Assignment 1
EE 6543 taught at UMSA
Adaptive Signal Processing
Ingeniería Electrónica
Universidad Mayor de San Andrés

Instructor:
Brent Petersen Ph.D. P.Eng.
b.petersen@ieee.org

No.	Mark
1	/10
2	1
Total	/11

Problem 1. ____ mark(s) / 10 mark(s)

A learning curve is a plot of $10\log_{10}\left(|e(n)|^2\right)$ versus n, where $e(n)$ is the error between the desired signal and the estimate. n is the time index.

For module SA (Simple Adaptive), a Monte Carlo simulation, has been provided.

Change the line:

```
sigman1     =  10^(-2) ;
```

to:

```
sigman1     =  10^(-5) ;
```

Run the MATLAB®or GNU Octave program. Submit the code and the output plot. Describe the change in the learning curve.

Problem 2. ____ mark(s) / 1 mark(s)

A learning curve is a plot of $10\log_{10}\left(|e(n)|^2\right)$ versus n, where $e(n)$ is the error between the desired signal and the estimate. n is the time index. For module SA (Simple Adaptive), a Monte Carlo simulation, has been provided. Howver, a few minor changes were made. One major change is that the no channel noise added to the receiver input signal. The program is shown here and is available online. Run the MATLAB®or GNU Octave program. The program is shown here.

Describe why the learning curve flattens out, even though there is no channel noise added.

The ordinates, also known as the y axis, are the squared error, expressed in deciBels. The negative of that particular value where the learning curve flattens is analogous to the signal-to-noise (SNR) ratio. For comparison purposes, the SNR of compact disc audio is 96 (dB). A radio signal transmitted across the Milky Way Galaxy would experience an attenuation of about 440 (dB). Describe why the curve flattens out at such a low value. Why is it that particular value?

```
% This is the adaptation in module SA, but with no noise added.

clear ;
clear functions ;
clf ;

Nbits      =  2^7 ;  % (-), number of bits per user.
mu         =  2^(-1) ;  % (-), LMS adaptation constant
Npoints_w  =  1 ;  % (-), number of points in w
D          =  0 ;  % (T-spaced samples), Decoding delay
a          =  1 ;  % Transmitter attenuation
b          =  1 ;  % Receiver attenuation

% transmitter data
d1    =   ( 2 * ( rand(1,Nbits) < 0.5 ) - 1 ) ;

% i = 0:(Nbits-1) ;
% stem(i,d1,'o') ;

s1 = a * b * d1 ;

% There is no noise added.
r1  =  s1 ;

w   =  zeros(1,Npoints_w) ;
rn  =  zeros(1,Npoints_w) ;

errors1   =   0 * d1 ;

wsave = [] ;
for i  =  1 : length(r1) ,
    if   ( 1 <= (i-D) )  &  ( (i-D) <= length(d1) )  ,

        rn  =  [ r1(i)  rn( 1 : (Npoints_w-1) ) ] ;
        u1  =  w * (rn .') ;

        e1  =  d1( i - D ) - u1 ;
        errors1(i-D) = e1 ;

        w   =   w  +  mu * e1 * rn ;
        wsave  =  [ wsave w ] ;
    end
end

iw = [ 0 : (length(w)-1) ] ;

subplot(311) ;
```

```
stem(iw,abs(w).^2,'o') ;
ylabel('|w|^2') ;
xlabel('Index, i, (-)') ;

subplot(312) ;
se_db  =  20 * log10(abs(errors1)+eps) ;
plot(se_db,'o') ;
% axis ( [ 0 (Nbits-1) -40 0] ) ;

subplot(313) ;
stem(wsave)
```