北京理工大學
Beijing Institute of Technology

本科实验报告

实验名称：LMS 算法在自适应滤波器中的性能分析

课程名称：	自适应信号处理	实验时间：	2019.04.28
任课教师：	许文龙	实验地点：	
实验教师：	许文龙	实验类型：	■原理验证 □综合设计 □自主创新
学生姓名：	刘仕聪		
学号/班级：	1120161380	第一组：	
学　院　：	信息与电子学院	同组搭档：	
专　业　：	电子信息类（实验班）	成　绩　：	

信息与电子学院
SCHOOL OF INFORMATION AND ELECTRONICS

LMS 算法在自适应滤波器中的性能分析

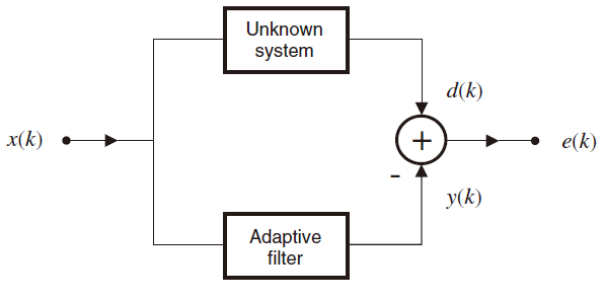
一、 实验目的

1. 完成 LMS 算法的仿真（系统辨识）
2. LMS 算法性能分析（包括时变）

二、 实验原理

1. 自适应滤波器系统辨识

系统辨识是自适应滤波器的一个重要应用。典型原理框图如下



如图所示，输入信号同时输入两个系统，得到信号的误差可以进行参数修正。对一个自适应滤波器而言，假设未知系统冲激响应无限长，则可以用$\boldsymbol{h}_\infty$表示。误差信号表示为

$$e(k) = d(k) - y(k) = \sum_{l=0}^{\infty} h(l)x(k-l) - \sum_{l=0}^{N} w_i(k)x(k-i)$$

此时均方误差为

$$\begin{aligned}\xi &= E[e^2(k)] = E\{[\boldsymbol{h}^T\boldsymbol{x}_\infty(k) - \boldsymbol{w}^T\boldsymbol{x}_{N+1}(k)]^2\} \\ &= E\left[\boldsymbol{h}^T\boldsymbol{x}_\infty(k)\boldsymbol{x}_\infty^T(k)\boldsymbol{h} - 2\boldsymbol{h}^T\boldsymbol{x}_\infty(k)\boldsymbol{x}_{N+1}(k)\boldsymbol{w} + \boldsymbol{w}^T\boldsymbol{x}_{N+1}(k)\boldsymbol{x}_{N+1}^T(k)\boldsymbol{w}\right] \\ &= \sigma_x^2\sum_{i=0}^{\infty} h^2(i) - 2\sigma_x^2\boldsymbol{h}^T\begin{bmatrix}\boldsymbol{I}_{N+1} \\ \boldsymbol{0}_{\infty\times(N+1)}\end{bmatrix}\boldsymbol{w} + \boldsymbol{w}^T\boldsymbol{R}_{N+1}\boldsymbol{w}\end{aligned}$$

采用无穷标注的为无穷多项，此时系统一定会出现残余误差。当我们最小化均方误差时，我们就会得到最佳滤波器抽头系数$\boldsymbol{w_0} = \boldsymbol{h_{N+1}}$，其中

$$\boldsymbol{h}_{N+1} = \boldsymbol{h}^T\begin{bmatrix}\boldsymbol{I}_{N+1} \\ \boldsymbol{0}_{\infty\times(N+1)}\end{bmatrix}$$

如果实际中没有测量噪声（和信道噪声等），则最终将得到最小 MSE 为 0。实际应用中，系统辨识常用于多径通信信道建模、控制系统、地震探测，某些通信系统的杂波回音消除等。本次仿真主要分析 LMS 算法在系统辨识中的若干性能。

