BSet 3: The Power of LTI

Problem 1

Calculate the impulse response of a system that
calculates the three element trailing average of the input :

$y[n] = \frac{1}{3}(x[n] + x[n-1] + x[n-2])$

In other words, what does the output of the system look like if the input is x[n] = δ[n] –
i.e., a x = 1 at n = 0 and 0 everywhere else?

$$h[n] = \frac{1}{3}(\text{d}[n]+x[n-1]+x[n-2])$$

```
δ[n_] := 0;
δ[0] := 1;
```

$$h[n\_] := \frac{1}{3}\delta[n] + \frac{1}{3}\delta[n-1] + \frac{1}{3}\delta[n-2];$$

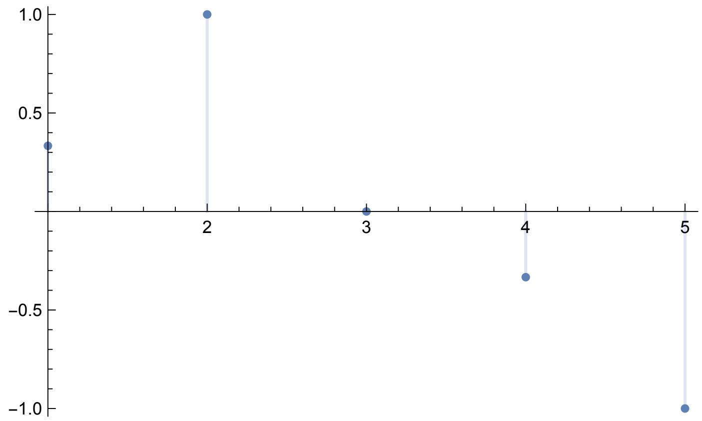
```
Table[h[n], {n, 0, 5}]
```

$$\left\{\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, 0, 0, 0\right\}$$

Problem 2: Given the x signal below, apply the filter to the signal

```
x = {1, 2, -3, 0, 0};
```

```
trailing = {1/3, 1/3, 1/3};
processedBad = Join[x[[1]] * trailing, {0, 0}] +
   Join[{0}, x[[2]] * trailing, {0}] + Join[{0, 0}, x[[3]] * trailing];
DiscretePlot[processedBad[[n]], {n, 1, 5}]
```



