In [1]:

```
from __future__ import print_function, division

%matplotlib inline

import thinkdsp
import thinkplot

import numpy as np

from ipywidgets import interact, interactive, fixed
import ipywidgets as widgets
from IPython.display import display
```

In [2]:

```
cos_sig = thinkdsp.CosSignal(freq=440, amp=1.0, offset=0)
sin_sig = thinkdsp.SinSignal(freq=880, amp=0.5, offset=0)
```

In [3]:

```
mix = sin_sig + cos_sig
mix
```

Out[3]:

```
<thinkdsp.SumSignal at 0x1c72f5536a0>
```

In [4]:

```
wave = mix.make_wave(duration=0.5, start=0, framerate=11025)
wave
```

Out[4]:

```
<thinkdsp.Wave at 0x1c732adc550>
```

In [5]:

```
from IPython.display import Audio
audio = Audio(data=wave.ys, rate=wave.framerate)
audio
```

Out[5]:

0:00 / 0:00

In [6]:

```
wave.make_audio()
```

Out[6]:

0:00 / 0:00

In [7]:

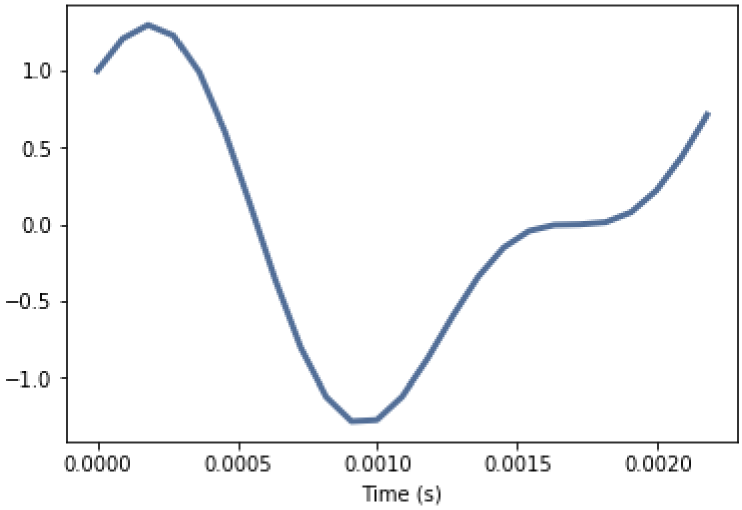
```
period = mix.period
segment = wave.segment(start=0, duration=period*1)
period
```

Out[7]:

```
0.0022727272727272726
```

In [8]:

```
segment.plot()
thinkplot.config(xlabel='Time (s)')
```

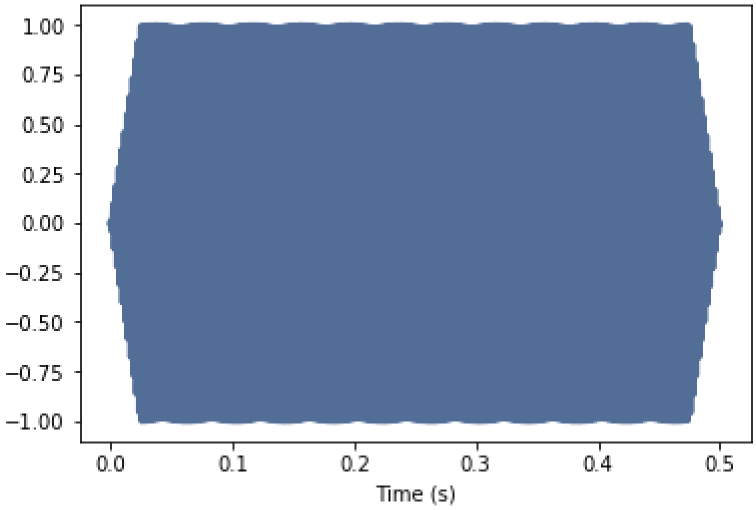


segment.plot() thinkplot.config(xlabel='Time (s)')

정규화 apodize tapers the beginning and end of the wave so it doesn't click when you play it.

