Example Matched Filter

January 25, 2017

0.1 Python Example

Construct a signal 'sig' (length 11):

- Importing relevant modules

```
In [1]: %matplotlib inline
        import numpy as np
        import matplotlib.pyplot as plt
        import scipy.signal as sp
```

- Signal Construction

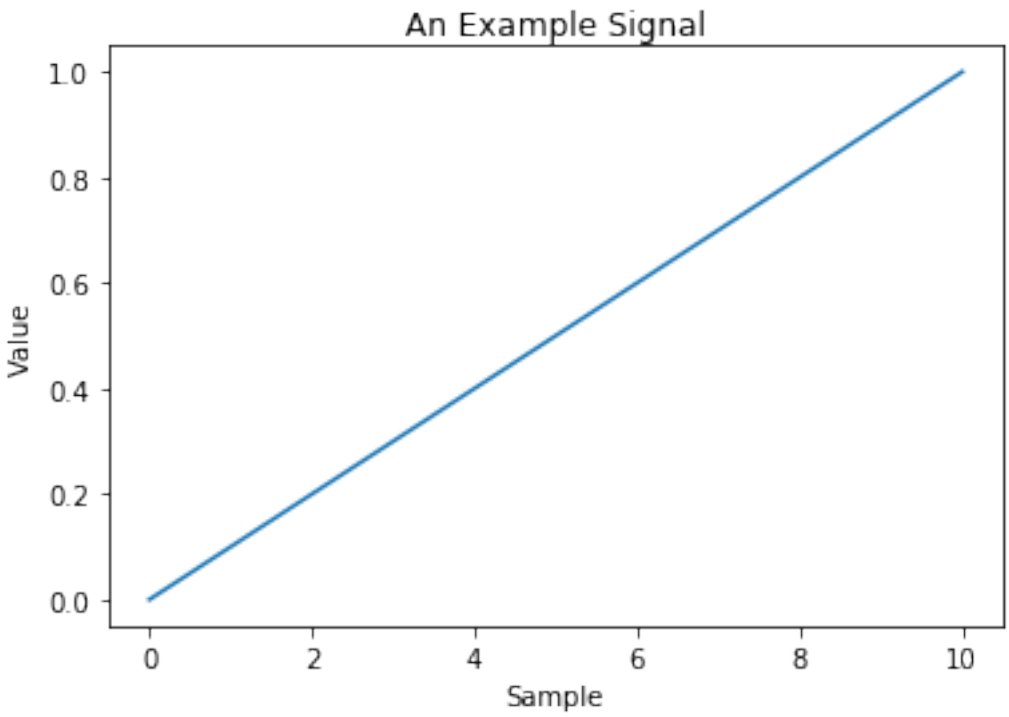
```
In [2]: sig = np.arange(0, 1.1, 0.1)
        sig
```

```
Out[2]: array([ 0. ,  0.1,  0.2,  0.3,  0.4,  0.5,  0.6,  0.7,  0.8,  0.9,  1. ])
```

- Plotting the signal

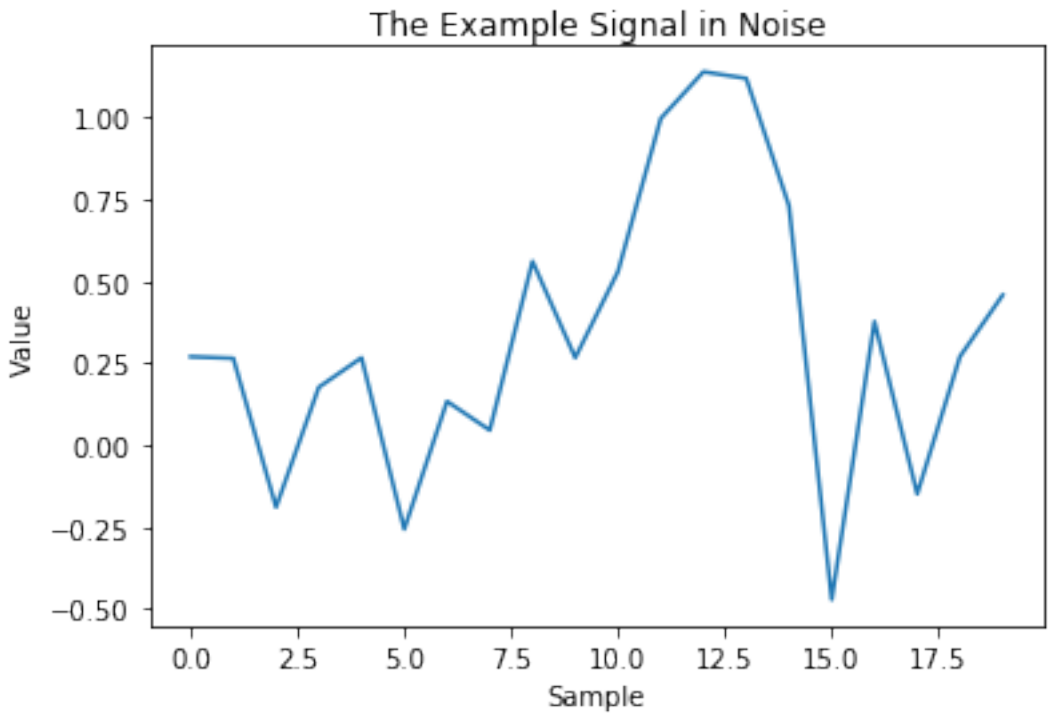
```
In [3]: plt.plot(sig)
        plt.xlabel('Sample')
        plt.ylabel('Value')
        plt.title('An Example Signal')
```

```
Out[3]: <matplotlib.text.Text at 0x89d8d70>
```



