

## 5 - calculating linear weights

```
In[1]:= c1 = (1/Sqrt[2])
```

```
c2 = (-1/Sqrt[2])
```

```
Out[1]= { {1/Sqrt[2]}, {1/Sqrt[2]} }
```

```
Out[2]= { {1/Sqrt[2]}, {-1/Sqrt[2]} }
```

```
In[3]:= q = (2/3)
```

```
Out[3]= {{2}, {3}}
```

## 6 - orthonormal

```
In[4]:= Vector[S_, Q_] := (1/Sqrt[S]) (Table[E^{I*2*Pi*i/(N-1)}, {N, 0, S-1}])
```

```
In[5]:= i = 0;
```

```
k = 1;
```

```
vi = Vector[8, i]
```

```
vk = Vector[8, k]
```

```
Out[7]= {1/(2 Sqrt[2]), 1/(2 Sqrt[2])}
```

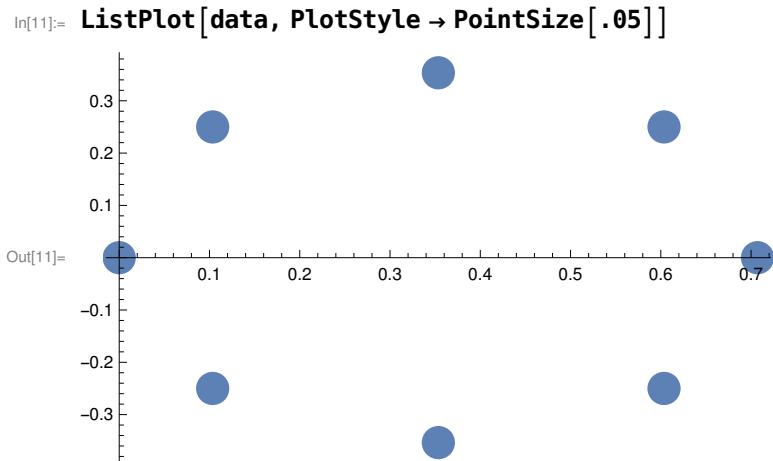
```
Out[8]= {E^(-I \pi /4)/Sqrt[2], 1/(2 Sqrt[2]), E^(I \pi /4)/Sqrt[2], I/(2 Sqrt[2]), E^(3 I \pi /4)/Sqrt[2], -1/(2 Sqrt[2]), E^(-3 I \pi /4)/Sqrt[2], -I/(2 Sqrt[2])}
```

```
In[9]:= sum = (#1 + #2) &[vi, vk]
```

```
Out[9]= {1/(2 Sqrt[2]) + E^(-I \pi /4)/Sqrt[2], 1/Sqrt[2], 1/(2 Sqrt[2]) + E^(I \pi /4)/Sqrt[2], 1/(2 Sqrt[2]) + I/(2 Sqrt[2]), 1/Sqrt[2] + E^(3 I \pi /4)/Sqrt[2], 0, 1/(2 Sqrt[2]) + E^(-3 I \pi /4)/Sqrt[2], 1/(2 Sqrt[2]) - I/(2 Sqrt[2])}
```

```
In[10]:= data = Transpose@{Re[sum], Im[sum]}
```

```
Out[10]= {{1/4 + 1/(2 Sqrt[2]), -1/4}, {1/Sqrt[2], 0}, {1/4 + 1/(2 Sqrt[2]), 1/4}, {1/(2 Sqrt[2]), 1/(2 Sqrt[2])}, {-1/4 + 1/(2 Sqrt[2]), 1/4}, {0, 0}, {-1/4 + 1/(2 Sqrt[2]), -1/4}, {1/(2 Sqrt[2]), -1/(2 Sqrt[2])}}
```



In[12]:= `N@Total[sum]`

Out[12]=  $2.82843 + 0. \text{i}$

In[13]:= `vi = Transpose[{vi}];  
vk = Transpose[{vk}];`

In[15]:= `Chop[N@ConjugateTranspose[vi].vk]`

Out[15]=  $\{\{0\}\}$

## 8 - dotting a unit vector with itself is 1

In[16]:= `vector = {1 2 34 5 6}`

Out[16]=  $\{\{1, 2, 34, 5, 6\}\}$

In[17]:= `vector = vector / Norm[vector]`

Out[17]=  $\left\{\left\{\frac{1}{\sqrt{1222}}, \sqrt{\frac{2}{611}}, 17 \sqrt{\frac{2}{611}}, \frac{5}{\sqrt{1222}}, 3 \sqrt{\frac{2}{611}}\right\}\right\}$