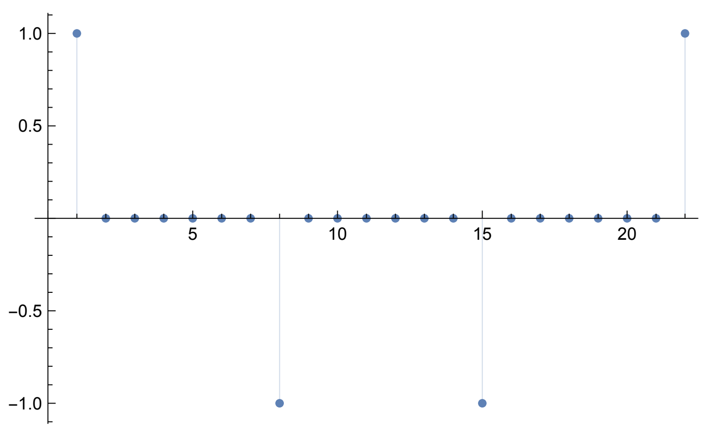
Impulse responses and convolution

Problem 3

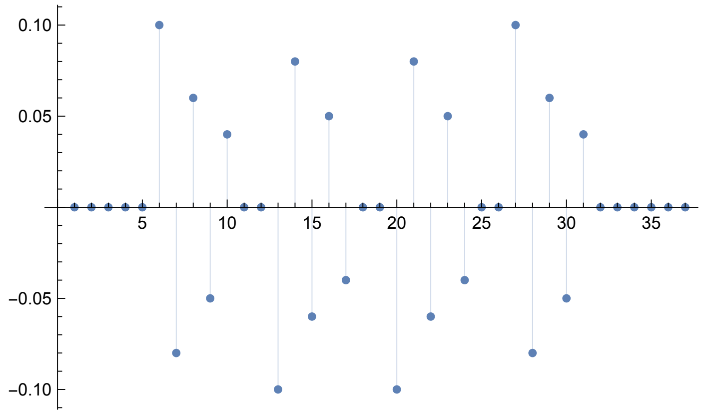
(a) Sketch output y[n]

```
ethernetX = {1, 0, 0, 0, 0, 0, 0, -1, 0, 0, 0, 0, 0, 0, -1, 0, 0, 0, 0, 0, 0, 1};
ethernetResponse = {0.1, -0.08, 0.06, -0.05, 0.04, 0.0};
```

```
ListPlot[ethernetX, Filling → Axis]
```



```
ListPlot[ListConvolve[
  ethernetResponse,
  Join[
   ConstantArray[0, 10],
   ethernetX,
   ConstantArray[0, 10]
  ]
 ], Filling -> Axis]
```



(b) What potential problems can you anticipate if the time between the ± 1 samples is too short?

It looks like the bits will interfere with each other

4. Slash playing guitar in a stairwell

(a) Explain why the recording of a finger being snapped is a decent approximation to the impulse response of this system

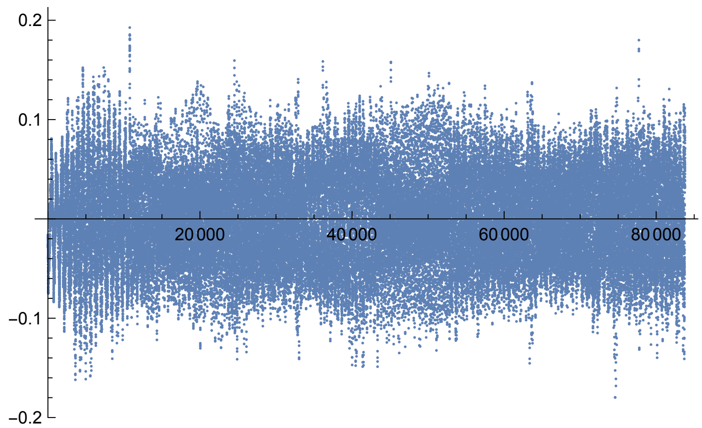
A finger being snapped is a very sharp, loud noise, and the response of the echos of the stairwell will be the response to the short snap sound.

(b) Using what you know about the convolution and impulse responses simulate the effect of the guitar sample being played in a stairwell.

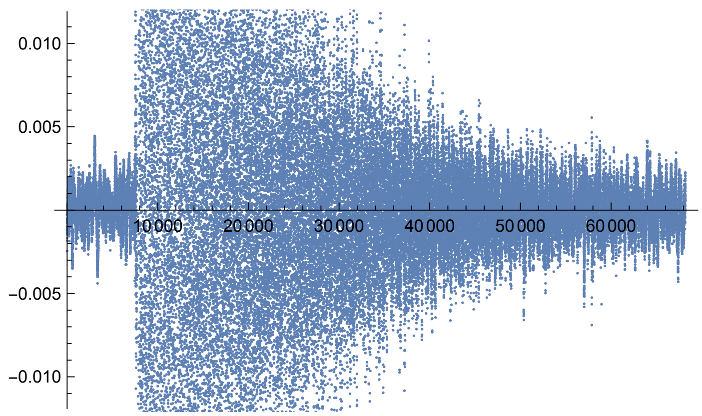
In[235]:=
```
SetDirectory[NotebookDirectory[]];
{fs, hStairwell, x} = Import["BSet3Files/guitar_stairwell.mat"];
fs = fs[[1, 1]];
x = Flatten[x];
hStairwell = Flatten[hStairwell];
```

