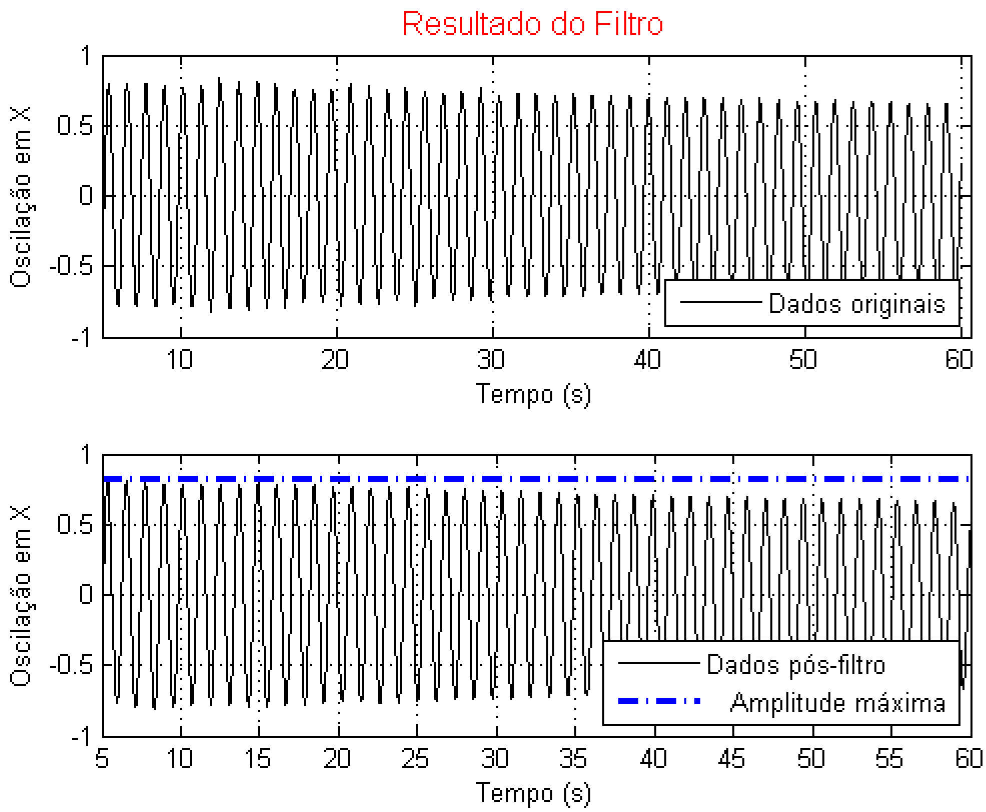
```
%fdatool %
```

Escolhemos um filtro banda baixa (low-pass), IIR, com frequencia de corte (cut-off frequency) igual a 1.5 Hz, frequencia de parada (stop-band frequency) igual a 2 Hz, e frequencia de amostragem 30. O filtro criado pode ser armazenado em um m-file para ser posteriormente chamado pela função 'filter'.

```
new_x_oscillations = filter(filtro_lowpass,x_oscillations);
```

Visualização do resultado do filtro

```
grafico_resultado_filtro(time,x_oscillations,new_x_oscillations);
```



Cálculo da constante de tempo

Uma vez calculado w, podemos calcular a constante de tempo de amortecimento da oscilição. Para fazer isso, utilizamos o comando 'cftool' do Curve Fitting Toolbox. O 'cftool' permite selecionar, manualmente, os pontos a serem ajustados, escolher a equação de ajuste, a partir de um conjunto de equações pre-definidas, ou definir uma equação. Além disso, permite visualizar imediatamente os coeficientes de ajuste e a curva.

Para calcular a constante de tempo, precisamos calcular o tempo de decaimento da oscilação. Consideramos apenas os pontos nos quais cos(w*t+z) = 1, isto é, o topo da curva de movimento, e ajustamos esse 'perfil'.