2. LMS 算法

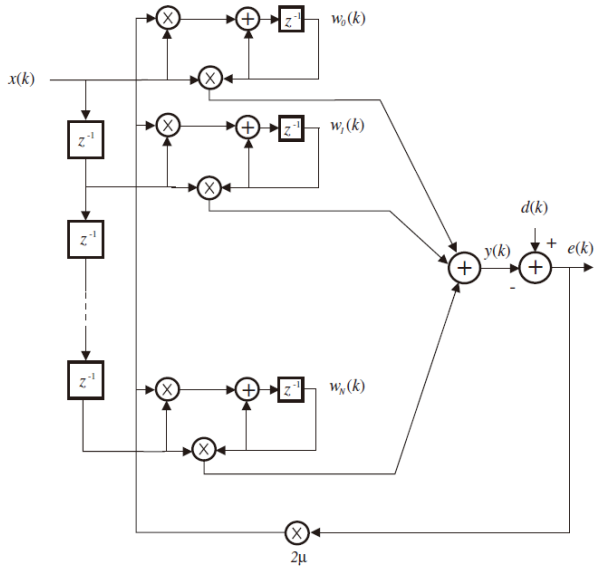
LMS 算法是自适应算法的一种实现方式。由于理论算法中梯度方向无法计算等原因，人们采用 LMS（最小均方误差）算法进行实现。LMS 算法对理论算法的改进是，采用瞬时值来代替理论中的所有值，采用瞬时采样到的信号来估计梯度，则

$$\begin{aligned}\hat{\boldsymbol{g}}_{\boldsymbol{w}}(k) &= -2d(k)\boldsymbol{x}(k) + 2\boldsymbol{x}(k)\boldsymbol{x}^T(k)\boldsymbol{w}(k) \\ &= -2e(k)\boldsymbol{x}(k)\end{aligned}$$

如果根据梯度计算误差函数，易知误差函数为$e^2(k)$。至此 LMS 算法的定义已经比较清楚，我们可以认为是采用瞬时误差代替了均方误差的算法。如果将该算法带入到梯度下降法中，我们得到

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) + 2\mu e(k)\boldsymbol{x}(k)$$

其中采用了一个控制因子对收敛进行控制。实现框图为



经过简单推导，我们容易知道，LMS 算法在统计平均上，梯度方向收敛于 MSE 算法时的梯度方向，系数向量收敛条件为

$$0 < \mu < \frac{1}{\lambda_{max}}$$

稳定性条件为

$$0 < \mu < \frac{1}{tr[\boldsymbol{R}]}$$

且在大部分情况下收敛因子应取不接近上界的值。最小误差与待辨识系统的冲激响应与测量噪声有关。当自适应滤波器抽头数量高于目标系统且不存在测量噪声的条件下时，最小误差为 0。实际上每一次迭代都需要一定的步长，而不是连续进行的，因此存在超量误差。分析可知，超量 MSE 可以描述为

$$\xi_{exc} = \frac{\mu\sigma_n^2 tr[\boldsymbol{R}]}{1 - \mu tr[\boldsymbol{R}]}$$

3. 有色噪声

一般认为，白噪声是在频谱中呈现为均匀分布的噪声，与时域分布无关。色噪声在频域就不是均匀分布的，我们认为有色噪声可以通过白噪声通过一阶全极点滤波器得到，例如传递函数为

$$H(z) = \frac{z}{z - a}$$

色噪声可以看做是一个 AR 信号，是白噪声信号通过自回归模型得到的。

4. 有限精度误差（量化效应）

在实际实现中，我们采用的存储器，运算器等都是存在精度的，这将导致我们的运算出现误差，即量化误差。此次课程中并没有对有限字长效应做出介绍，此处也不对原理进行分析，仅列出仿真结果进行分析。

需要注意的是，当采用有限精度实现的时候，由于收敛因子的过小选取，有时往往会带来更大的超量误差。

5. 非平稳问题性能分析

实际问题往往都是非平稳问题，对于可变的目标响应，此时滤波器系数的滞后会带来之后超量误差，为了计算超量误差，我们假设最优系数向量的每一个元素都可以用一个马尔科夫过程来描述，这种非平稳情形可以看做是一种简化。

$$\boldsymbol{w}(k) = \lambda_{\boldsymbol{w}}\boldsymbol{w}(k-1) + \boldsymbol{n}_{\boldsymbol{w}}(k)$$