In [9]:

```
wave.normalize()
wave.apodize()
wave.plot()
thinkplot.config(xlabel='Time (s)')
```



In [10]:

```
wave.write('temp.wav')
```

```
Writing temp.wav
```

wave파일 읽기

In [11]:

```
thinkdsp.play_wave(filename='temp.wav', player='aplay')
```

In [12]:

```
wave = thinkdsp.read_wave('463324__jb-multimedia19__soundsaladv2.wav')
```

세그먼트 만들기

In [13]:

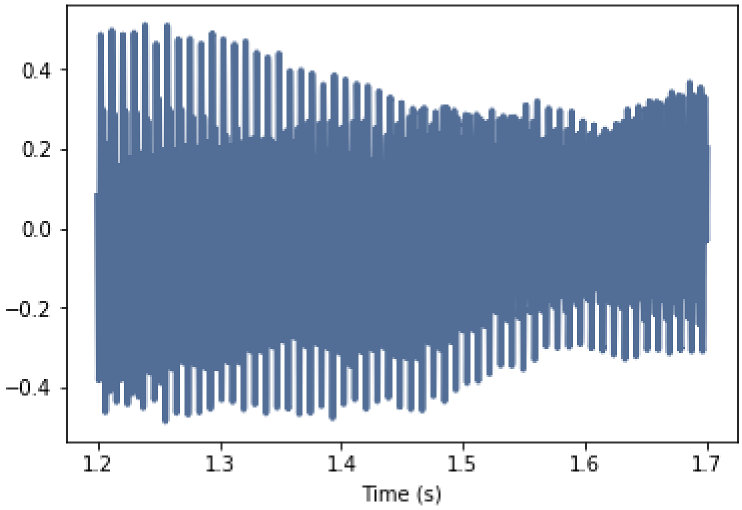
```
wave.make_audio()
```

Out[13]:

0:00 / 0:20

In [14]:

```
start = 1.2
duration = 0.5
segment = wave.segment(start, duration)
segment.plot()
thinkplot.config(xlabel='Time (s)')
```



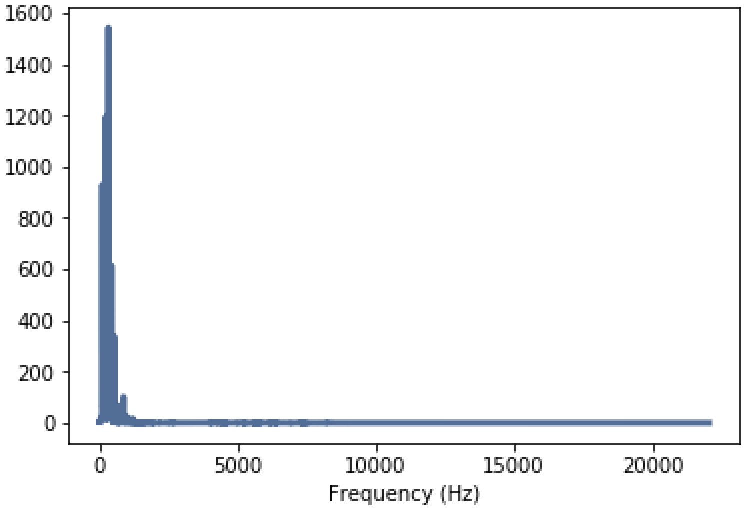
In [15]:

```
spectrum = segment.make_spectrum()
```

기저주파수를 440hz와 880hz로 설정하여 wave를 만들어서 harmonic을 만들었다.

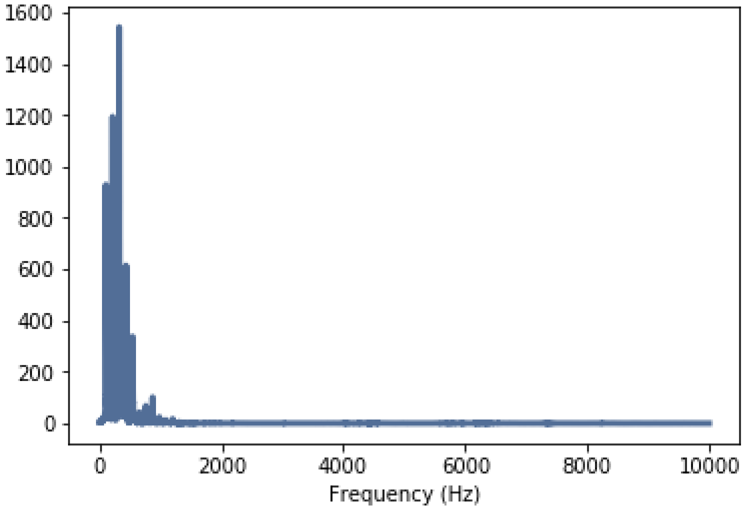
In [16]:

```
spectrum.plot()
thinkplot.config(xlabel='Frequency (Hz)')
```



In [17]:

```
spectrum.plot(high=10000)
thinkplot.config(xlabel='Frequency (Hz)')
```

