Example LMSexample

January 25, 2017

In [2]:
```
import numpy as np
from sound import *
import matplotlib.pyplot as plt
%matplotlib inline

x, fs = wavread('speech8kHz.wav')
```

```
('Number of channels: ', 1)
('Number of bytes per sample:', 2)
('Sampling rate: ', 8000)
('Number of samples:', 60246)
```

- Normalized float, -1<x<1

In [3]:
```
x = np.array(x,dtype=float)/2**15
print np.size(x)
e = np.zeros(np.size(x))
h = np.zeros(10)
```

```
60246
```

In [5]:
```
for n in range(10, len(x)):

    #prediction error and filter, using the vector of the time reversed IR:
    e[n] = x[n] - np.dot(x[n-10+np.arange(0,10)], np.flipud(h))

    #LMS update rule, according to the definition above:
    h = h + 1.0* e[n]*np.flipud(x[n-10+np.arange(0,10)])
```

In [6]:
```
print "Mean squared prediction error:", np.dot(e, e) /np.max(np.size(e))
```

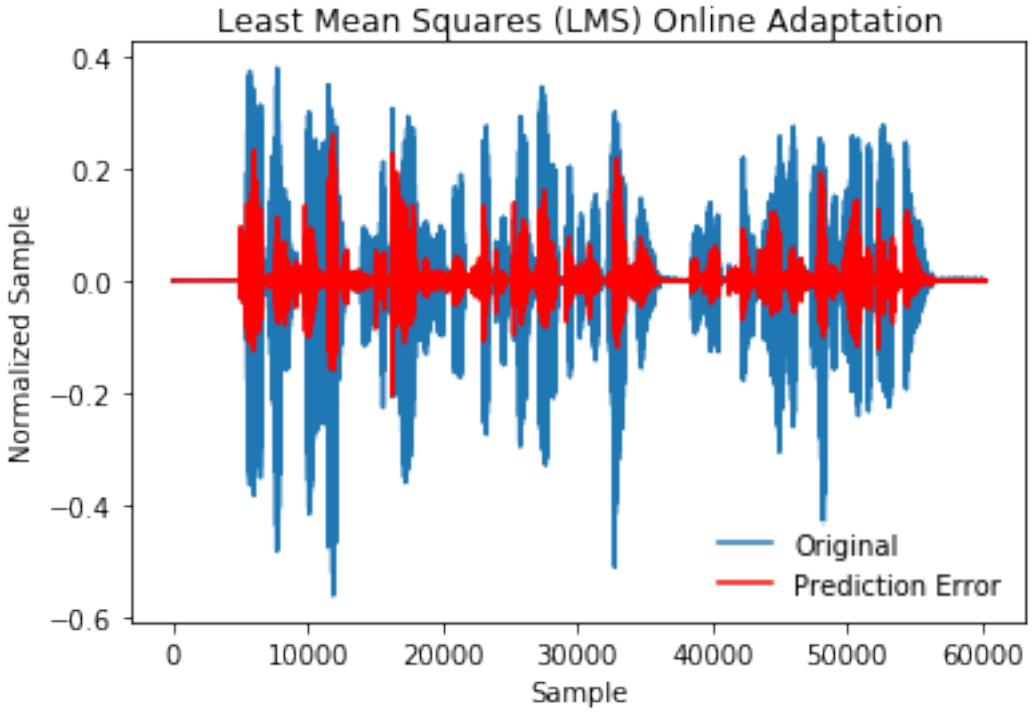
```
Mean squared prediction error: 0.000302959016563
```

- Listen to to the prediction error:

In [7]:
```
sound(2**15*e, fs)
```

```
* done
```

```
In [8]: plt.figure()
        plt.plot(x)
        plt.plot(e,'r')
        plt.xlabel('Sample')
        plt.ylabel('Normalized Sample')
        plt.title('Least Mean Squares (LMS) Online Adaptation')
        plt.legend(('Original','Prediction Error'))
        plt.show()
```



Observe: its prediction error is bigger than in the **LPC** case, but we also don't need to transmit the prediction coefficients as side information.

The comparison plot of the original to the prediction error(above plot)