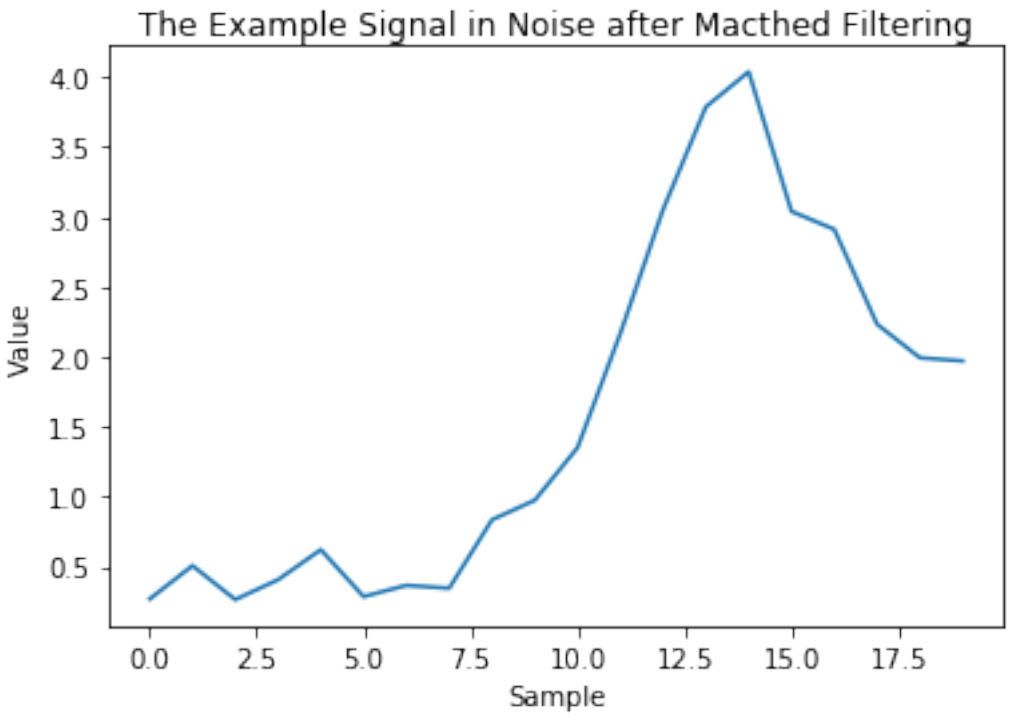
- Now we add noise and extend the length of our signal:

```
In [4]: signoise = np.random.rand(20) - 0.5 + np.hstack([np.zeros(4), sig, np.zeros(5)])
        plt.plot(signoise)
        plt.xlabel('Sample')
        plt.ylabel('Value')
        plt.title('The Example Signal in Noise')

Out[4]: <matplotlib.text.Text at 0x8bdbd70>
```



- Now we apply our matched filter to it:

```
In [5]: h = sig[::-1] # fliplr
        signoisemf = sp.lfilter(h, 1, signoise)
        plt.plot(signoisemf)
        plt.xlabel('Sample')
        plt.ylabel('Value')
        plt.title('The Example Signal in Noise after Macthed Filtering')

Out[5]: <matplotlib.text.Text at 0x8d59110>
```



This is now the output of our matched filter. We can see that we have a maximum at time 14, which marks the end of our **detected signal**. Hence we know that the signal started at sample 14-L(length of the filter) = 14-11=3, which was indeed the case since we added 4 zeros in the beginning. So matched filtering did a good job!

The **matched filtering** process can also be viewed as computing the **correlation** of the noisy signal with the original signal.