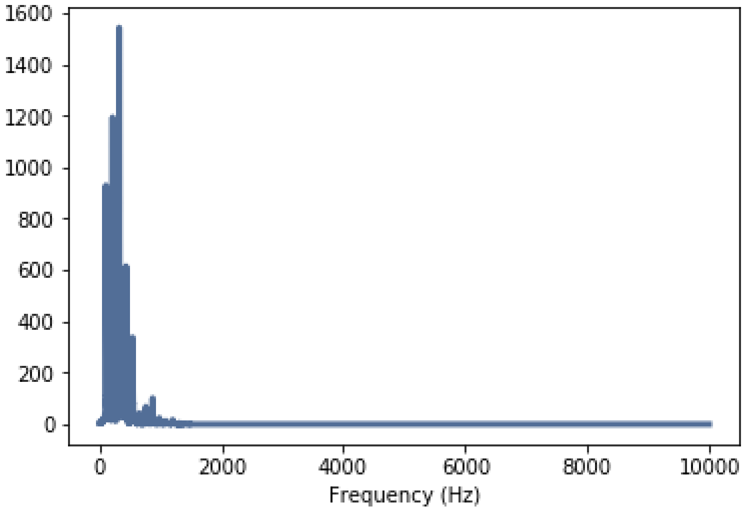


In [18]:

```
spectrum.low_pass(1500)
```

In [19]:

```
spectrum.plot(high=10000)
thinkplot.config(xlabel='Frequency (Hz)')
```



In [20]:

```
spectrum.high_pass(400)
```

In [21]:

```
spectrum.band_stop(420,1400)
```

In [22]:

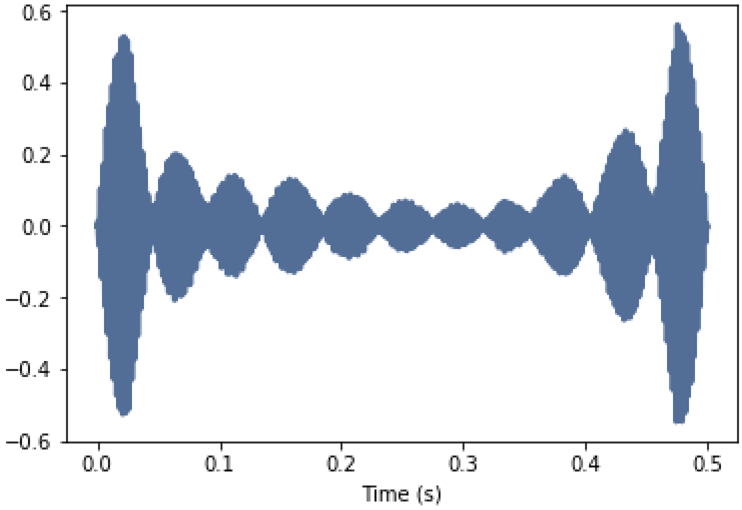
```
filtered = spectrum.make_wave()
```

In [23]:

```
filtered.normalize()
```

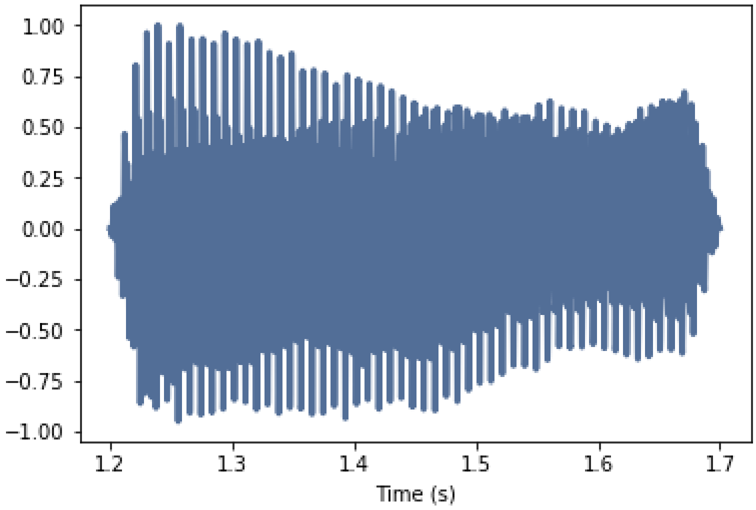
In [24]:

```
filtered.apodize()
filtered.plot()
thinkplot.config(xlabel='Time (s)')
```



In [25]:

```
segment.normalize()
segment.apodize()
segment.plot()
thinkplot.config(xlabel='Time (s)')
```



In [86]:

```
segment.make_audio()
```

Out[86]:

0:00 / 0:00

In [87]:

```
filtered.make_audio()# band_stop low pass high pass를 통해 변형된 소리
```

Out[87]:

0:00 / 0:00

최종적으로 420~1400hz까지만 filtering하였습니다.