HW1 CEE 744 Spring 2013

Nasser M. Abbasi, Feb 2, 2013

Reading the data from file and ready it for processing

In[9]:=

```
SetDirectory[NotebookDirectory[]];
Clear[data, yy, y, t];
data = Import["free_vibr.txt", "Elements"]
```

Out[11]=

```
{Data, Lines, Plaintext, String, Words}
```

In[12]:=

```
data = StringSplit[Import["free_vibr.txt", "Lines"]];
Dimensions[data]
```

Out[13]=

```
{8192, 3}
```

Show 3 lines of data

In[14]:=

```
data[[1 ;; 3]] // TableForm
```

Out[14]//TableForm=

```
1/22/2013    12:52:00.987959    -1.171216E-1
1/22/2013    12:52:00.988936    -1.152905E-1
1/22/2013    12:52:00.989912    -1.183423E-1
```

pull out the time and the voltage columns

In[15]:=

```
filteredData = Transpose[{ToExpression[Part[StringSplit[#, ":"], 3]] & /@ data[[All, 2]],
    Internal`StringToDouble[#] & /@ data[[All, 3]]}];
Dimensions[filteredData]
```

Out[16]=

```
{8192, 2}
```

Show 3 lines of the above result

In[17]:=

```
filteredData[[1 ;; 3]] // TableForm
```

Out[17]//TableForm=

```
0.987959    -0.117122
0.988936    -0.115291
0.989912    -0.118342
```

Filter the data

Make the data start at zero

In[18]:=
```
filteredData[[All, 1]] = filteredData[[All, 1]] - filteredData[[1, 1]];
```

Normalize the data by subtracting the mean

In[19]:=
```
mean = Mean[filteredData[[All, 2]]]
```

Out[19]=
```
-0.00600079
```

In[20]:=
```
filteredData[[All, 2]] = filteredData[[All, 2]] - mean;
```

In[21]:=
```
filteredData[[1 ;; 3]] // TableForm
```

Out[21]//TableForm=

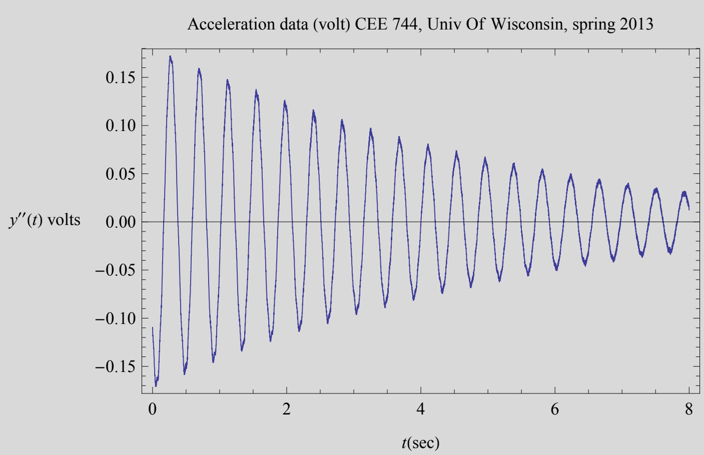
0.	-0.111121
0.000977	-0.10929
0.001953	-0.112342

Plot the data before analysis

first in raw data as volts

In[22]:=
```
ListLinePlot[filteredData, Frame → True,
 FrameLabel → {{Row[{y''[t], " volts"}], None}, {Row[{t, "(sec)"}],
    "Acceleration data (volt) CEE 744, Univ Of Wisconsin, spring 2013"}},
 RotateLabel → False, GridLines → Automatic, GridLinesStyle → LightGray]
```

