UTCDateTime

Documentation: UTCDateTime

In [1]:
```
from obspy.core import UTCDateTime
t = UTCDateTime()
t1 = UTCDateTime("2014-03-14T10:15:00+01:00")
t2= UTCDateTime(2012, 9, 7, 12, 15, 0)
t3= UTCDateTime(1347020100.0)
print t.year, t.julday, t.timestamp, t.weekday
print t+86400
```

```
2014 91 1396357393.59 1
2014-04-02T13:03:13.587113Z
```

Načítání seismogramů

In [2]:
```
from obspy.core import read
st = read('http://examples.obspy.org/RJOB_061005_072159.ehz.new
# st = read('example.gse2')
print st
tr = st[0]  # assign first and only trace to new variable
print tr
```

```
1 Trace(s) in Stream:
.RJOB..Z | 2005-10-06T07:21:59.849998Z - 2005-10-
06T07:24:59.844998Z | 200.0 Hz, 36000 samples
.RJOB..Z | 2005-10-06T07:21:59.849998Z - 2005-10-
06T07:24:59.844998Z | 200.0 Hz, 36000 samples
```

In [3]:
```
print tr.stats
tr.stats.station
tr.stats.gse2.datatype
```

```
         network:
         station: RJOB
        location:
         channel: Z
       starttime: 2005-10-06T07:21:59.849998Z
         endtime: 2005-10-06T07:24:59.844998Z
   sampling_rate: 200.0
           delta: 0.005
            npts: 36000
           calib: 0.0948999971151
         _format: GSE2
```

```
            gse2: AttribDict({'instype': '      ', 'datatype':
'CM6', 'hang': -1.0, 'auxid': 'RJOB', 'vang': -1.0, 'calper':
1.0})
```

Out[3]: 'CM6'

In [4]:
```
print tr.data
print tr.data[0:3]
print len(tr)
```

```
[-38  12  -4 ..., -14  -3  -9]
[-38  12  -4]
36000
```

Konverze formátů, ukládání

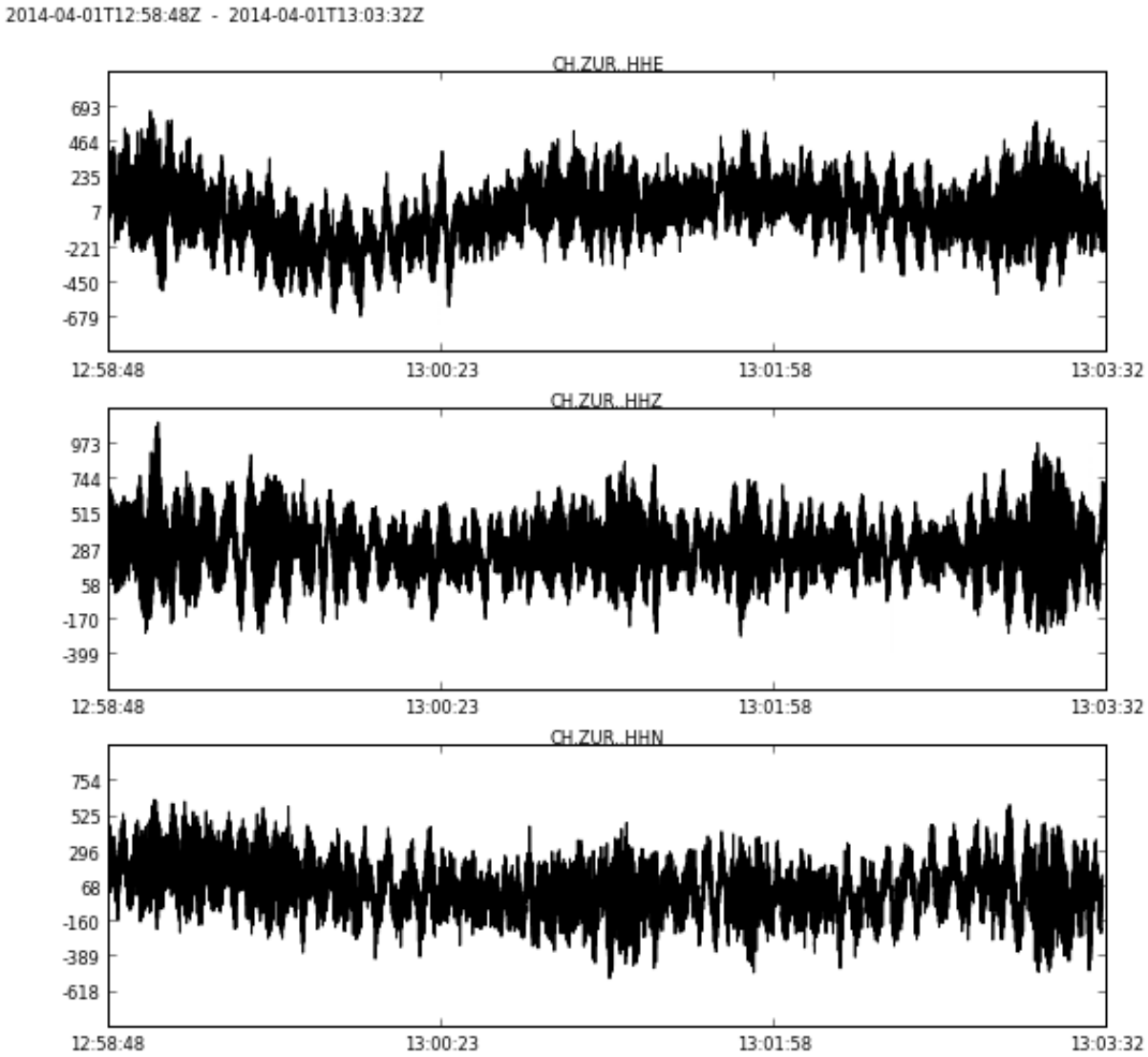
Dokumentace: funkce Stream.write, podporované formáty