通过一些巧妙的数学方法，我们能推导出超量误差等参数的近似表达式。

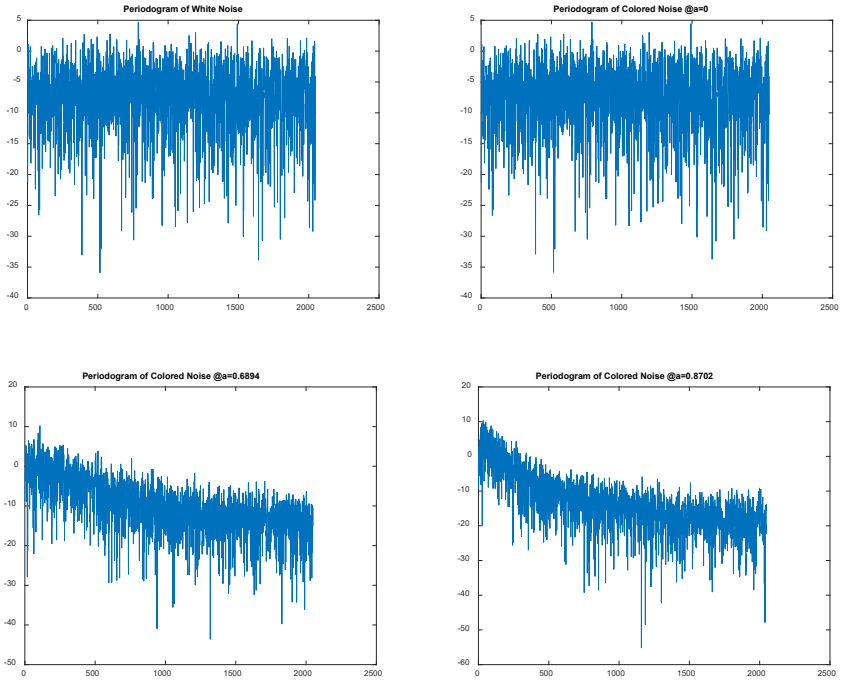
三、 实验内容

a. 运行算法对每种特征值扩展的情形进行分析评价

例题中假设信号是色噪声输入，相当于白噪声信号通过一个一阶滤波器产生。如果滤波器传递函数为

$$H(z)=\frac{z}{z-a}$$

此处首先给出噪声的功率谱。白噪声与三种色噪声：



根据图中的结果，我们可以推测，当特征值扩展增大的时候，噪声的高频削减越严重。实际上是这样的，当极点越来越大时，我们相当于做了越来越不明显的指数变换（根据反 z 变换可以推知），而指数接近 1，下降越不明显，因此低频分量占比就会越大。

则此时信号的相关矩阵可以描述为

$$R=\frac{\sigma_v^2}{1-a^2}\begin{bmatrix}1 & a & a^2\\ a & 1 & a\\ a^2 & a & 1\end{bmatrix}$$

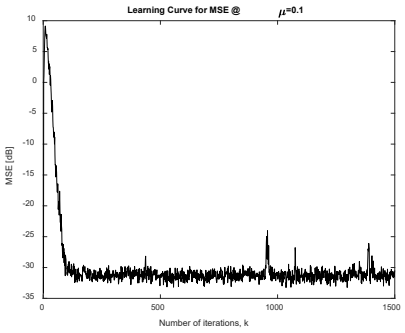
当然对于本题采用七阶滤波器，应有八个抽头，相关函数应该描述为 8 个系数。此处简述为三个系数。假设阶数较低，我们可以计算特征值扩展，但是对于如此高阶数的结果，这里不易直接计算特征值扩展。根据书中文献 [20] 的 124 页证明，我们知道特征值扩展可以近似描述为

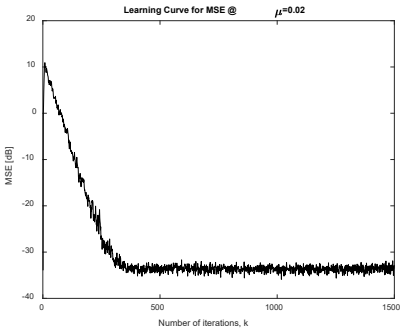
$$\frac{\lambda_{max}}{\lambda_{min}}\sim\left\{\frac{1+a}{1-a}\right\}^2$$

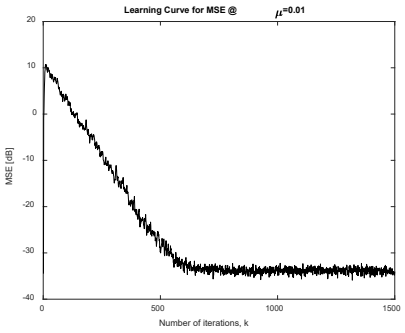
利用上述关系进行计算时不完全准确的。准确结果参考为$a=0.6894$和$a=0.8702$。考虑到输入信号的功率为1，因此可以搭建仿真平台进行运算。为了保证收敛，此处选择收敛因子为$\mu=\mu_{max},\frac{\mu_{max}}{5},\frac{\mu_{max}}{10}$。此处计算