Out[22]=



Convert to fractions of g

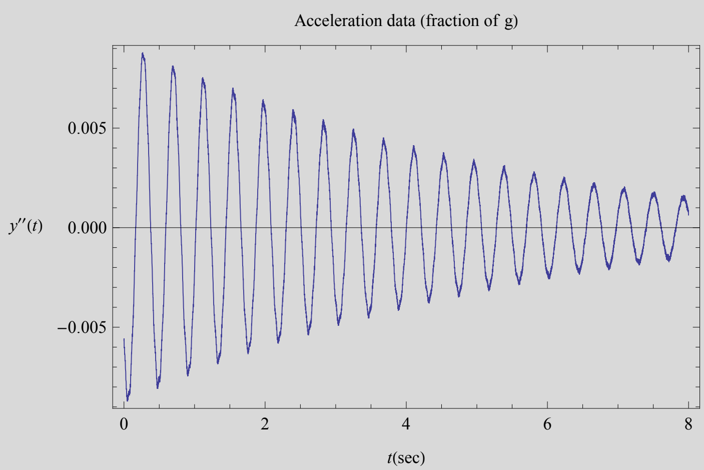
In[23]:=

```
filteredData[[All, 2]] = filteredData[[All, 2]] * 0.5 / 9.81;
```

In[24]:=

```
ListLinePlot[filteredData, Frame → True, FrameLabel →
  {{y''[t], None}, {Row[{t, "(sec)"}], "Acceleration data (fraction of g)"}},
 RotateLabel → False, GridLines → Automatic, GridLinesStyle → LightGray]
```

Out[24]=



Convert to inches per second^2

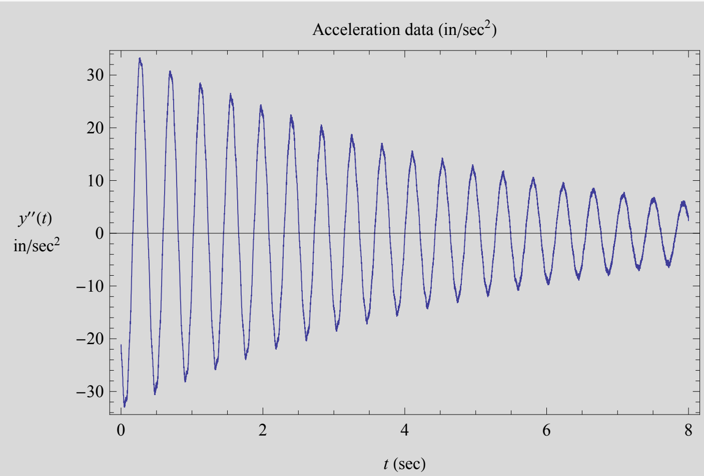
In[25]:=

```
filteredData[[All, 2]] = filteredData[[All, 2]] * 386 * 9.81;
```

In[26]:=

```
ListLinePlot[filteredData, Frame → True,
 FrameLabel → {{Column[{y''[t], " in/sec^2"}, Alignment → Center], None},
   {Row[{t, " (sec)"}], "Acceleration data (in/sec^2)"}},
 RotateLabel → False, GridLines → Automatic, GridLinesStyle → LightGray]
```

Out[26]=

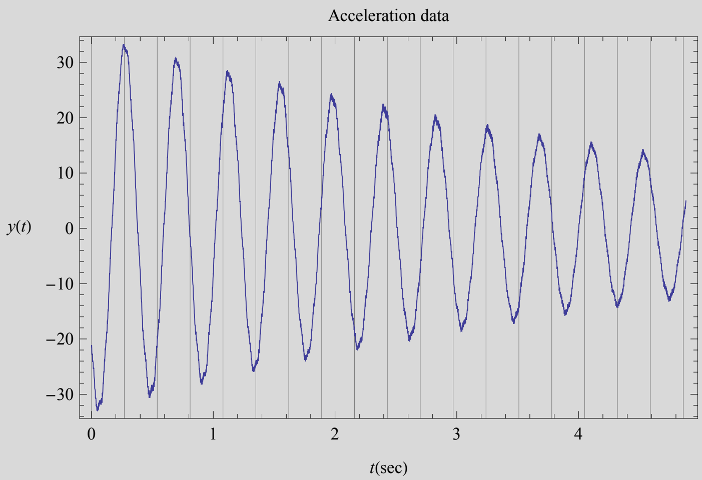


Plot few values (3 seconds)

In[27]:=

```
ListLinePlot[filteredData[[1 ;; 5000]], Frame → True,
 FrameLabel → {{y[t], None}, {Row[{t, "(sec)"}], "Acceleration data"}},
 RotateLabel → False, GridLines → {Range[0, 5, .27], None},
 GridLinesStyle → Gray, Axes → None]
```

