

Removing Echo from Audio using Cepstrum Analysis

▼ Introduction

This application will

- import an audio file that has an echo
- employ the cepstrum function to identify the start of the echo
- use this information to generate and apply an IIR filter to remove the echo
- and write the de-echoed audio back to a sound file

This speaker component is needed to play audio:



```
> restart:
with(SignalProcessing):
with(AudioTools):
with(DocumentTools):
with(ColorTools):
with(plots):
```

▼ Import and Plot Waveform of Audio File

Import audio and extract sample rate

```
> origAudio := Read("maplesim_echo.wav"):
   Fs := attributes(origAudio)[1];
                                   Fs := 11025 (2.1)
```

Let's play the audio - you'll notice an echo that starts at around 0.5 seconds.

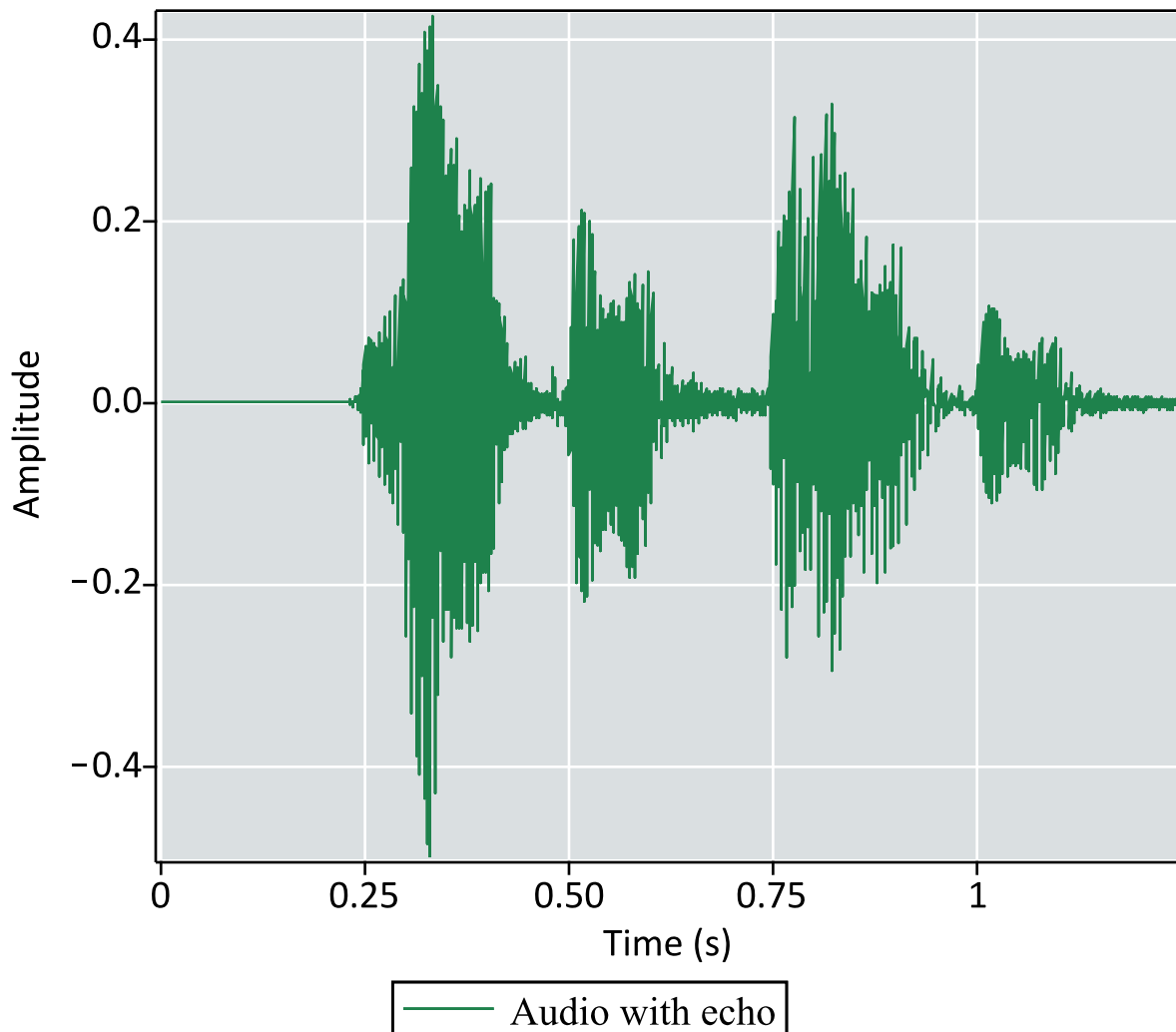
```
> SetProperty("Speaker0", samplerate, Fs);
   Play(origAudio, "Speaker0")

> t := Vector(numelems(origAudio), i-> 1.0*i/Fs, datatype = float
[8]):

pltOrigAudio := plot(t, origAudio, thickness = 0, color = Color
```

```
("RGB",[30/255, 130/255, 76/255]), legend = "Audio with echo"):
```

```
display(pltOrigAudio, axes = boxed, size = [1000, 400],
gridlines, legendstyle = [font=[Calibri], font = [Calibri],
labels = ["Time (s)", "Amplitude"], labeldirections =
[horizontal, vertical], labelfont = [Calibri], background =
Color("RGB", [218/255, 223/255, 225/255]), axis = [gridlines =
[5, color = Color("RGB", [1, 1, 1])]])
```