$\mu_{max} = 0.125$，考虑到书中给出的稳定性原则，$\mu_{max} < \frac{1}{10\lambda_{max}}$，因此此处选为$\mu_{max} = 0.02$。

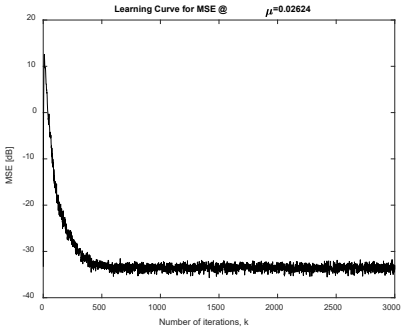
对于特征值扩展为 1 的条件，进行三组μ的仿真。实验中采用 100 次过程进行平均。

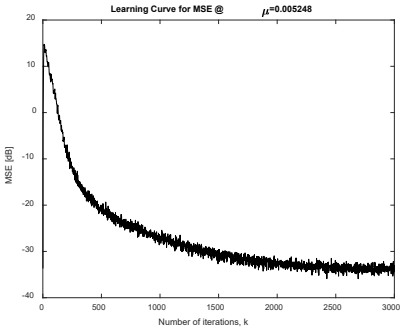


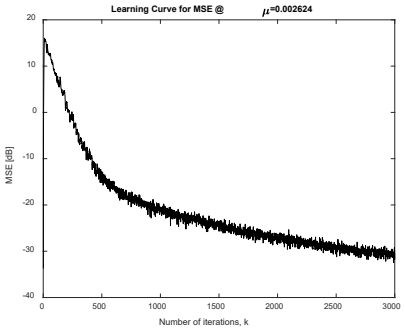




如图所示，三者分别在约为 80 次、300 次与 600 次迭代后达到收敛。增大特征值扩展为 20 时，得到







采用相同的迭代次数进行仿真时，我们发现最后一种情况还没达到收敛。此时对比已经十分明显，在特征值扩展越大（信号变得差，存在较高互相关）的条件下，收敛就变得越困难。而且收敛时的次数比例也与收敛系数的反比不成关系（相比之下白噪声时收敛迭代次数与收敛系数基本为反比关系）。

b. 对每个情形下失调进行计算，并与理论结果进行比较

失调量，是超量误差与误差的比值。

$$M = \frac{\mu(N+1)\sigma_x^2}{1-\mu(N+1)\sigma_x^2}$$

与课本类似的，这里列出表格进行统计

a	步长		理论偏差	仿真偏差
$a=0$	μ_{max}	0.1	3.4634	4.5132
	$\frac{\mu_{max}}{5}$	0.02	0.1928	0.2214
	$\frac{\mu_{max}}{10}$	0.01	0.0905	0.0883
$a=0.6894$	μ_{max}	0.0262	0.2653	0.2247
	$\frac{\mu_{max}}{5}$	0.0052	0.0437	0.0477
	$\frac{\mu_{max}}{10}$	0.0026	0.0216	0.0114
$a=0.8702$	μ_{max}	0.0173	0.1613	0.2577
	$\frac{\mu_{max}}{5}$	0.0034	0.0248	0.0544
	$\frac{\mu_{max}}{10}$	0.0017	0.0140	0.0360

c. 考虑采用定点数算法，在一组实验中运行算法，计算特定情境下的均方损失。

这里采用定点数相当于采用了指定的量化算法。根据本书中推导的定点数量化效应算法，我们可以得到

$$E\left[\left\|\Delta \boldsymbol{w}(k)_Q\right\|^2\right] = \frac{\mu(\sigma_n^2+\sigma_c^2)(N+1)}{1-\mu(N+1)\sigma_x^2} + \frac{(N+1)\sigma_w^2}{4\mu\sigma_x^2[1-\mu(N+1)\sigma_x^2]}$$