Out[27]=



Find the natural frequency

From the above plot, Using the first 8 peaks, the period is **11*0.27=2.97/7 =0.424 seconds**. Hence the frequency is 2.35 Hz

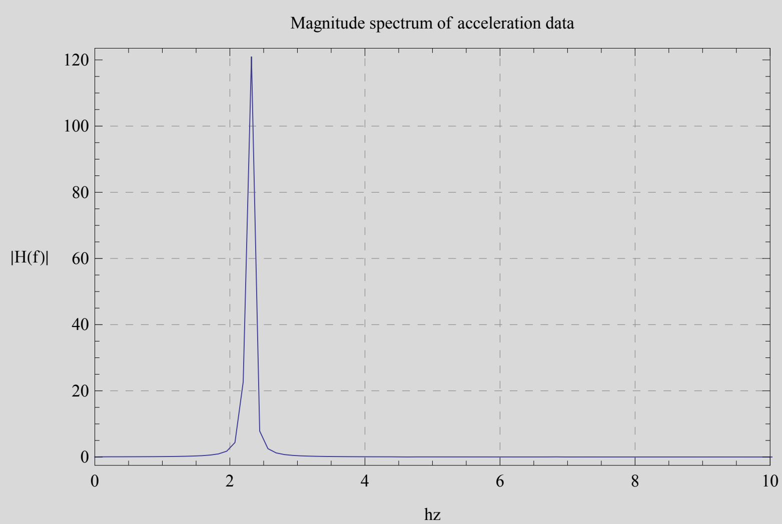
Finding the natural frequency using Fourier transform to obtain the spectrum

In[28]:=

```
py = Fourier[filteredData[[All, 2]] , FourierParameters → {1, -1}];
nSamples = Length[filteredData[[All, 2]]];
nUniquePts = Ceiling[(nSamples + 1) / 2];
py = py[[1 ;; nUniquePts]];
py = Abs[py];
py = py / nSamples;
py = py^2;

If[OddQ[nSamples], py[[2 ;; -1]] = 2 * py[[2 ;; -1]], py[[2 ;; -2]] = 2 * py[[2 ;; -2]]];
fs = 1000;
f = N[(Range[0, nUniquePts - 1] fs) / nSamples];
ListPlot[Transpose[{f, py}], Joined → True,
 FrameLabel → {{"|H(f)|", None}, {"hz", "Magnitude spectrum of acceleration data"}},
 ImageSize → 400, Frame → True, RotateLabel → False, GridLines → Automatic,
 GridLinesStyle → Dashed, PlotRange → {{0, 10}, All}]
```

Out[38]=

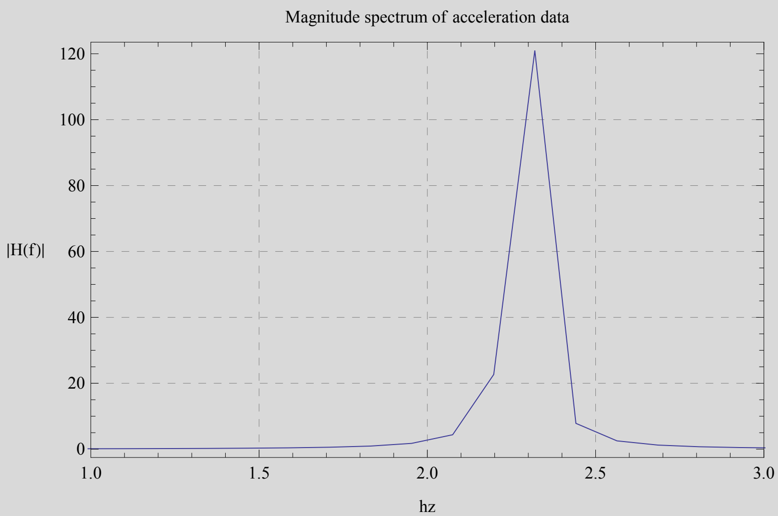


We see from the above that $f = 2.3$ cycles per second. Here is a zoom in view

In[39]:=

```
ListPlot[Transpose[{f, py}], Joined → True,
 FrameLabel → {{"|H(f)|", None}, {"hz", "Magnitude spectrum of acceleration data"}},
 ImageSize → 400, Frame → True, RotateLabel → False, GridLines → Automatic,
 GridLinesStyle → Dashed, PlotRange → {{1, 3}, All}]
```

Out[39]=



We see that the **above result matches that we obtained by counting the peaks from the plot directly**. But using the spectrum would be a better method to use.