In[240]:=
```
ListPlot[x]
ListPlot[hStairwell]
```

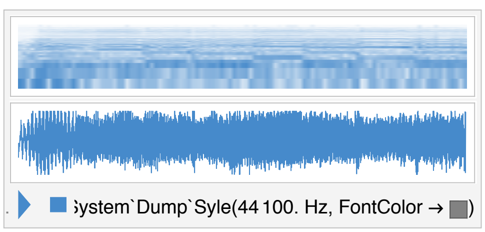
Out[240]=



Out[241]=

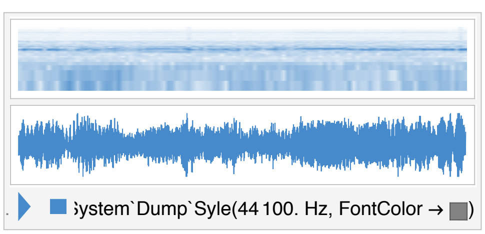



```
ListPlay[x, SampleRate → fs]
```

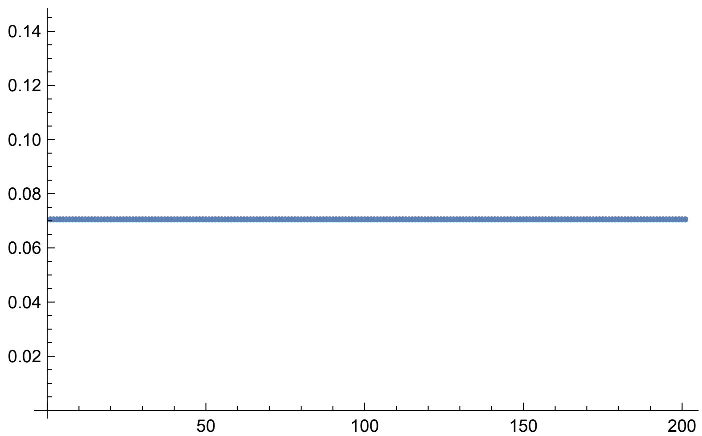



```
guitarInStairwell = ListConvolve[hStairwell, x, 1];
```

```
ListPlay[guitarInStairwell, SampleRate → fs]
```



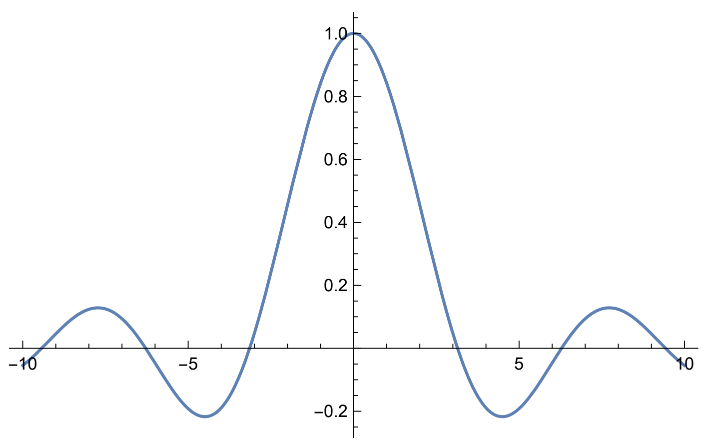
```
(* This is really cool and it's because of sinc *)
ListPlot[Abs[Fourier[Join[
    ConstantArray[0, 100],
    {1},
    ConstantArray[0, 100]
   ]]]]
```




```
sincOrSomething = Integrate[Exp[I * ω * t], ω]
```

$$-\frac{i\, e^{i\, t\, \omega}}{t}$$

```
Plot[Re[sincOrSomething /. {ω → 1}], {t, -10, 10}]
```



5.