Ajuste dos dados para obter a constante de tempo de decaimento

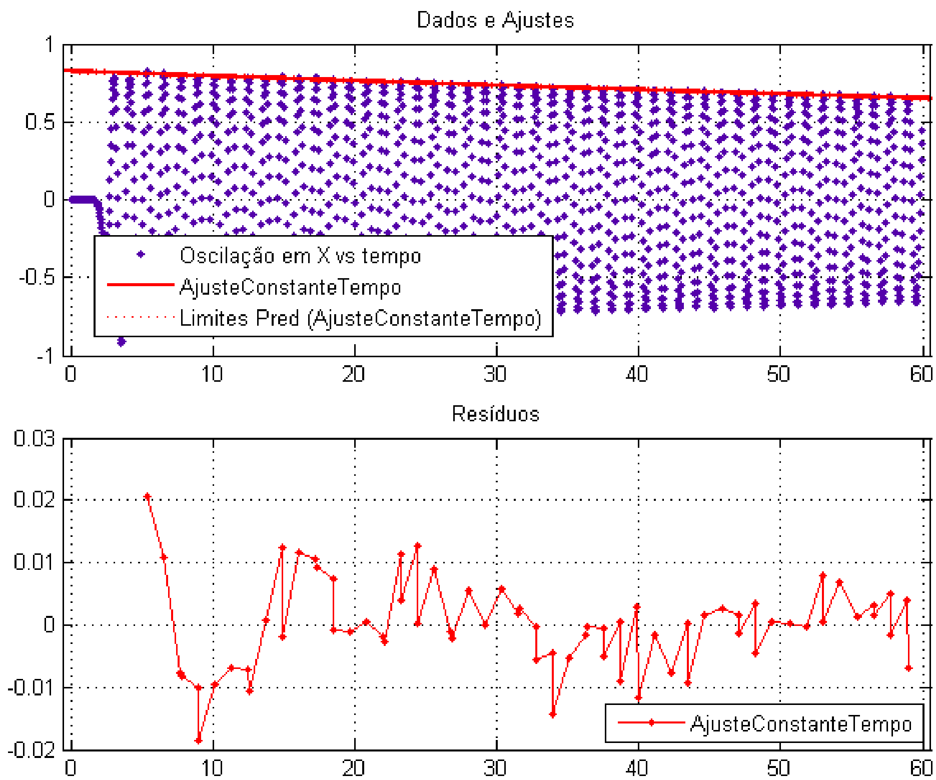
```
%cftool %
```

Usa o m-file, automaticamente, gerado pelo

'cftool' para extrair os parametros

```
[Res,Coeff,Coeff_int] = ajuste_oscilacao_x(time,new_x_oscillations);
Coeff
```

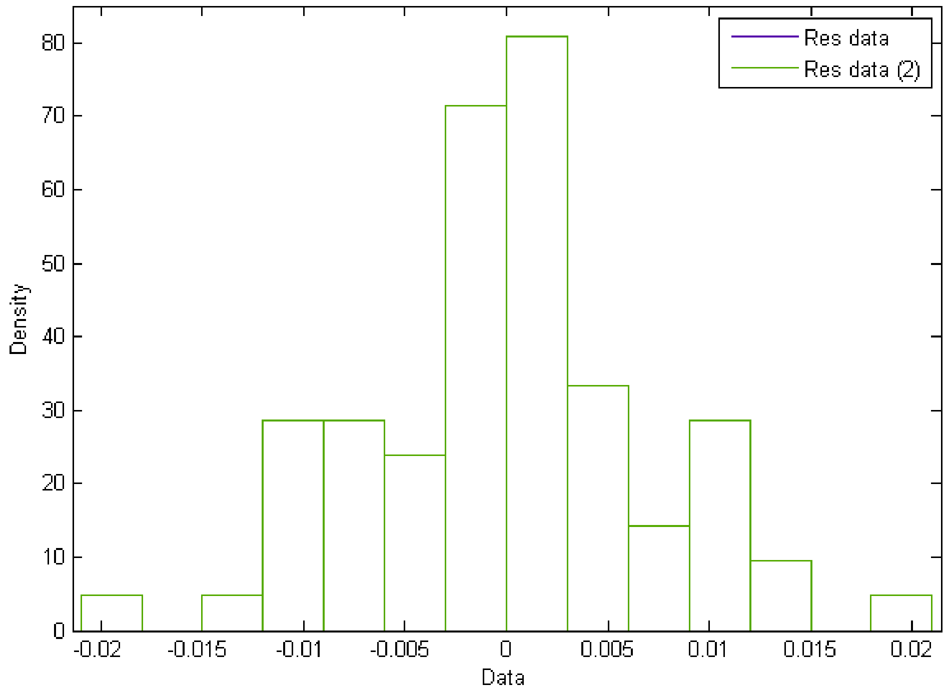
```
Coeff =

   0.825296936343800  -0.003929783127837
```



