Example 2 Filter

January 25, 2017

0.1 The filtering operation itself works similarly in Matlab or Octave or Python, in the time domain. The function is "filter" or "lfilter". Given an input signal in the vector x, and filter coefficients in vectors A and B, the filtered output y of our filter is simply:

0.2

$$y = lfilter(B, A, x)$$

Use the function lfilter to obtain the impulse response of this IIR filter.

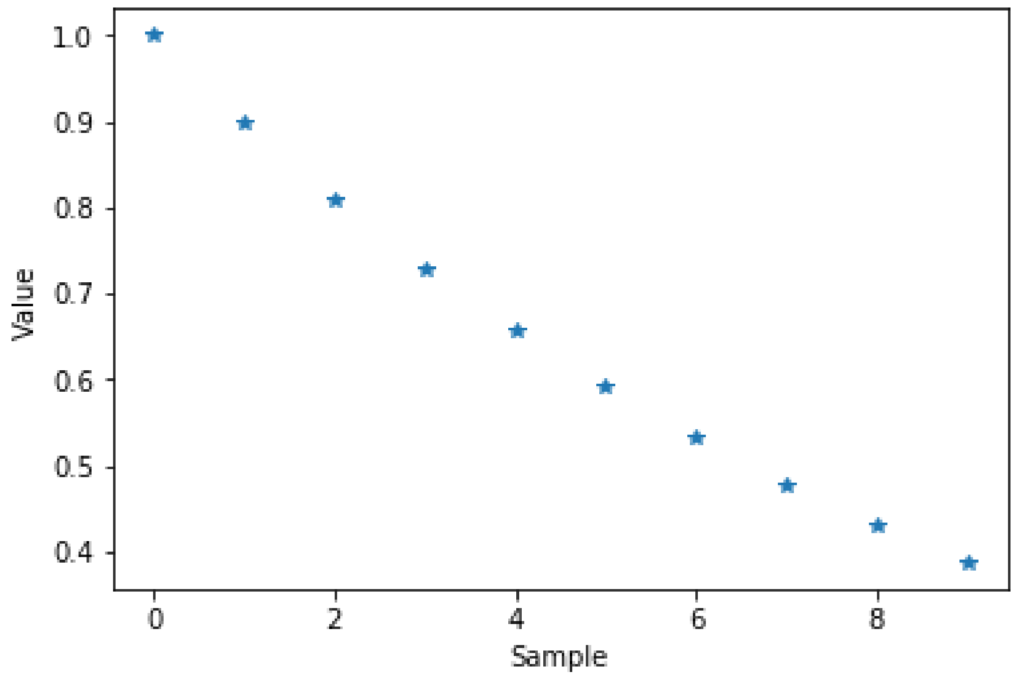
```
In [1]: %matplotlib inline
        import scipy.signal
        import matplotlib.pyplot as plt
        import numpy as np

        #Startăwithăaăunităpulseăasăinputăx:
        x = np.zeros(10)
        x[0] = 1

        #BăandăAăareăgivenăasăbefore:
        A = [1, -0.9]
        B = [1]

        #Nowăcalculateătheăimpulseăresponse:
        y = scipy.signal.lfilter(B, A, x)
        plt.plot(y, '*')
        plt.xlabel('Sample')
        plt.ylabel('Value')
```

```
Out[1]: <matplotlib.text.Text at 0x8dc9510>
```



Here we can see the indeed exponetial decaying function (the sequence $ir(n) = p^n$ for p=0.9). In this way we can also test more complicated IIR filters. This exponential decaying impulse response again shows the stability of the filter, which was to be expected because the pole of its transfer function in the z-domain is placed at z=0.9, and hence inside the unit-circle!