In [5]:
```
from obspy.core import read
st = read('example.gse2')
st.write('example.sac', format='SAC')
st.write('example.ascii', format='SLIST')
```

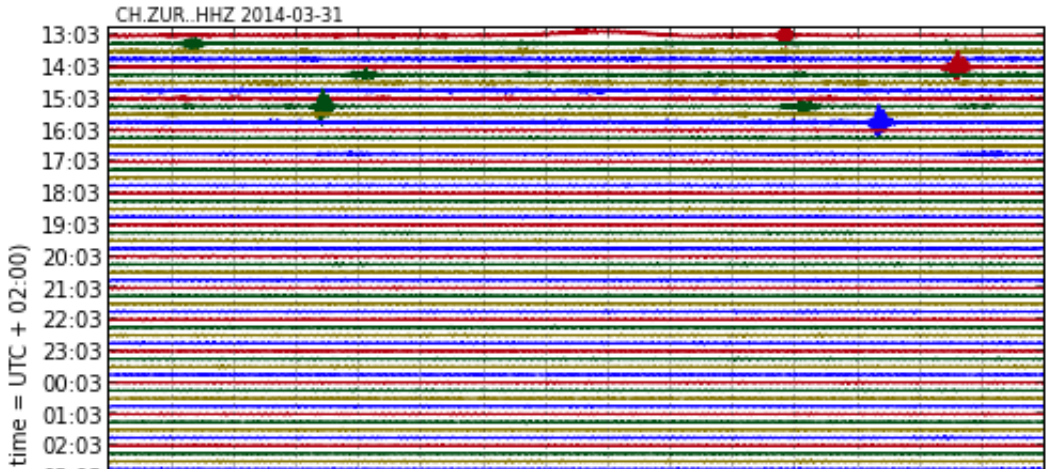
Arclink

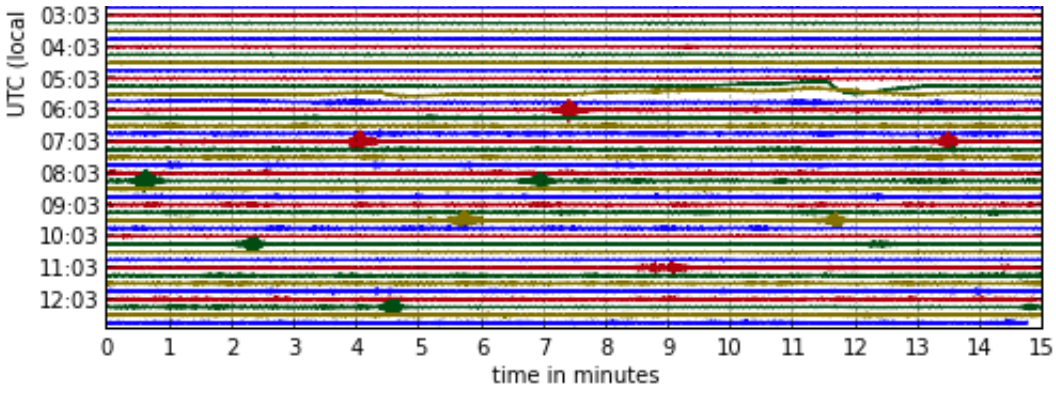
In [6]:
```
from obspy.core import UTCDateTime
from obspy.arclink import Client

t = UTCDateTime()
try:
    client = Client(host="arclink.ethz.ch", user="jiri.vackar@s
    st = client.getWaveform('CH', 'ZUR', '', 'HH*', t-300, t)
except:
    print 'stahovani selhalo'
    from obspy.core import read
    st = read('ZUR.mseed')
else:
    print st
st.plot()
```

```
3 Trace(s) in Stream:
CH.ZUR..HHE | 2014-04-01T12:58:48.841667Z - 2014-04-