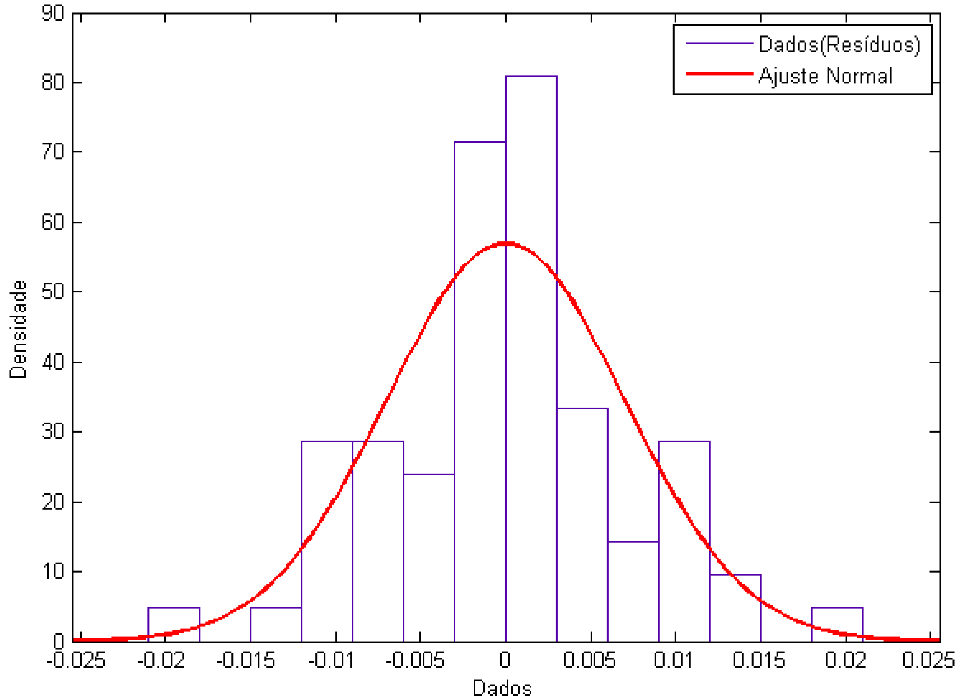
Avalia a qualidade do ajuste

Os resíduos devem ter distribuição normal/gaussiana

```
dfittool(Res) %
```




```
distr_residuos(Res)
```



Cálculo de outros parâmetros

Por meio de contas simples, podemos obter outros parâmetros a partir dos valores conhecidos de m e w

Podemos obter c1 = a/2m ===> a = 2m * c1; (como esperado, o valor é negativo)

```
c1 = Coeff(2);
a = 2*m_ball*c1                % kg/s

% w = (4km - a^2)/4m    ===>  k = m*c1 + w
k = m_ball*(c1^2 + w_x^2)      % kg/s^2
```

```
a =

  -0.001831278937572


k =

   6.373659343048167
```

Cálculo do valor de L (comprimento final da mola)

Podemos calcular o valor de L (o novo comprimento da mola, após o alongamento) usando dois tipos diferentes de relação:

Lei do Pêndulo

T = 2*pi*sqrt(L/g)

e a Equação de Equilíbrio do Pêndulo de mola k*L = m*g

Os dois valores obtidos deverão ser iguais

```
L1 = (Tx^2)*g/(4*(pi^2))
L2 = m_ball*g/k
```

```
L1 =

    0.358255935279968


L2 =

    0.358255733025728
```

Este material pode ser obtido em:

http://www.mathworks.com/matlabcentral/

Published with MATLAB® 7.9