▼ Use Cepstrum Function to Identify Start of Echo

```
> Cepstrum := proc(V)
  local N:
  uses SignalProcessing:
  N := numelems(V):
  return Re(InverseFFT(ln~(abs(FFT(V) * sqrt(1.*N)))) / sqrt
    (1.*N)) ;
end proc:
```

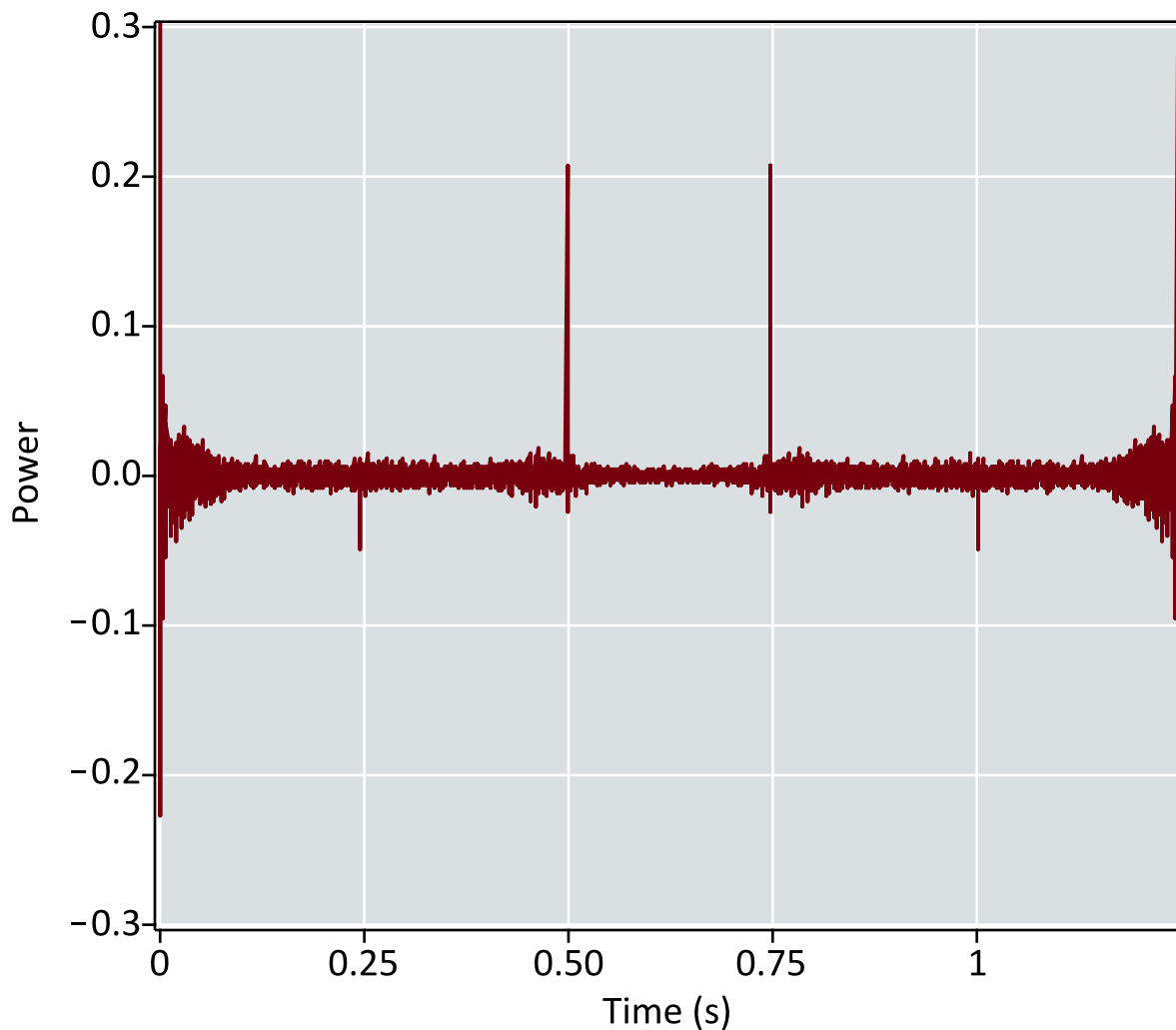
```
> c := Cepstrum(origAudio):
```

Warning, size of Array must be a power of two greater than two,

[using DFT instead and suppressing this warning for the remainder of this session](#)

```
> plotCeps := plot(t, c):
```

```
display(plotCeps, axes = boxed, size = [1000, 400], view =  
[default, -0.3..0.3], font = [Calibri], labels = ["Time (s)",  
"Power"], labeldirections = [horizontal, vertical], labelfont =  
[Calibri], background = Color("RGB", [218/255, 223/255, 225/255]  
, axis = [gridlines = [5, color = Color("RGB", [1, 1, 1])]])
```



▼ Apply an IIR Filter to Remove Echo

The cepstrum plot shows a strong discontinuity at 0.5 s - that's where the echo probably begins. Let's find the precise location

```
> threshold := map(x -> piecewise(x < 0.2, 0, 1), c):  
ind := ArrayTools:-SearchArray(threshold, 5);
```

$$ind := \begin{bmatrix} 1 \\ 2 \\ 4 \\ 5514 \\ 8228 \end{bmatrix} \quad (4.1)$$

So the echo starts at an index of 5514. Now use this information to generate an IIR filter to remove the echo. The "0.5" in the filter coefficients is an attenuation factor.