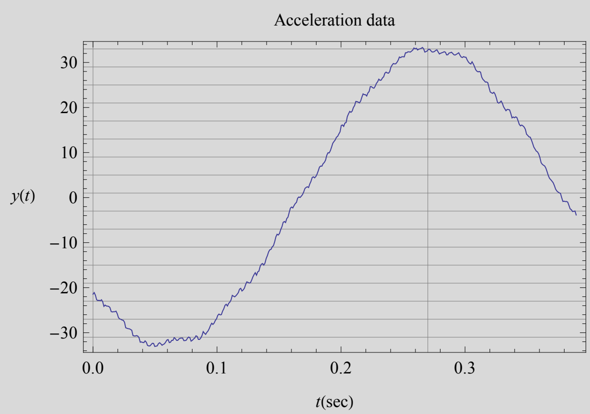
Finding the damping ζ

We first need to generate a list of say 10 peak values of y''(t) and the corresponding time. From the plot we see that the first positive peak is located at time 0.27 seconds. Hence we start from that point and look for a value at each sample point that is 1/f away from it. The data is available such that the separation in time between each data point is one milliseconds. First here is the plot showing the initial phase

In[40]:=

```
to = 400;
ListLinePlot[filteredData[[1 ;; to]], Frame → True,
 FrameLabel → {{y[t], None}, {Row[{t, "(sec)"}], "Acceleration data"}},
 RotateLabel → False, GridLines → {Range[0, to / 1000, .27], Range[-35, 35, 4]},
 GridLinesStyle → Gray, Axes → None, ImageSize → 300]
```

Out[41]=



Here is a list of the first 10 peaks

In[42]:=

```
period = 1 / 2.3;
initial = 0.27;
scale = 1000; (*one sample per millisecond*)
peaks = Table[ Flatten[
    {n + 1, Part[filteredData[[ Round[(initial + n * period) * scale]]]]}], {n, 0, 9}];
TableForm[peaks, TableHeadings → {None, {"peak #", "time", "peak"}}]
```

Out[46]//TableForm=

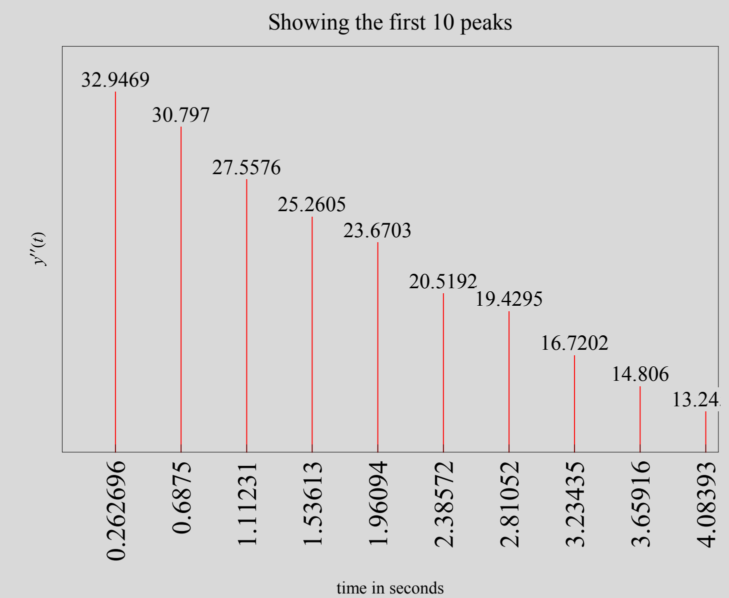
peak #	time	peak
1	0.262696	32.9469
2	0.6875	30.797
3	1.11231	27.5576
4	1.53613	25.2605
5	1.96094	23.6703
6	2.38572	20.5192
7	2.81052	19.4295
8	3.23435	16.7202
9	3.65916	14.806
10	4.08393	13.2452