01T13:03:31.958333Z | 120.0 Hz, 33975 samples
CH.ZUR..HHZ | 2014-04-01T12:58:48.841700Z - 2014-04-
01T13:03:32.883367Z | 120.0 Hz, 34086 samples
CH.ZUR..HHN | 2014-04-01T12:58:48.841667Z - 2014-04-
01T13:03:31.391667Z | 120.0 Hz, 33907 samples
```



In [7]:
```
try:
    st = client.getWaveform('CH', 'ZUR', '', 'HHZ', t-86400, t)
except:
    st = read('ZUR2.mseed')
st[0].plot(type='dayplot')
```





Filtrování

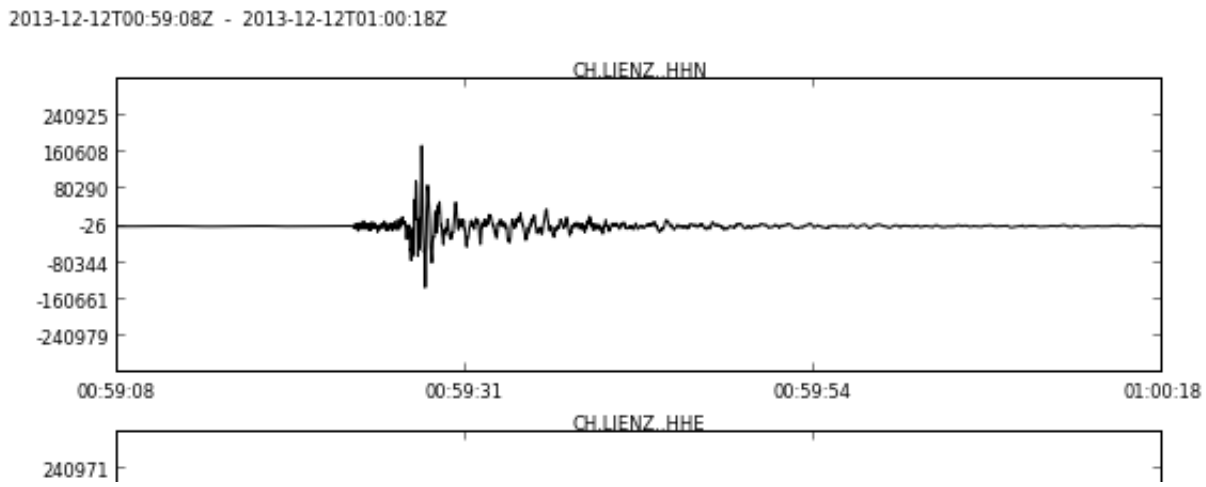
The following script shows how to filter a seismogram. The example uses a zero-phase-shift low-pass filter with a corner frequency of 1 Hz using 2 corners. This is done in two runs forward and backward, so we end up with 4 corners de facto.