In[114]:=
```
Ωc = Pi / 4;

q5$lowPass = Sin[Ωc * n] / (n * π);
q5$lowPass // TraditionalForm
```

Out[116]//TraditionalForm=

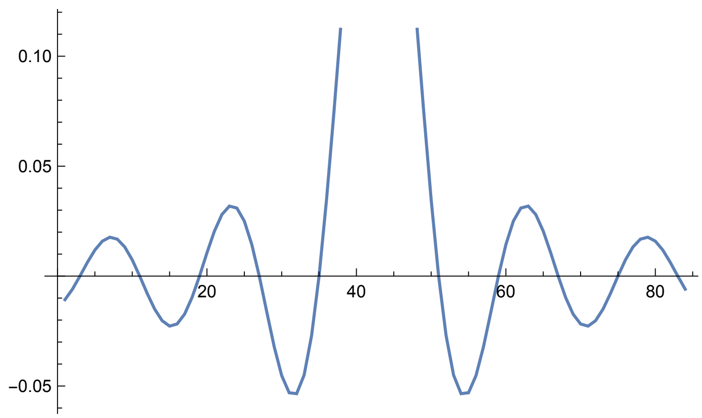
$$\frac{\sin\left(\frac{\pi n}{4}\right)}{\pi n}$$

In[6]:=
```
SetDirectory[NotebookDirectory[]];
q5$handel = Import["BSet2Files/handel.wav"];
```

(a)

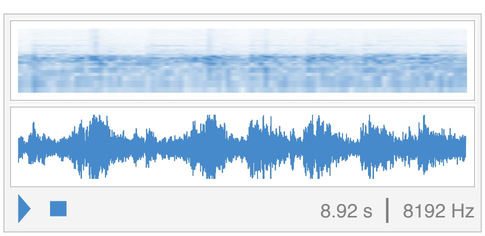
In[129]:=
```
q5$n = Range[-42, 41];
q5$h = Ωc / Pi * Sinc[(Ωc * n / Pi) * (Pi / 2)] /. n → q5$n;
ListLinePlot[q5$h]
```

Out[131]=



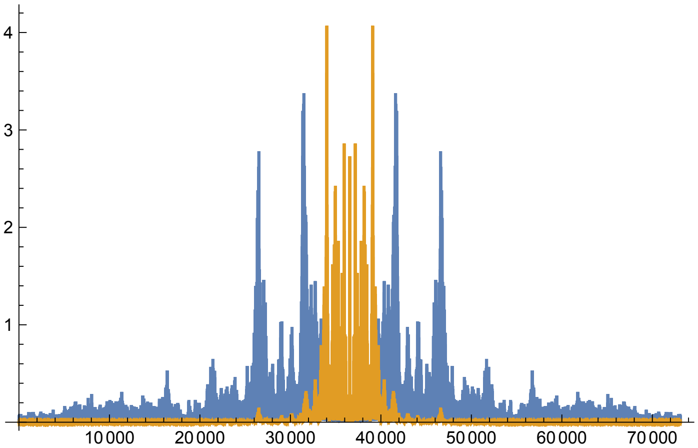
In[189]:=
```
q5$x = q5$handel // AudioData;
q5$y = ListConvolve[q5$h, q5$x, 1];
q5a$lowPassed =
 ListPlay[q5$y, SampleRate → QuantityMagnitude[AudioSampleRate[q5$handel]]]
```

Out[191]=



In[93]:=
```
fftShift[x_] := RotateRight[Fourier[x], Floor[Length[x] / 2]];
ListLinePlot[{fftShift[q5$x] // Abs, fftShift[q5$y] // Abs}, PlotRange → All]
```

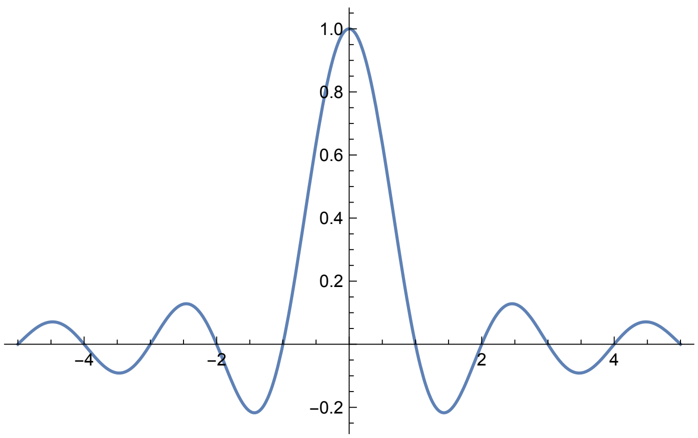
Out[94]=



(b)

In[134]:=
```
Plot[Sinc[d * Pi], {d, -5, 5}, PlotRange → Full]
```

Out[134]=

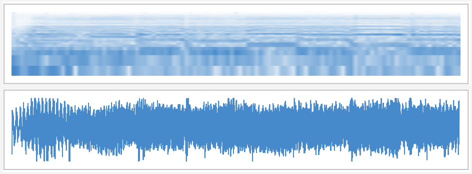


Matlab uses a sinc function that crosses zero at x mod 1 = 1, dang.