In[18]:= `vector.Transpose@vector`

Out[18]=  $\{\{1\}\}$

## 11 - calculate the 64 point dft of a cos of frequency pi/4 radians

In[19]:= `numberSamples = 64;  
numberRange = Range[0, numberSamples - 1];  
x = Cos[Pi/4 * numberRange];  
DFT = FourierMatrix[numberSamples];`

In[23]:= **xDFT = DFT.x**

$$\left\{ \theta, \dots 62 \dots, \frac{\frac{i\pi}{32}}{8\sqrt{2}} + \frac{\frac{i\pi}{32}}{8\sqrt{2}} - \frac{\frac{3i\pi}{32}}{8\sqrt{2}} - \frac{\frac{3i\pi}{32}}{8\sqrt{2}} - \frac{1}{8} e^{-\frac{i\pi}{8}} - \frac{1}{8} e^{\frac{i\pi}{8}} - \frac{\frac{5i\pi}{32}}{8\sqrt{2}} + \frac{\frac{5i\pi}{32}}{8\sqrt{2}} + \frac{\frac{7i\pi}{32}}{8\sqrt{2}} + \frac{\frac{7i\pi}{32}}{8\sqrt{2}} + \right.$$

$$\frac{1}{8} e^{-\frac{i\pi}{4}} + \frac{1}{8} e^{\frac{i\pi}{4}} + \frac{\frac{9i\pi}{32}}{8\sqrt{2}} + \frac{\frac{9i\pi}{32}}{8\sqrt{2}} - \frac{\frac{11i\pi}{32}}{8\sqrt{2}} - \frac{\frac{11i\pi}{32}}{8\sqrt{2}} - \frac{1}{8} e^{-\frac{3i\pi}{8}} - \frac{1}{8} e^{\frac{3i\pi}{8}} - \frac{\frac{13i\pi}{32}}{8\sqrt{2}} + \frac{\frac{13i\pi}{32}}{8\sqrt{2}} + \frac{\frac{15i\pi}{32}}{8\sqrt{2}} + \frac{\frac{15i\pi}{32}}{8\sqrt{2}} +$$

$$\frac{\frac{17i\pi}{32}}{8\sqrt{2}} + \frac{\frac{17i\pi}{32}}{8\sqrt{2}} - \frac{\frac{19i\pi}{32}}{8\sqrt{2}} - \frac{\frac{19i\pi}{32}}{8\sqrt{2}} - \frac{1}{8} e^{-\frac{5i\pi}{8}} - \frac{1}{8} e^{\frac{5i\pi}{8}} - \frac{\frac{21i\pi}{32}}{8\sqrt{2}} - \frac{\frac{21i\pi}{32}}{8\sqrt{2}} + \frac{\frac{23i\pi}{32}}{8\sqrt{2}} + \frac{\frac{23i\pi}{32}}{8\sqrt{2}} + \frac{1}{8} e^{-\frac{3i\pi}{4}} +$$

$$\left. \frac{1}{8} e^{\frac{3i\pi}{4}} + \frac{\frac{25i\pi}{32}}{8\sqrt{2}} + \frac{\frac{25i\pi}{32}}{8\sqrt{2}} - \frac{\frac{27i\pi}{32}}{8\sqrt{2}} - \frac{\frac{27i\pi}{32}}{8\sqrt{2}} - \frac{1}{8} e^{-\frac{7i\pi}{8}} - \frac{1}{8} e^{\frac{7i\pi}{8}} - \frac{\frac{29i\pi}{32}}{8\sqrt{2}} - \frac{\frac{29i\pi}{32}}{8\sqrt{2}} + \frac{\frac{31i\pi}{32}}{8\sqrt{2}} + \frac{\frac{31i\pi}{32}}{8\sqrt{2}} \right\}$$

[large output](#)