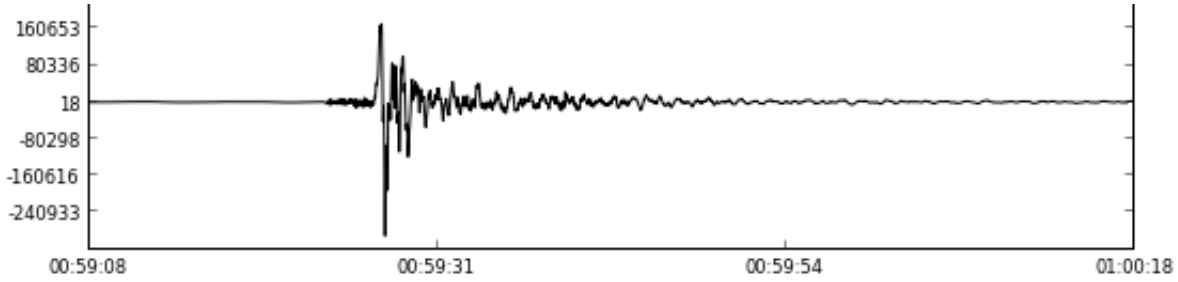
The available filters are:

- bandpass
- bandstop
- lowpass
- highpass

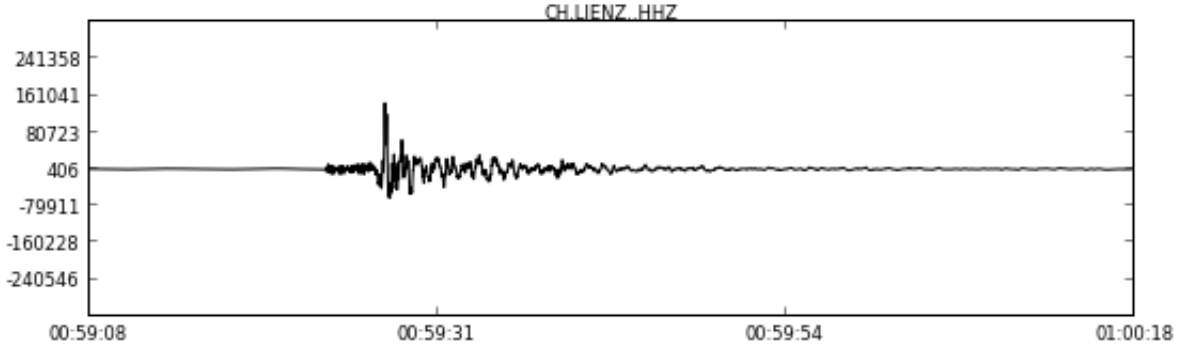
Dokumentace: [obspy.core.stream.Stream.filter]

In [8]:
```
t = UTCDateTime("2013-12-12 0:59:18")
try:
    st = client.getWaveform('CH', 'LIENZ', '', 'HH*', t-10, t+6(
except:
    st = read('LIENZ.mseed')
st.plot()
st_filt = st.copy()
st_filt.filter('lowpass', freq=1.0, corners=2, zerophase=True)
st_filt.plot()
```



