Digital Signal Processing EE 351M
Prof. Haris Vikalo HW #5 Part 1

Solutions to Homework #5 Part 1

1. (**5 pts**) *Suppose we would like to identify an unknown LTI system. Denote its impulse and frequency response by* $h[n]$ *and* $H^{(}e^{j\omega})$*, respectively. We will rely on an adaptive filtering method that employs transversal filter of length* $m = 50$*, and LMS algorithm with parameter* $\eta = 0.01$*. The length of the signal* $x[n]$ *applied to the inputs of both the unknown system and the transversal adaptive filter is* 1000 *samples. For this problem, you need to download file 'hw5data', which contains* $x[n]$ *and the output of the unknown system* $d[n]$*.*

   (a) *Plot the coefficients of the transversal filter at the end of the adaptive filtering procedure (use 'stem' command). This is your estimate of the impulse response* $h[n]$*.*
   (b) *Plot the first 500 samples of the squared error* $e^2[n]$ *generated by the LMS algorithm (use 'stem' command).*
   (c) *Do you think* $h[n]$ *is an IIR or an FIR system?*

   (a) We implement the transversel filter with the following Python code:

```python
import numpy as np
import matplotlib.pyplot as plt
import scipy
from scipy.io import loadmat
%matplotlib inline
```

```python
# Read the data file
data = loadmat('hw5mat.mat')
dt = data['d'].T
xt = data['x'].T
```

```python
# Implement the LMS algorithm with M=50 taps
time = len(dt)
M = 49
mu = 0.02
w = np.ones([M+1,1])
x_t = np.zeros([M+1,1])
error = np.zeros([time,])
squared_error = np.zeros([time,])
for i in range(time):
    if(i<M+1):
        x_t[:i+1] = xt[i+1:0:-1]
    else:
        x_t = xt[i-M:i+1]
        x_t = np.flipud(x_t)
    error[i] = dt[i]-np.dot(w.T,x_t) # error at time i
    grad = -x_t*error[i] # gradient at time i
    w = w - mu*grad # update the weights according to the gradient
    squared_error[i] = error[i]**2 # store the squared error at time t
```

   We plot the weights of the transversal filter using the following code in Figure 1.

```python
plt.stem(w)
plt.title('Weights of Lenght 50 Transversal Filter')
plt.xlabel('Index of Tap')
plt.ylabel('w')
```

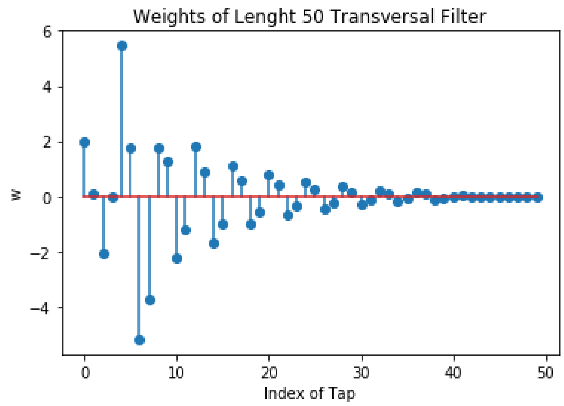

Figure 1: Weights of the transversal filter of lenght 50.

(b) We plot the squared error over first 500 steps using the following code in Figure 2.

```
plt.stem(squared_error[:500])
plt.title('Squared Error over Time')
plt.xlabel('Time')
plt.ylabel('Squared Error')
```

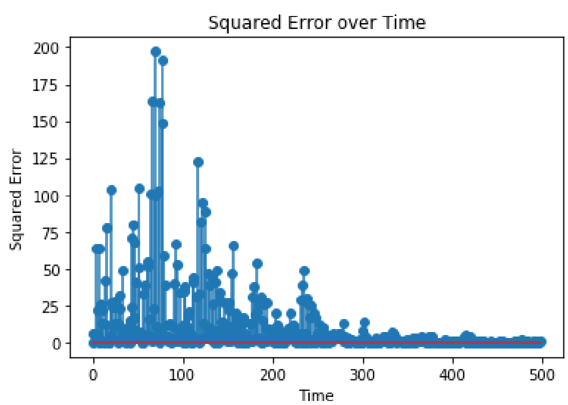

Figure 2: Squared Error for first 500 steps.

(c) The system is FIR - coefficients of the transversal filter diminish for higher-valued indices. [In fact, the LTI system is a 40-samples long FIR fillter.]

2. (**3 pts**) *Find the equation for the update of the coefficient $w[n]$ in the leaky LMS algorithm.* We define the error at time n as

$$\begin{aligned} L &= \frac{1}{2}e^2[n] + \frac{1}{2}\lambda w[n]^T w[n] \\ &= \frac{1}{2}(d[n] - u[n]^T w[n])^2 + \frac{1}{2}\lambda w[n]^T w[n]. \end{aligned} \tag{1}$$

The gradient corresponding to loss defined in (1) is

$$\nabla_w L = -(d[n] - u[n]^t w[n])u[n] + \lambda w[n].$$

Then, we write the Leaky-LMS update as

$$\begin{aligned} w[n+1] &= w[n] - \mu \nabla_w L \\ &= w[n] - \mu\lambda w[n] + \mu u[n](d[n] - w[n]^T u[n]) \\ &= (1 - \mu\lambda) w[n] + \mu u[n](d[n] - u[n]^T w[n]). \end{aligned}$$