

## DSP II 第二次计算机作业

姓名：汤宁标

学号：SC22219139

1、Consider an AR process  $x(n)$  defined by the difference equation

$$x(n) = -a_1x(n-1) - a_2x(n-2) + v(n)$$

where  $v(n)$  is an additive white noise of zero mean and variance  $\sigma_v^2$ . The AR parameters  $a_1$  and  $a_2$  are both real valued:

$$a_1 = 0.1$$

$$a_2 = -0.8$$

a) Calculate the noise variance  $\sigma_v^2$  such that the AR process  $x(n)$  has unit variance. Hence, generate different realization of the process  $x(n)$ .

解：因为  $R(t) = E\{x(n)x(n-t)\}$ ,  $x(n) = -a_1x(n-1) - a_2x(n-2) + v(n)$

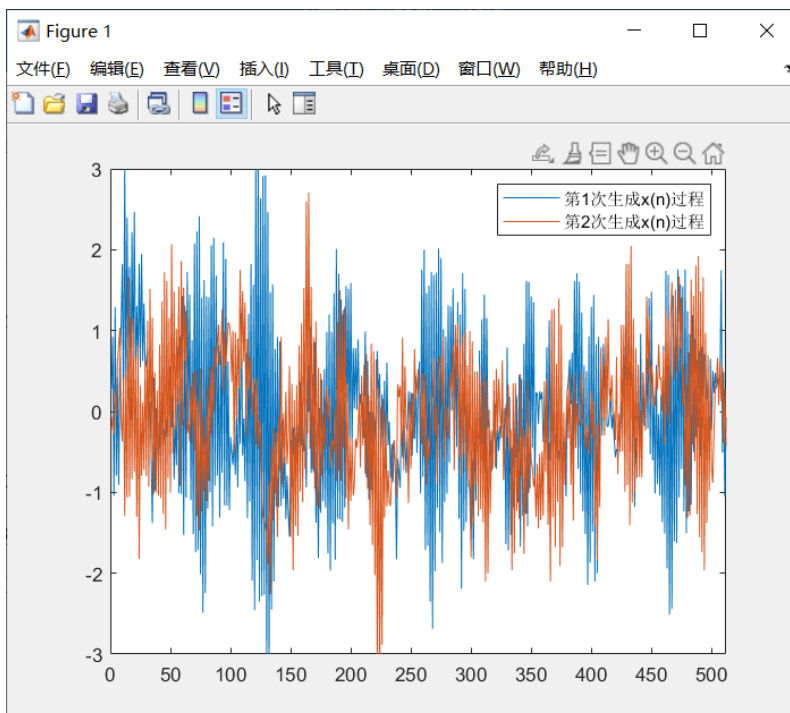
$$\text{所以 } R(1) = E\{x(n)x(n-1)\} = E\{x(n-1)[-a_1x(n-1) - a_2x(n-2) + v(n)]\} = -a_1R(0) - a_2R(1)$$

$$R(2) = E\{x(n)x(n-2)\} = E\{x(n-2)[-a_1x(n-1) - a_2x(n-2) + v(n)]\} = -a_1R(1) - a_2R(0)$$

代入数值  $R(0)=1$ ,  $a_1=0.1$ ,  $a_2=-0.8$ , 解得  $R(1)=-0.5$ ,  $R(2)=0.85$

$$\text{则 } \sigma_v^2 = R(0) - h_1R(1) - h_2R(2) = R(0) + a_1R(1) + a_2R(2) = 0.27$$

Matlab 生成  $x(n)$  图像如下所示 (N=512)



b) Given the input  $x(n)$ , an LMS filter of length  $M = 2$  is used to estimate the unknown AR parameters  $a_1$  and  $a_2$ . The step size  $\delta$  is assigned the value 0.05. Compute and plot the ensemble average curve of  $a_1$  and  $a_2$  by averaging the value of parameters  $a_1$  and  $a_2$  over an ensemble of 100 different realization of the filter. Calculate the time constant according to the experiment results and compare with the corresponded theoretical value.