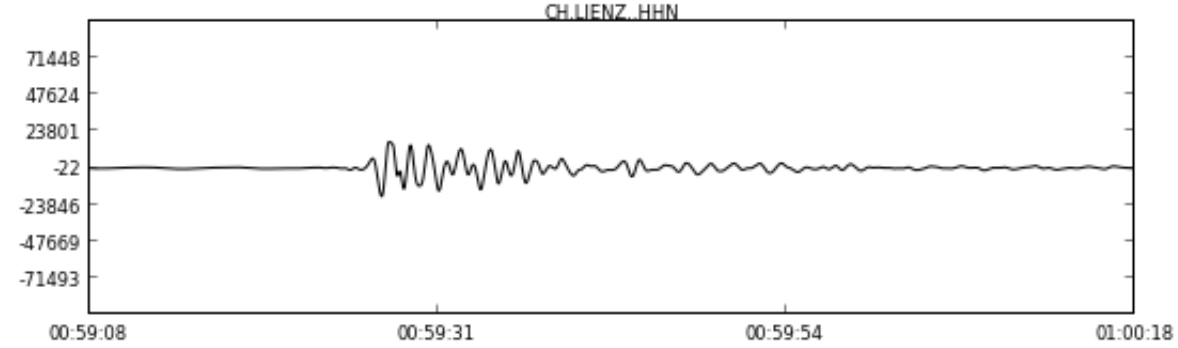


CH.LIENZ..HHZ

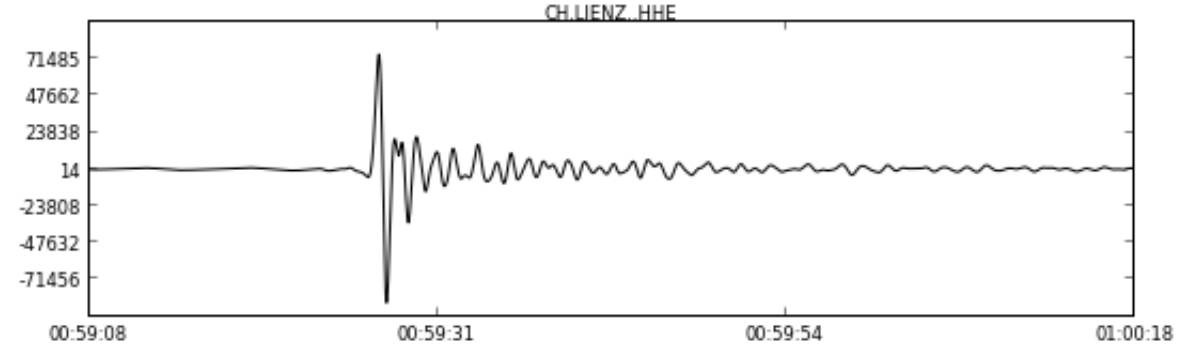


2013-12-12T00:59:08Z - 2013-12-12T01:00:18Z

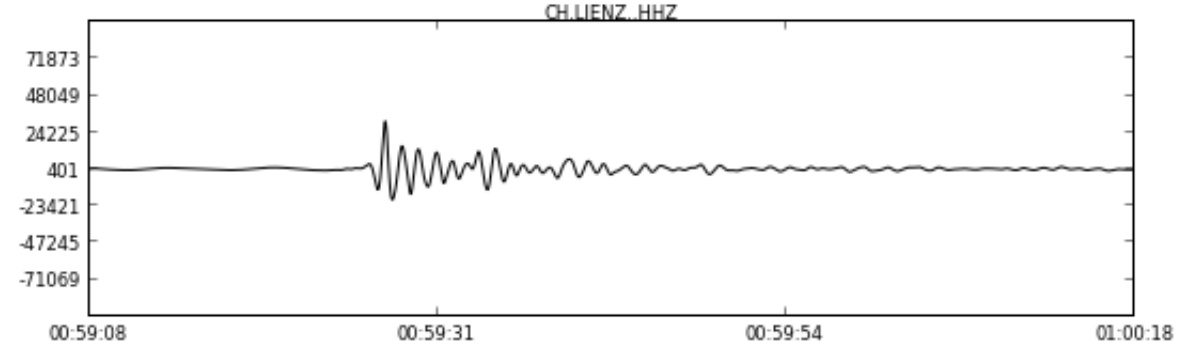
CH.LIENZ..HHN



CH.LIENZ..HHE



CH.LIENZ..HHZ

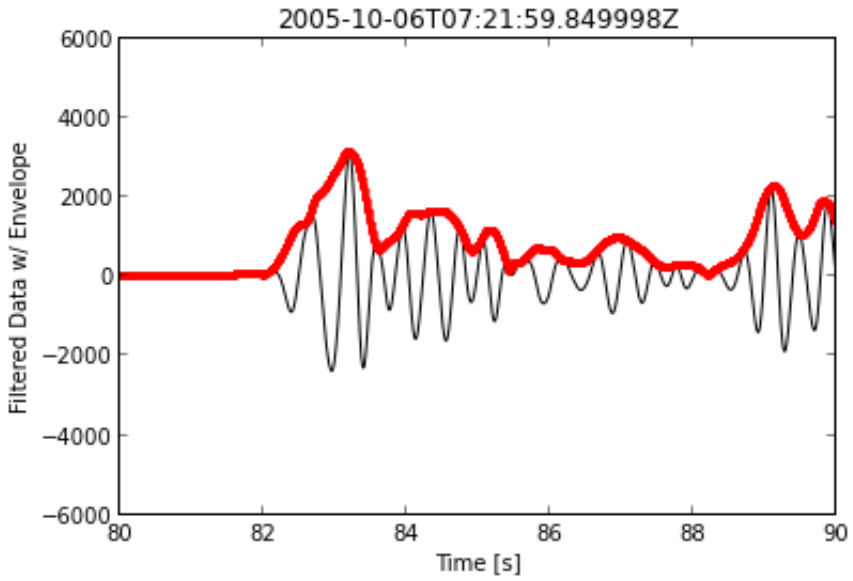


Obálky seismogramů a vykreslování pomocí matplotlib

Dokumentace: Seismogram envelopes Matplotlib: plot

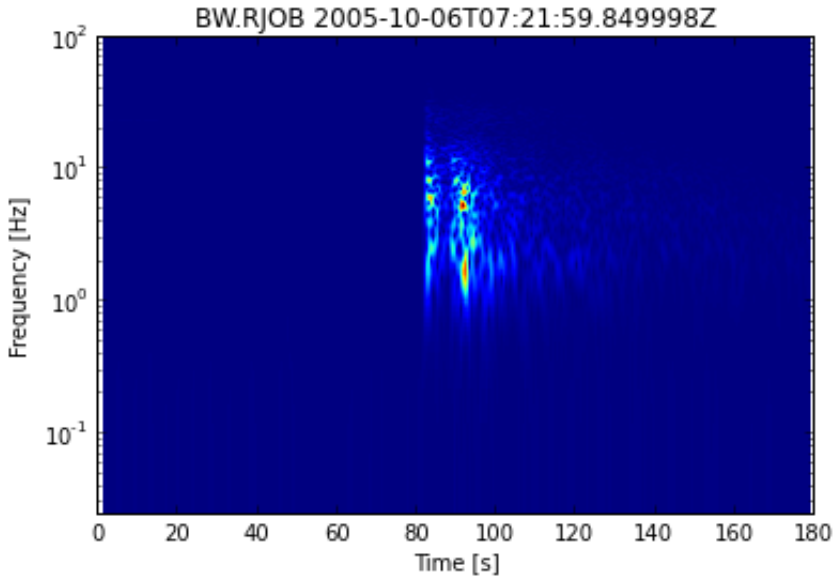
In [13]:
```
import numpy as np
import matplotlib.pyplot as plt
from obspy.core import read
import obspy.signal

st = read("http://examples.obspy.org/RJOB_061005_072159.ehz.new
# st = read('example.gse2')
npts = st[0].stats.npts
samprate = st[0].stats.sampling_rate

st.filter('bandpass', freqmin=1, freqmax=3, corners=2, zerophas

# Envelope of filtered data
data_envelope = obspy.signal.filter.envelope(st[0].data)

# The plotting, plain matplotlib
t = np.arange(0, npts / samprate, 1 / samprate)
plt.plot(t, st[0].data, 'k')
print len(t), len(data_envelope)
plt.plot(t, data_envelope, 'r.')
plt.title(st[0].stats.starttime)
plt.ylabel('Filtered Data w/ Envelope')
plt.xlabel('Time [s]')
plt.xlim(80, 90)
plt.show()
```

```
36000 36000
```



Spektrogram

In [14]:
```
from obspy.core import read

st = read("http://examples.obspy.org/RJOB_061005_072159.ehz.new
# st = read('example.gse2')
st.spectrogram(log=True, title='BW.RJOB ' + str(st[0].stats.sta
```