In[256]:=
```
q5b$Ωc = Pi / 4;
q5$hHighPass = ((-q5b$Ωc / Pi) * Sinc[(q5b$Ωc * n / Pi) * (Pi / 2)] /. n → q5$n) +
    (If[# == 0, 1, 0] & /@ q5$n);
```

In[258]:=
```
q5b$y = ListConvolve[q5b$hHighPass, x, 1];

q5b$highPassed = ListPlay[q5b$y, SampleRate → fs]
```

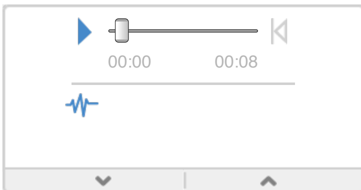
Out[259]=



System`Dump`Syle(44 100. Hz, FontColor →)

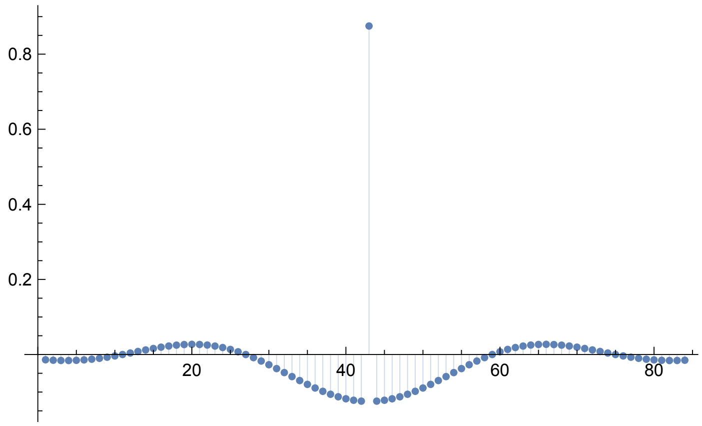
In[201]:=
```
q5$handel
```

Out[201]=

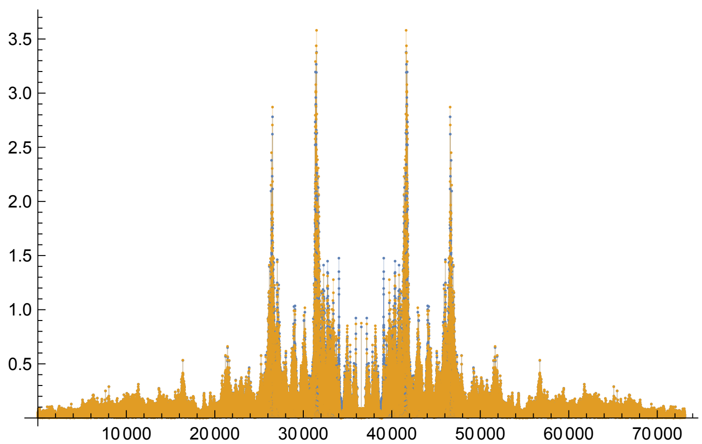


In[212]:=
```
ListPlot[q5$hHighPass, PlotRange → Full, Filling → Axis]
ListPlot[{
  fftShift[q5$x] // Abs,
  fftShift[q5b$y] // Abs
 }, PlotRange → All, Filling → Axis]
```