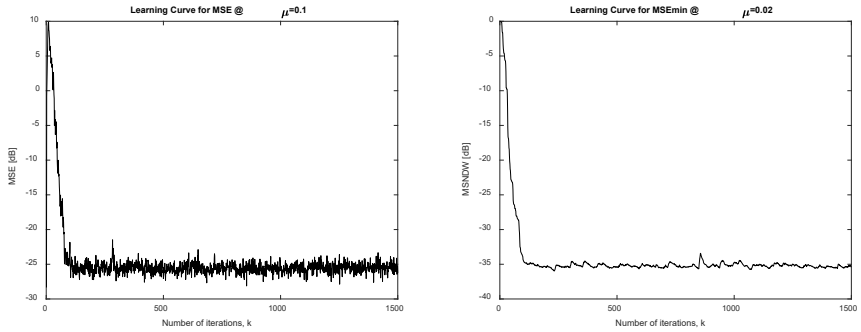
且有

$$\xi(k)_Q = \frac{\sigma_e^2 + \sigma_n^2}{1 - \mu(N+!)\sigma_x^2} + \frac{(N+1)\sigma_x^2}{4\mu[1 - \mu(N+!)\sigma_x^2]}$$

在$\mu = 0.1$条件下仿真得到的结果为

比特数	$\xi(k)_Q$		$E\left[\left\|\Delta \boldsymbol{w}(k)_Q\right\|^2\right]$	
	实验值	理论值	实验值	理论值
16	$3.1e-4$	$3.1e-4$	0.018	0.017
12	$2.9e-4$	$3.1e-4$	0.018	0.017
10	$3.3e-4$	$3.2e-4$	0.019	0.018

量化误差带来的变化很小。其中一次结果的效果如下



d. 对于下述情况重复试验

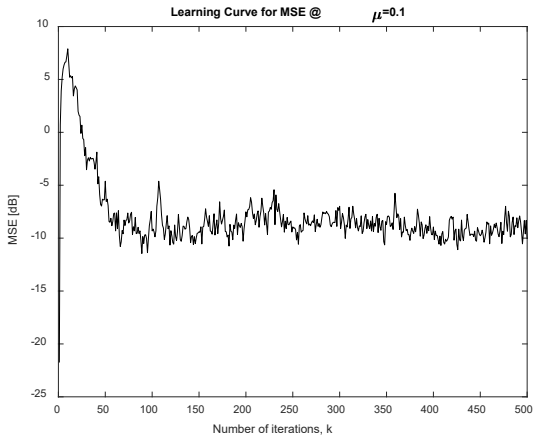
该部分结果都在上文中。实际上我们发现，在量化点数足够大时，量化精度对结果影响不大。

e. 假设未知系统时变。

我们的假设是，滤波器系数是一个马尔科夫过程，此时迭代方程为

$$\boldsymbol{w}(k) = \lambda_{\boldsymbol{w}} \boldsymbol{w}(k-1) + n_{\boldsymbol{w}}(k)$$

根据题目假设，我们取$\lambda_{\boldsymbol{w}} = 0.99$，$\sigma_{\boldsymbol{w}}^2 = 0.0015$，测量噪声增大到$\sigma_{\boldsymbol{n}}^2 = 0.01$，此时仿真得到



可以看到 MSE 在-10dB 左右位置未再下降，系统 MSE 收敛在了此处。实验结果中理论超量 MSE 大约为0.08，实验结果约为0.128，存在一定误差。

四、 问题与思考

1. 仿真结果与实际并不完全相同

   本次实验中一个比较眼中的问题就是，仿真结果与实际计算存在不小的误差。根据书本中的相似实验仿真，实际的结果与理论的结果应没有什么差异，但是当我自己仿真的时候，得到的结果却与课本不同。

   相同的问题也发生在了许多其他同学身上，对此结果我们仍没有找到合适的解释。

2. 由于 LMS 算法本身的特点，在应对非平稳问题时的效果要比平稳问题的效果差的多（大约有 20dB 的 MSE 差距）

3. 本次仿真一并验证了特征值扩展对收敛速度的影响。实际上特征值扩展不同相当于噪声不同，此处采用色噪声，自适应滤波器在收敛变慢的条件下仍然能够得到令人满意的结果。