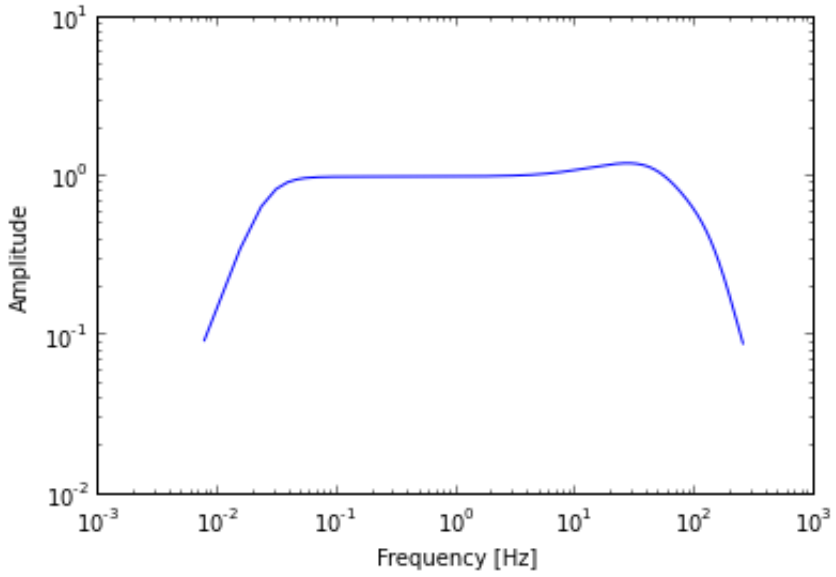


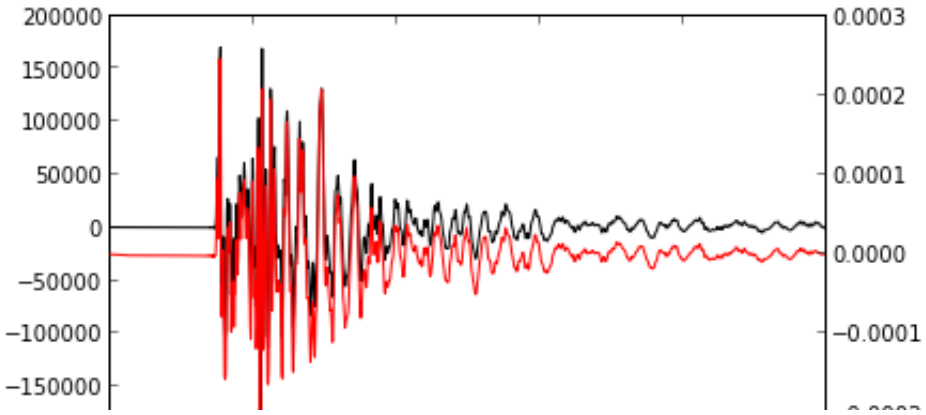
Out[14]: [None]

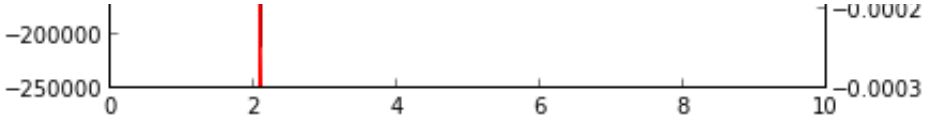
Nuly a póly, přenosová funkce přístroje

In [15]:
```
from obspy.core import UTCDateTime
from obspy.arclink import Client
from obspy.signal import pazToFreqResp
from obspy.sac.sacio import attach_paz
import matplotlib.pyplot as plt

t = UTCDateTime("2013-12-12 10:34:21")
try:
    client = Client(host="arclink.ethz.ch", user="jiri.vackar@s
    st = client.getWaveform('CH', 'VANNI', '', 'HHZ', t, t+10)
    st.write('VANNI.mseed', format='MSEED')
except:
    st = read('VANNI.mseed')
tr = st[0]
tr2 = tr.copy()

# stahneme nuly a poly pres ArcLink, popr. nacteme ze souboru
```

```
try:
    paz = client.getPAZ('CH', 'VANNI', '', 'HHZ', t)
except:
    attach_paz(tr, 'VANNI.resp', tovel=True)
    paz = tr.stats.paz

# nakreslime frekvencni odezvu
h,f = pazToFreqResp(paz['poles'], paz['zeros'], paz['gain'], 0.(
plt.figure()
plt.loglog(f, abs(h))
plt.xlabel('Frequency [Hz]')
plt.ylabel('Amplitude')
plt.show()

# provedeme instrumentalni korekci
tr2.simulate(paz_remove=paz)

# vykreslime
t = np.arange(0, tr.stats.npts / tr.stats.sampling_rate, 1 / tr
plt.plot(t, tr.data, 'k')
plt2 = plt.twinx()
plt2.plot(t, tr2.data, 'r')
plt.show()
```






−200000
−250000
0
2
4
6
8
10
−0.0002
−0.0003