Plot the above peaks to verify

In[47]:=

```
ticks = {{None, None}, {{#, Style[Rotate[#, 90 Degree], 14]} & /@ peaks[[ All, 2]], None}};
ListPlot[peaks[[ All, {2, 3}]], Filling → Axis, FillingStyle → Red,
 Frame → True, FrameTicks → ticks, ImageMargins → 30, Epilog → MapThread[
   Text[Style[#2, 11], {#1, #2}] &, {peaks[[ All, 2]], peaks[[ All, 3]]}], FrameLabel →
  {{y''[t], None}, {"time in seconds", Style["Showing the first 10 peaks", 12]}},
 PlotRange → {Automatic, {10, 35}}]
```

Out[48]=



Damping based on two successive peaks using the first formula

The formula to use here is $\ln\left(\frac{y_1}{y_2}\right) = 2\pi\,\xi$. Therefore, using the first 2 values we found above we obtain

In[49]:=

```
y1 = peaks[[ 1, 3]];
y2 = peaks[[ 2, 3]];
ξ = Log[y1/y2] / (2 π)
```

Out[51]=

```
0.0107394
```

Hence this shows that $\boxed{\xi = 1.074\ \%}$

Damping based on two successive peaks using the series expansion

The formula to use here is $\frac{y_1}{y_2} = 1 + 2\,\pi\,\xi$ therefore using the first 2 peaks we obtain

In[55]:=
```
y1 = peaks[[1, 3]];
y2 = peaks[[2, 3]];
ξ = 1/(2 π) (y1 - y2)/y2
```
Out[57]=
```
0.01111
```

This shows that $\boxed{\xi = 1.111\,\%}$

Damping based more than 2 successive peaks, using the final formula with an interval of “m” peaks

Here we use the formula $\frac{y_1}{y_{1+m}} = 1 + 2\,\pi\,m\,\zeta$ where m is a number we can change. Using $m = 5$ for example gives

In[58]:=
```
y1 = peaks[[1, 3]];
m = 5;
y2 = peaks[[1 + m, 3]];
ξ = 1/(2 m π) (y1/y2 - 1)
```
Out[61]=
```
0.0192788
```

Hence using $m = 5$ gives $\boxed{\xi = 1.93\,\%}$

Trying for $m = 9$ gives

In[62]:=
```
y1 = peaks[[1, 3]];
m = 9;
y2 = peaks[[1 + m, 3]];
ξ = (y1/y2 - 1)/(2 m π)
```
Out[65]=
```
0.0263042
```

Hence using $m = 9$ gives $\boxed{\xi = 2.63\,\%}$

Finding number of cycles to have the amplitude decay by 1 / 2

Using $\ln(2) = 2\, m\, \pi\, \zeta\, \frac{\omega}{\omega_d}$ we can estimate m the number of cycles for the amplitude to decay by half. We use ξ=0.0107394 from above since that is the ξ value found from the same formula. Hence

In[66]:=
```
ξ = 0.0107394;
m = Log[2] Sqrt[1 - ξ^2] / (2 π ξ)
```

Out[67]=
```
10.2717
```

This shows that it takes 10 cycles for the amplitude to decay by half. Looking again at the plots, this is verified

Applet to analyze the data allowing different formulas to be selected and different values for M