五、 附录

本次仿真的代码如下

```
%% P77 e.g. 3.5

%% Question (a)+(b)

L           =   1500;               % Iterations and Siganal
length
a           =   0;                  %a = 0, 1; a = 0.6894, 20; a
= 0.8702, 80
```

```
sigma_v2    =   1-a^2;              % Make sure colored noise
power = 1
sigma_n2    =   0.0001;             % measure noise
R           =   [1,a,a^2,a^3,a^4,a^5,a^6,a^7;
                 a,1,a  ,a^2,a^3,a^4,a^5,a^6;
                 a^2,a,1,a  ,a^2,a^3,a^4,a^5;
                 a^3,a^2,a,1,a  ,a^2,a^3,a^4;
                 a^4,a^3,a^2,a,1,a  ,a^2,a^3;
                 a^5,a^4,a^3,a^2,a,1,a  ,a^2;
                 a^6,a^5,a^4,a^3,a^2,a,1,a  ;
                 a^7,a^6,a^5,a^4,a^3,a^2,a,1];
[A,B]       =   eig(R);


v=sqrt(sigma_v2)*(randn(L,1));      % White Noise
n=sqrt(sigma_n2)*(randn(L,1));

plot(10*log10(periodogram(v)),'-k');
title('Periodogram of White Noise')
x=zeros(L,1);
for i=1:L-1
    x(i+1)=a*x(i)+v(i);             % Generate Colored Noise
end
figure
plot(10*log10(periodogram(x)),'-k');
title(['Periodogram of Colored Noise @a=' num2str(a)])

%   Definitions:
ensemble    = 100;                   % number of
realizations within the ensemble
K           = L;                     % number of iterations
H           = [0.1 0.3 0,-0.2 -0.4 -0.7 -0.4 -0.2].';
Wo          = H;                     % unknown system
N           = 8;                     % number of
coefficients of the adaptive filter
mu          = 1/B(8,8)/(N+2)/10                  %
convergence factor (step)  (0 < mu < 1)

%   Initializing & Allocating memory:
W       = ones(N,(K+1),ensemble);   % coefficient vector for
each iteration and realization; w(0) = [1 1 1 1].'
MSE     = zeros(K,ensemble);        % MSE for each realization
MSEmin  = zeros(K,ensemble);        % MSE_min for each
realization
```

```
M=0;
%   Computing:
for l=1:ensemble,

    X       = zeros(N,1);               % input at a certain
iteration (tapped delay line)
    d       = zeros(1,K);               % desired signal
    n=sqrt(sigma_n2)*(randn(L,1));
    v=sqrt(sigma_v2)*(randn(L,1));
    x=zeros(L,1);
    for i=1:L-1
        x(i+1)=a*x(i)+v(i);
    end
    % nw=sqrt(sigma_w2)*randn(1,K);

    for k=1:K,
        X       =   [x(k,1)
                     X(1:(N-1),1)];             % input signal
(tapped delay line)
        % Wo=lambda*Wo+nw(k);                    % For Non-
stationary
        d(k)    =   (Wo'*X(:,1))+n(k);          % desired signal
    end
    S   =   struct('step',mu,'filterOrderNo',(N-
1),'initialCoefficients',W(:,1,l));
    [y,e,W(:,:,l)]  =   LMS(d,transpose(x),S);
    MSE(:,l)    =   MSE(:,l)+(abs(e(:,1))).^2;
    MSEmin(:,l) =   MSEmin(:,l)+(abs(n(:))).^2;
    M=M+mu*8*var(x)/(1-mu*8*var(x));

end
%   Averaging:
W_av = sum(W,3)/ensemble;
MSE_av = sum(MSE,2)/ensemble;
MSEmin_av = sum(MSEmin,2)/ensemble;

Mreal=(sum(MSE_av(K-99:K))/100-
mean(MSEmin_av))/mean(MSEmin_av)

M=M/ensemble

%   Plotting:
figure,
plot(1:K,10*log10(MSE_av),'-k');
```

```
title(['Learning Curve for MSE @\mu=' num2str(mu)]);
xlabel('Number of iterations, k'); ylabel('MSE [dB]');

figure,
plot(1:K,10*log10(MSEmin_av),'-k');
title(['Learning Curve for MSEmin @\mu=' num2str(mu)]);
xlabel('Number of iterations, k'); ylabel('MSEmin [dB]');

figure,
subplot 211, plot(real(W_av(1,:))),...
title(['Evolution of the 1st coefficient (real part)@\mu='
num2str(mu)]);
xlabel('Number of iterations, k'); ylabel('Coefficient');
subplot 212, plot(imag(W_av(1,:))),...
title(['Evolution of the 1st coefficient (imaginary
part)@\mu=' num2str(mu)]);
xlabel('Number of iterations, k'); ylabel('Coefficient');
```

对于后面的非平稳情况，只需要在生成目标信号的时候将每一次迭代的目标向量进行一定更改即可。