In [1]:

```
from __future__ import print_function, division

import thinkdsp
import thinkplot
import thinkstats2

import numpy as np
import pandas as pd

import warnings
warnings.filterwarnings('ignore')

%matplotlib inline
```

In [2]:

```
wave=thinkdsp.read_wave('13793__soarer__north-sea.wav')
wave.normalize()
wave.make_audio()
```

Out[2]:

0:00 / 4:58

In [3]:

```
def serial_corr(wave,lag=1):
    n=len(wave)
    y1=wave.ys[lag:]
    y2=wave.ys[:n-lag]
    corr=np.corrcoef(y1,y2,ddof=0)[0,1]
    return corr
```

In [4]:

```
def autocorr(wave):
    lags=range(len(wave.ys)//2)
    corrs=[serial_corr(wave,lag) for lag in lags]
    return lags, corrs
```

In [5]:

```
def estimate_fundamental(segment, low=30, high=200):
    lags, corrs = autocorr(segment)
    lag = np.array(corrs[low:high]).argmax() + low #argmax()함수는 제일 큰 값의 인덱스를 return
    period = lag / segment.framerate
    frequency = 1 / period
    return frequency
```

In [6]:

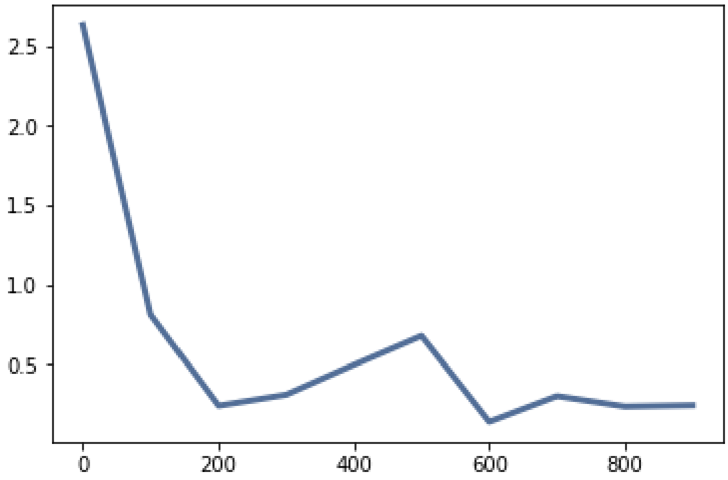
```
duration = 0.01
segment = wave.segment(start=0.2, duration=duration)
freq = estimate_fundamental(segment)
freq
```

Out[6]:

```
512.7906976744185
```

In [7]:

```
duration = 0.01
segment = wave.segment(start=0.2, duration=duration)
spectrum=segment.make_spectrum()
spectrum.plot(high=1000)
```



In [8]:

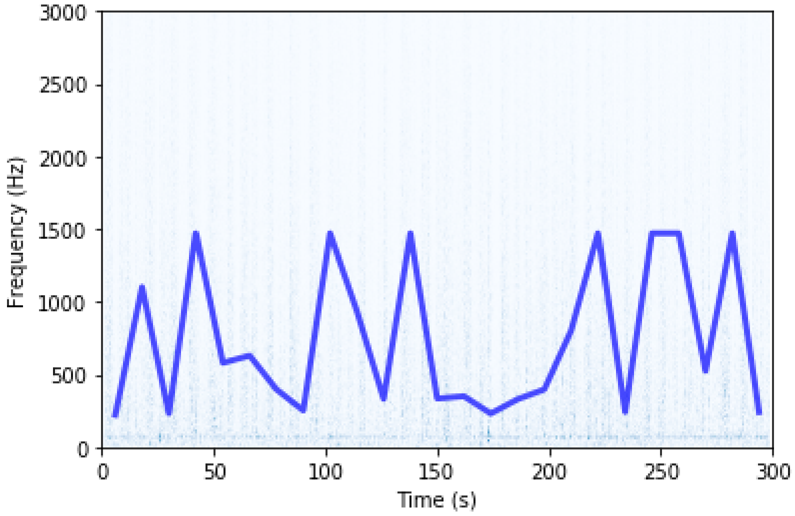
```
step = 12.0
starts = np.arange(0.0, 300.0, step)

ts = []
freqs = []

for start in starts:
    ts.append(start + step/2)
    segment = wave.segment(start=start, duration=duration)
    freq = estimate_fundamental(segment)
    freqs.append(freq)
```

In [12]:

```
wave.make_spectrogram(4096).plot(high=3000)
thinkplot.plot(ts, freqs, color='blue')
thinkplot.config(xlabel='Time (s)',
                 ylabel='Frequency (Hz)',
                 xlim=[0, 300.0],
                 ylim=[0, 3000])
```



In [ ]:

```

```