This is a small applet to help analyze this data. It allows you to select the formula to determine ξ and also select m for the final formula. For each formula used, the corresponding value of number of cycles for the first peak to decay by half is computed.

```
Manipulate[
 Module[{dataPlot},

  dataPlot = ListPlot[
    filteredData[[1 ;; tscale]],
    Joined → True,
    Frame → True,
    FrameLabel → {
      {None, None}, {Row[{t, " (sec)"}], "Acceleration data y''(t)(in/sec^2)"}},
    GridLines → Automatic,
    GridLinesStyle → LightGray,
    ImageSize → {250},
    ImageMargins → 0,
    ImagePadding → {{20, 5}, {40, 20}}];

  Grid[{
    {Row[{ξ, " = ", padIt2[100 * findZeta[formula, mm], {5, 4}], " %"}],
     Row[{"frequency ", " = ", 2.3, " Hz"}]},
    {dataPlot, spectrum},
    {peaksPlot, tbl}
   }, Frame → All, Alignment → Center]
 ],

 Grid[
  {
   {
```

```
    Row[{Style["m ", 12],
      Manipulator[Dynamic[mm, {mm = #} &], {1, 8, 1}, ImageSize → Tiny],
      Style[Dynamic@padIt2[mm, 1], 11]
     }], SpanFromLeft
   },

   {
    Row[{Style["ξ formula", 11],
      PopupMenu[Dynamic[formula, {formula = #} &],
       {
        1 → Row[{Style["first method ", Bold],
            Style[TraditionalForm[Log[y_m/y_(m+1)] == 2 ξ π], 10]}],
        2 → Row[{Style["series method ", Bold], Style[
             TraditionalForm[y_m/y_(m+1) == 1 + 2 ξ π], 10]}],
        3 → Row[{Style["m method ", Bold], Style[TraditionalForm[
              y_1/y_(1+m) == 1 + 2 ξ m π], 10]}],
       }
       , ImageSize -> All
      ]
     }
    ],
    ,
    Row[{Style["time scale ", 12],
      Manipulator[Dynamic[tscale, {tscale = #} &],
       {1, 8192, 1}, ImageSize → Tiny], Spacer[5],
      Style[Dynamic@padIt2[tscale, 4], 11], Spacer[5], "ms"
     }]
   }
  }, Alignment → Left
 ],

 {{mm, 1}, None},
 {{formula, 3}, None},
 {{tscale, 4000}, None},
 SynchronousUpdating → True,
 ControlPlacement → Top,
 Alignment → Center,
 SynchronousInitialization → True,
 ContinuousAction → True,
 AutorunSequencing → Automatic,
 TrackedSymbols → {mm, formula, tscale},
```

```
Initialization :> {

  SetDirectory[NotebookDirectory[]];
  data = Import["free_vibr.txt", "Elements"];
  data = StringSplit[Import["free_vibr.txt", "Lines"]];

  filteredData =
   Transpose[{ToExpression[Part[StringSplit[#, ":"], 3]] & /@ data[[All, 2]],
     Internal`StringToDouble[#] & /@ data[[All, 3]]}];

  filteredData[[All, 1]] = filteredData[[All, 1]] - filteredData[[1, 1]];
  mean = Mean[filteredData[[All, 2]]];
  filteredData[[All, 2]] = filteredData[[All, 2]] - mean;
  filteredData[[All, 2]] = filteredData[[All, 2]] * 0.5 / 9.81;
  filteredData[[All, 2]] = filteredData[[All, 2]] * 386 * 9.81;

  py = Fourier[filteredData[[All, 2]] , FourierParameters → {1, -1}];
  nSamples = Length[filteredData[[All, 2]]];
  nUniquePts = Ceiling[(nSamples + 1) / 2];
  py = py[[1 ;; nUniquePts]];
  py = Abs[py];
  py = py / nSamples;
  py = py^2;

  If[OddQ[nSamples],
   py[[2 ;; -1]] = 2 * py[[2 ;; -1]], py[[2 ;; -2]] = 2 * py[[2 ;; -2]]];
  fs = 1000;
  f = N[(Range[0, nUniquePts - 1] fs) / nSamples];

  spectrum = ListPlot[Transpose[{f, py}], Joined → True, FrameLabel →
     {{"|H(f)|", None}, {"hz", "Magnitude spectrum of acceleration data"}},
    Frame → True, RotateLabel → False, GridLines → Automatic, GridLinesStyle → Dashed,
    PlotRange → {{0, 10}, All}, ImageSize → {250}, ImageMargins → 0];


  period = 1 / 2.3;
  initial = 0.27;
  scale = 1000;  (*one sample per millisecond*)
  peaks = Table[  Flatten[
     {n + 1, Part[filteredData[[ Round[(initial + n * period) * scale]]]]}], {n, 0, 9}];
  tbl = TableForm[peaks[[1 ;; 10]], TableHeadings → {None, {"#", "time", "peak"}}];

  ticks = {{None, None}, {{#, Style[Rotate[padIt2[#, {3, 2}], 90 Degree], 14]} & /@
      peaks[[ 1 ;; 10, 2]], None}};
  peaksPlot = ListPlot[peaks[[ 1 ;; 10, {2, 3}]],
    Filling → Axis,
    FillingStyle → Red,
    Frame → True,
    FrameTicks → ticks,
    ImageMargins → 0,
    Epilog → MapThread[Text[Style[padIt2[#2, {3, 1}], 10], {#1, #2}, {0, -1}] &,
      {peaks[[ 1 ;; 10, 2]], peaks[[ 1 ;; 10, 3]]}], FrameLabel →
```

```
    {{y''[t], None}, {"time in seconds", Style["Showing the first 10 peaks", 12]}},
   PlotRange → {Automatic, {10, 39}},
   ImageSize → {250},
   ImagePadding → {{20, 5}, {50, 20}}];

 (*-------------------------------------------*)
 padIt1[v_, f_List] := AccountingForm[Chop[v],
   f, NumberSigns → {"-", "+"}, NumberPadding → {"0", "0"}, SignPadding → True];
 (*-------------------------------------------*)
 padIt2[v_, f_List] := AccountingForm[Chop[v],
   f, NumberSigns → {"", ""}, NumberPadding → {"0", "0"}, SignPadding → True];
 (*-------------------------------------------*)
 padIt2[v_, f_Integer] := AccountingForm[Chop[v],
   f, NumberSigns → {"", ""}, NumberPadding → {"0", "0"}, SignPadding → True];

 findZeta[formula_, m_] := Module[{y1, y2},
   Which[formula == 1,
    y1 = peaks[[m, 3]];
    y2 = peaks[[m + 1, 3]];
    Log[y1/y2]/(2 π),
    formula == 2,
    y1 = peaks[[m, 3]];
    y2 = peaks[[m + 1, 3]];
    1/(2 π) (y1 - y2)/y2,
    formula == 3,
    y1 = peaks[[1, 3]];
    y2 = peaks[[1 + m, 3]];
    1/(2 m π) (y1/y2 - 1)
   ]
  ];

}
]
```