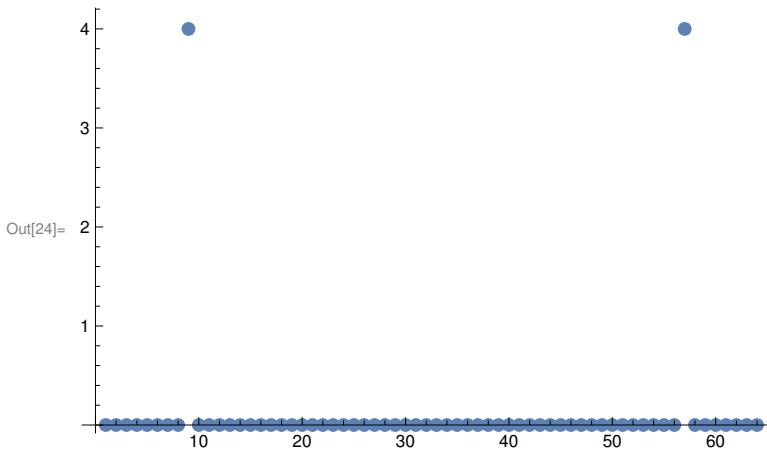
[show less](#)

[show more](#)

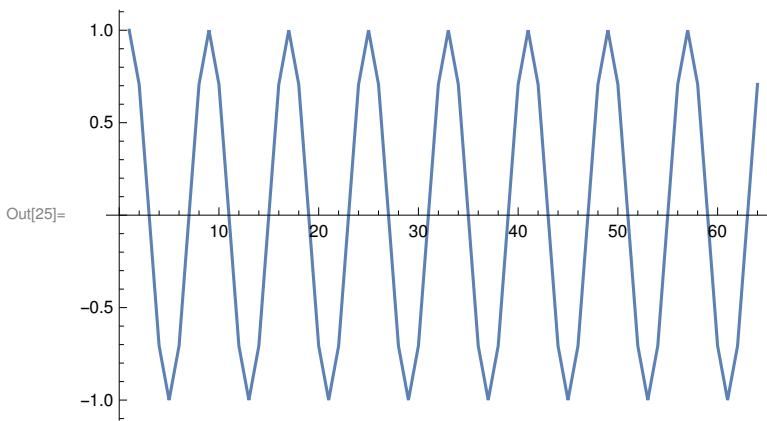
[show all](#)

[set size limit...](#)

In[24]:= **ListPlot[xDFT, PlotStyle → PointSize[.02]]**



In[25]:= **ListLinePlot[x]**



In[26]:=

## 13 - low pass using fourier and inverse fourier

```
In[27]:= lowpassFourierFilter[data_] := Module[{fWeights, lWeights},
  fWeights = Fourier[data];
  lWeights = Join[fWeights[[;; Floor[Length[data]/4]]],
    .1 * fWeights[[Floor[Length[data]/4] + 1 ;;]]];
  Return[Re[Chop[InverseFourier[lWeights]]]]];
In[28]:= Fs = 8192
data = Flatten[Transpose[Import[
  "/home/nathan/olin/fall2016/QEAFall2016Homework/bset1/matlab/handel.csv"]]];
Out[28]= 8192
```

highest frequency:

```
In[30]:= Fs/2
Out[30]= 4096
```

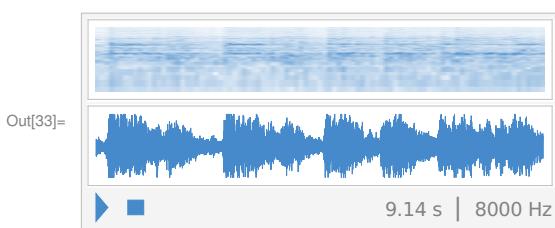
lowest frequency

```
In[31]:= 0
Out[31]= 0
```

```
In[32]:= dataFiltered = lowpassFourierFilter[data]
Out[32]= {-0.00538123, -0.0177226, -0.025308, -0.0160855, 0.000450921, 0.0183174,
  0.0161275, -0.00682954, -0.026557, -0.0399389, -0.0572852, -0.08448,
  -0.106186, -0.0919217, -0.0421564, -0.0118159, -0.0191307, -0.0312334,
  ... 73 077 ..., 0.0838113, 0.0722977, 0.0473331, 0.0359092, 0.0480158,
  0.0165222, -0.0353874, -0.0828198, -0.103739, -0.0815486, -0.0313233,
  0.0222903, 0.0677911, 0.0880384, 0.0959882, 0.105543, 0.0756967, 0.0241307}
```