Out[212]=

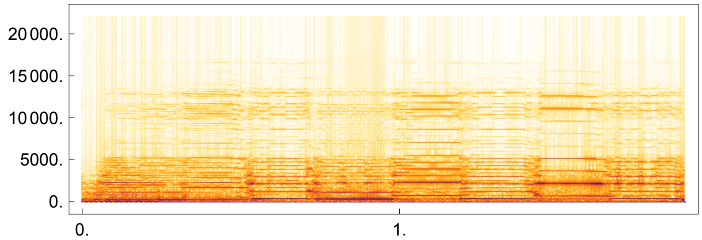


Out[213]=

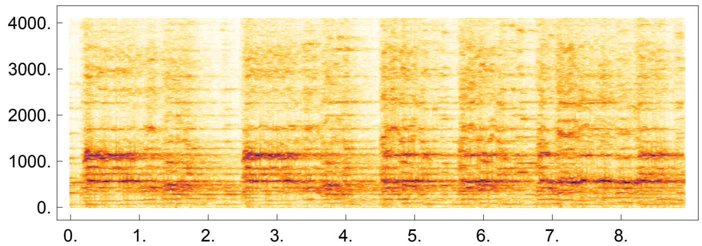


In[260]:=
```
Spectrogram[q5b$highPassed]
Spectrogram[q5$handel]
Spectrogram[q5a$lowPassed]
```

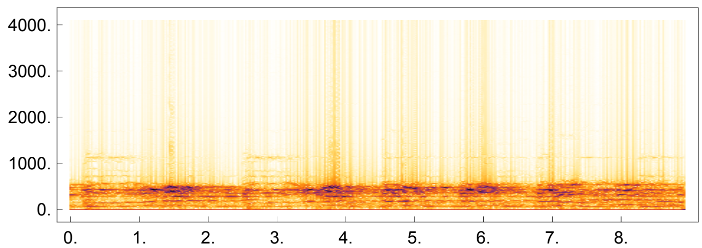
Out[260]=



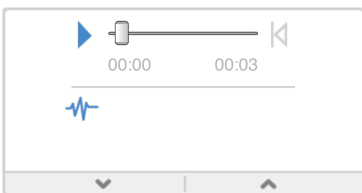
Out[261]=

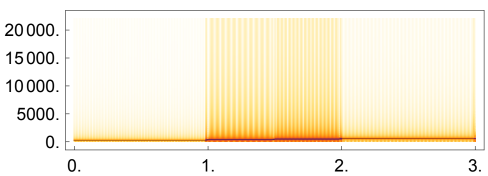


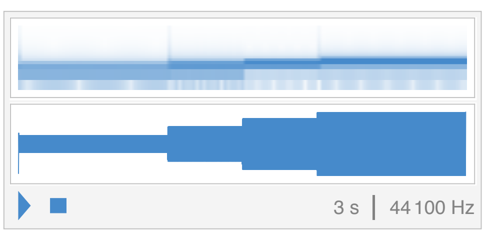
Out[262]=

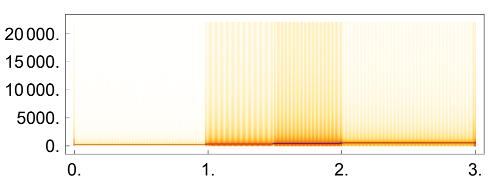


In[286]:=
```
ts = TimeSeries[{{0, 300}, {1, 400}, {1.5, 500}, {2, 600}}];
stuff = AudioGenerator[{"Sin", ts}, 3]
Spectrogram[stuff]

stuff$Ωc = Pi / 64;
stuff$n = Range[-42, 41];
stuff$hHighPass =
  ((-stuff$Ωc / Pi) * Sinc[(stuff$Ωc * n / Pi) * (Pi / 2)] /. n → stuff$n) +
   (If[# == 0, 1, 0] & /@ q5$n);

stuff$y = ListConvolve[stuff$hHighPass, AudioData[stuff], 1];

stuff$highPassed =
 ListPlay[stuff$y, SampleRate → QuantityMagnitude[AudioSampleRate[stuff]]]
Spectrogram[stuff$highPassed]
```

Out[287]=


Out[288]=


Out[293]=


Out[294]=


6.