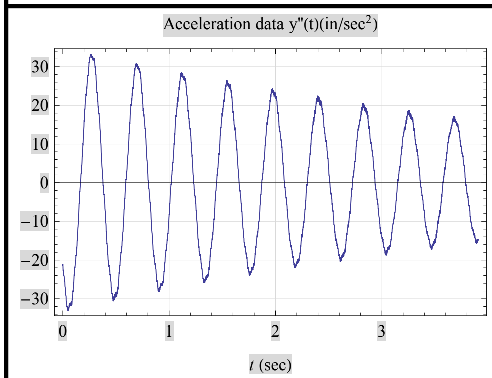
Out[68]=

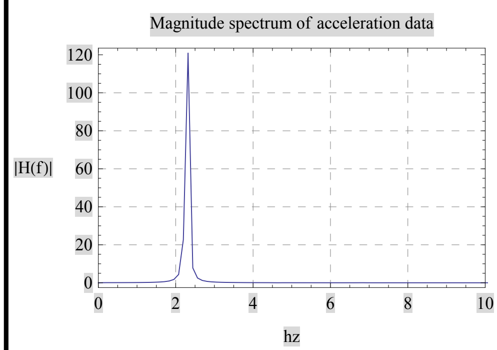
m 1

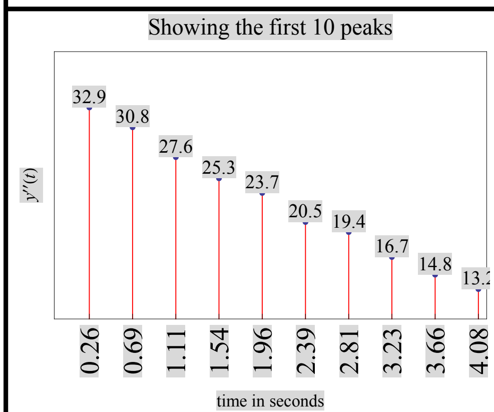
ξ formula **m method** $\frac{y_1}{y_{11.2717}} = 1.69311$ time scale 4000 ms

0.0107394 = 1.1110 %

frequency = 2.3 Hz







#	time	peak
1	0.262696	32.9469
2	0.6875	30.797
3	1.11231	27.5576
4	1.53613	25.2605
5	1.96094	23.6703
6	2.38572	20.5192
7	2.81052	19.4295
8	3.23435	16.7202
9	3.65916	14.806
10	4.08393	13.2452