[large output](#) [show less](#) [show more](#) [show all](#) [set size limit...](#)

```
In[33]:= ListPlay[dataFiltered]
```



## 14 - filter hum

```
In[34]:= guitarData = Import[
  "/home/nathan/olin/fall2016/QEAFall2016Homework/bset1/matlab/guitar1_60Hz_hum.wav",
  "Data"]
ListPlay[guitarData, SampleRate → 44100]
```

Out[34]=

```
{-0.0154724, -0.0160522, -0.0166626, -0.0168457, -0.0168762,
-0.0173645, -0.0177917, -0.0185242, -0.0195618, -0.0202026, -0.0209045,
-0.0213623, -0.0210266, -0.0201416, -0.0189819, -0.0175171, -0.015625,
-0.0146484, -0.0148621, -0.0149536, ..., 83 751 ..., 0.0338145, 0.0344249,
0.0414747, 0.0433668, 0.0404065, 0.0424207, 0.0491043, 0.0540483,
0.0586261, 0.0653706, 0.0737632, 0.0828883, 0.084994, 0.0780969,
0.073336, 0.0682699, 0.0531022, 0.0375683, 0.0326548, 0.0314951}
```

large output

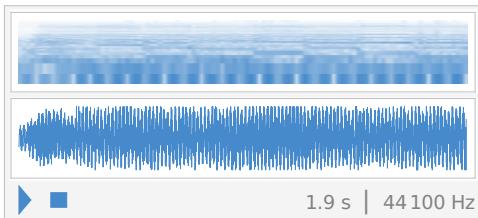
show less

show more

show all

set size limit...

Out[35]=



```
In[36]:= guitarDataFiltered = lowpassFourierFilter[guitarData]
ListPlay[guitarData, SampleRate → 44100]
```

Out[36]=

```
{-0.00277813, -0.0103082, -0.0115646, -0.00897426, -0.00820755,
-0.0101797, -0.0112135, -0.0102478, -0.0100633, -0.0116529, -0.0126471,
-0.0118344, -0.0111206, -0.0115651, -0.0114154, -0.00973719, -0.00832838,
-0.00851535, ..., 83 755 ..., 0.0216298, 0.0225205, 0.0227415, 0.0237073,
0.0260745, 0.029039, 0.0321897, 0.0359821, 0.040538, 0.0446006, 0.0458807,
0.043966, 0.0404823, 0.0354255, 0.0286758, 0.0227108, 0.0181235, 0.0113857}
```

large output

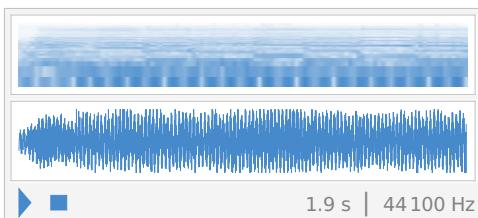
show less

show more

show all

set size limit...

Out[37]=



```
In[38]:= data = Re[Fourier[guitarDataFiltered]]
```

```
{-0.192972, -0.0099005, 0.00241204, -0.00854094, -0.0119331, -0.011744,  
-0.0153276, -0.0213279, -0.0102596, -0.0172746, -0.00750142, -0.0010313,  
-0.0102605, -0.0190867, -0.00774538, -0.00828969, -0.0171675, 0.000548939,  
... 83 756 ..., 0.000548939, -0.0171675, -0.00828969, -0.00774538, -0.0190867,  
-0.0102605, -0.0010313, -0.00750142, -0.0172746, -0.0102596, -0.0213279,  
-0.0153276, -0.011744, -0.0119331, -0.00854094, 0.00241204, -0.0099005}
```

large output

show less

show more

show all

set size limit...

```
In[39]:= Length[data]
```

```
Out[39]= 83 791
```

```
In[40]:= ListPlot[data[[Floor[83 791/2] ;;]], PlotRange -> All]
```