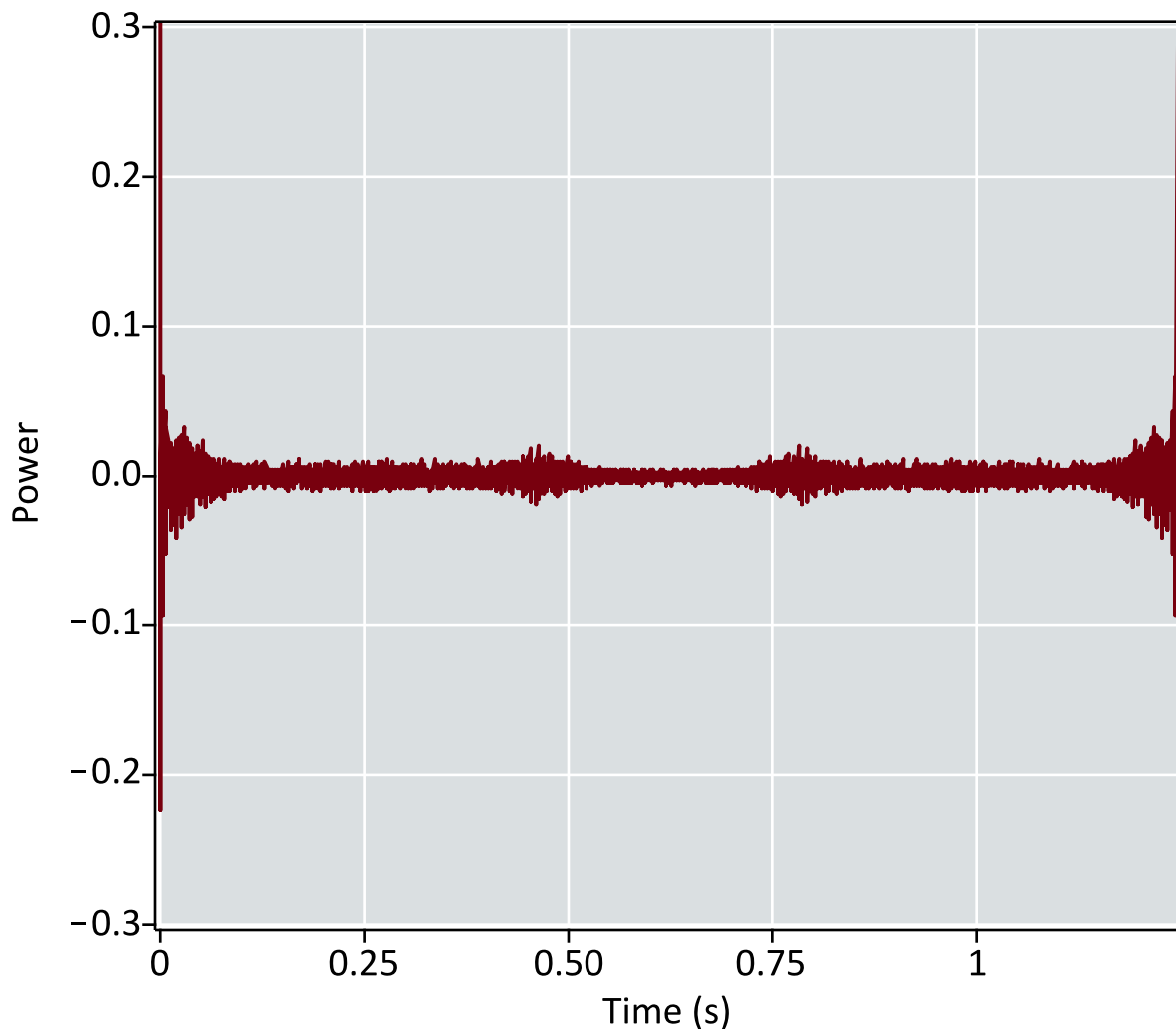
```
> filterCoeffs := Array([1, 0$5512, 0.5]):
  cleanAudio := Filter(origAudio, filterCoeffs, Array([1])):
```

View the filtered cepstrum

```
> c_filtered := Cepstrum(cleanAudio):

plotCeps_filtered := plot(t, c_filtered):

display(plotCeps_filtered, axes = boxed, size = [1000, 400],
view = [default, -0.3..0.3], font = [Calibri], labels = ["Time
(s)", "Power"], labeldirections = [horizontal, vertical],
labelfont = [Calibri], background = Color("RGB", [218/255,
223/255, 225/255]), axis = [gridlines = [5, color = Color("RGB",
[1, 1, 1])]])
```



Playing the audio reveals that the echo has been removed

```
> Play(cleanAudio, "Speaker0")
```

Write filtered audio to wave file

```
> Write("maplesim_clean.wav", Create(cleanAudio, rate = Fs)):
```

▼ Visualize Waveform of Original and De-Echoed Audio

```
> pltCleanAudio := plot(t, cleanAudio, thickness = 0, color =
  "black", legend = "Audio with echo removed"):
```

```
display(pltOrigAudio, pltCleanAudio, size = [1000, 400], axes =
  boxed, gridlines, legendstyle = [font = [Calibri]], axesfont =
  [Calibri], labels = ["Time (s)", "Amplitude"], labeldirections =
  [horizontal, vertical], labelfont = [Calibri], background =
  Color("RGB", [218/255, 223/255, 225/255]), axis = [gridlines =
  [5, color = Color("RGB", [1, 1, 1])]])
```