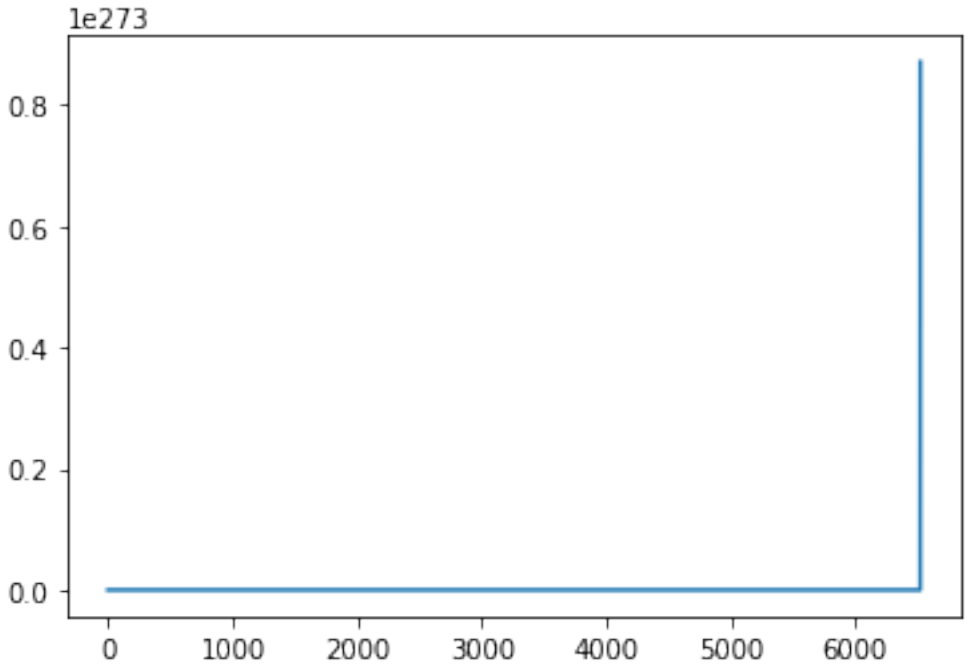
- For the **decoder** we get the reconstruction

```
In [18]: h = np.zeros(10);
         xrek = np.zeros(np.size(x));
         for n in range(10, len(x)):
             if n> 4000 and n< 4010:
                print "decoder h: ", h
             P=np.dot(xrek[n-10+np.arange(10)], np.flipud(h))
             xrek[n] = e[n] + P
```

```
            #LMS update:
            h = h + 1.0 * e[n]*np.flipud(xrek[n-10+np.arange(10)]);
        plt.plot(xrek)
        plt.show()
        #Listen to the reconstructed signal:
        sound(2**15*xrek,fs)
```

```
decoder h:  [ -1.36650874e-05   9.81964536e-06  -5.72557714e-06   2.32909304e-06
  -4.35770302e-07   7.07384488e-08  -2.30022390e-07   2.21651788e-07
  -6.70549385e-08  -1.76020424e-07]
decoder h:  [ -1.36725379e-05   9.81964536e-06  -5.72930239e-06   2.33281830e-06
  -4.39495557e-07   7.81889603e-08  -2.41198158e-07   2.29102300e-07
  -7.45054506e-08  -1.68569912e-07]
decoder h:  [ -1.36874387e-05   9.82709579e-06  -5.72930239e-06   2.33654351e-06
  -4.43220774e-07   8.19141774e-08  -2.48648592e-07   2.40277952e-07
  -8.19558856e-08  -1.61119477e-07]
decoder h:  [ -1.37060648e-05   9.84572185e-06  -5.73861542e-06   2.33654351e-06
  -4.47877287e-07   8.65706904e-08  -2.53305105e-07   2.49590979e-07
  -9.59254259e-08  -1.51806450e-07]
decoder h:  [ -1.37293472e-05   9.86434779e-06  -5.75724136e-06   2.34585649e-06
  -4.47877287e-07   9.12271762e-08  -2.57961591e-07   2.54247464e-07
  -1.05238398e-07  -1.37836991e-07]
decoder h:  [ -1.37433164e-05   9.87831692e-06  -5.76841667e-06   2.35703179e-06
  -4.53464940e-07   9.12271759e-08  -2.60755417e-07   2.57041290e-07
  -1.08032224e-07  -1.32249339e-07]
decoder h:  [ -1.37461100e-05   9.88297292e-06  -5.77307267e-06   2.36075659e-06
  -4.57189741e-07   9.30895765e-08  -2.60755417e-07   2.57972491e-07
  -1.08963424e-07  -1.31318139e-07]
decoder h:  [ -1.37451786e-05   9.88017871e-06  -5.76841565e-06   2.35609957e-06
  -4.53464126e-07   8.93639623e-08  -2.58892610e-07   2.57972491e-07
  -1.08032021e-07  -1.32249542e-07]
decoder h:  [ -1.37461099e-05   9.87924736e-06  -5.76562160e-06   2.35144282e-06
  -4.48807377e-07   8.56385623e-08  -2.55167210e-07   2.56109791e-07
  -1.08032021e-07  -1.33180892e-07]
```

```
c:\python27\lib\site-packages\ipykernel\__main__.py:9: RuntimeWarning: invalid value encountered
```




```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)

<ipython-input-18-026c0ce1893c> in <module>()
     11 plt.show()
     12 #Listen to the reconstructed signal:
---> 13 sound(2**15*xrek,fs)


C:\Users\vijay\workspace\DSP\modules\sound.pyc in sound(s, FS)
     30         samples=s[i*CHUNK:((i+1)*CHUNK)];
     31         samples=clip(samples,-2**15,2**15-1)
---> 32         data=struct.pack('h' * CHUNK, *samples);
     33         #print data[1]
     34         #Writing data back to audio output stream:


ValueError: cannot convert float NaN to integer
```

In [17]:
```
#Listen to the reconstructed signal:
sound(2**15*xrek, fs)
```

```
---------------------------------------------------------------------------

ValueError                                Traceback (most recent call last)

<ipython-input-17-851a0cc24452> in <module>()
  1 #Listen to the reconstructed signal:
----> 2 sound(2**15*xrek, fs)


C:\Users\vijay\workspace\DSP\modules\sound.pyc in sound(s, FS)
 30         samples=s[i*CHUNK:((i+1)*CHUNK)];
 31         samples=clip(samples,-2**15,2**15-1)
---> 32         data=struct.pack('h' * CHUNK, *samples);
 33         #print data[1]
 34         #Writing data back to audio output stream:


ValueError: cannot convert float NaN to integer
```

Due to prediction error is being generated sample wise i.e., filter design(keeps changing) is based on the past samples and then it predicts the future sample, so the prediction error keeps diverging and even the small changes cumulatively adding up, results into a large value and hence there is an explosion at around 6500th sample. Hence it is just playable uptil the same number samples.