解：LMS 算法公式如下

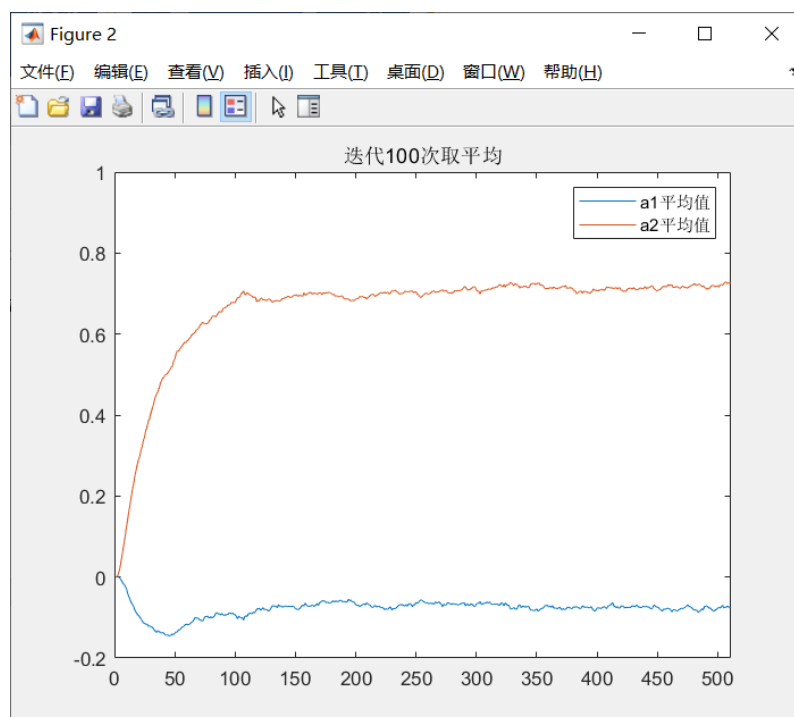
$$e(n+1) = y(n+1) - H^T(n)X(n+1)$$

$$H(n+1) = H(n) + \delta e(n+1)X(n+1)$$

$$X(n+1) = [x(n), \dots, x(n-M+1)]^T$$

$$H(n) = [a_1(n), a_2(n)]^T \text{ or } [h_1(n), h_2(n)]^T$$

迭代 100 次后  $a_1$ 、 $a_2$  的均值曲线如下所示



$$R_{xx} = \begin{bmatrix} R(0) & R(1) \\ R(1) & R(0) \end{bmatrix} = \begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix}$$

解得矩阵的两个特征值为  $\lambda_1=0.5$ ,  $\lambda_2=1.5$ , 则计算得理论时间常数为

$$\frac{1}{\delta\lambda_{\max}} = 13.3 < \tau_a < \frac{1}{\delta\lambda_{\min}} = 40$$

根据时间常数的定义，从  $a_1$ 、 $a_2$  曲线中各取两个点计算 ( $N=4$ 、 $9$ )

$$a(4) = [-0.0008, 0.0149]^T \quad a(9) = [-0.0235, 0.1036]^T$$

$$a(\infty) = [-0.0763, 0.7264]^T$$

根据公式

$$a(t_1) - a(\infty) = e^{-\frac{t_1 - t_2}{\tau}} [a(t_2) - a(\infty)]$$

解得  $\tau_{a1}=13.9509$ ,  $\tau_{a2}=37.5653$ , 与理论值相符

c) For one realization of the LMS filter, compute the prediction error

$$f(n) = x(n) - \tilde{x}(n)$$

And the two tap-weight errors

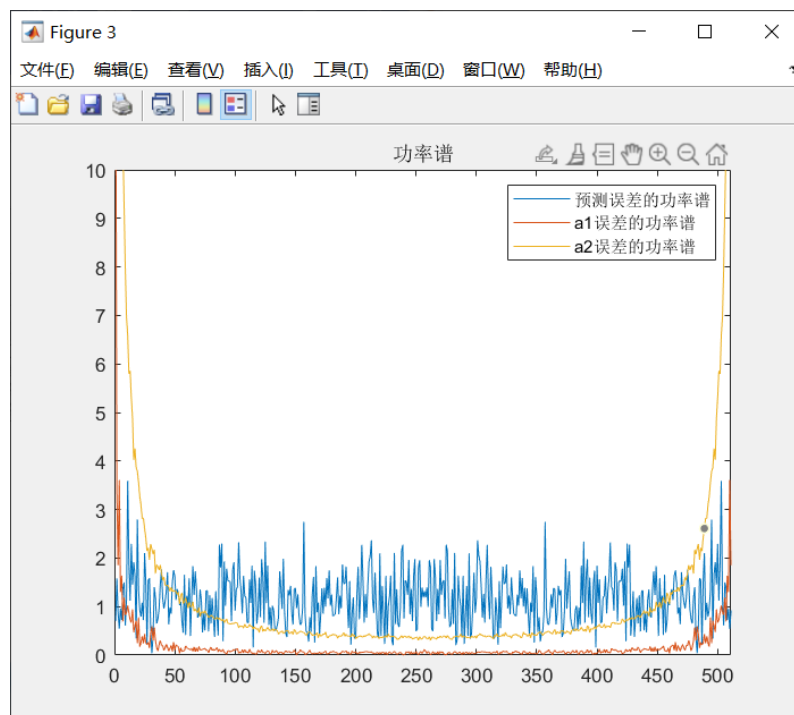
$$\varepsilon_1(n) = -a_1 - h_1(n)$$

and

$$\varepsilon_2(n) = -a_2 - h_2(n)$$

Using power spectral plots of  $f(n)$ ,  $\varepsilon_1(n)$  and  $\varepsilon_2(n)$ , show that  $f(n)$  behaves as white noise, whereas  $\varepsilon_1(n)$  and  $\varepsilon_2(n)$  behave as low-pass process. Explain the reason for this phenomenon.

解:  $f(n)$ ,  $\varepsilon_1(n)$ 和 $\varepsilon_2(n)$ 的功率谱如下图所示



由图像可知,  $f(n)$ 具有白噪声特性, 而 $\varepsilon_1(n)$ 和 $\varepsilon_2(n)$ 具有低通特性。

因为系数最终会收敛到 LMS 算法的最优值, 所以  $f(n)$ 会趋向于  $v(n)$ , 即具有白噪声特性。

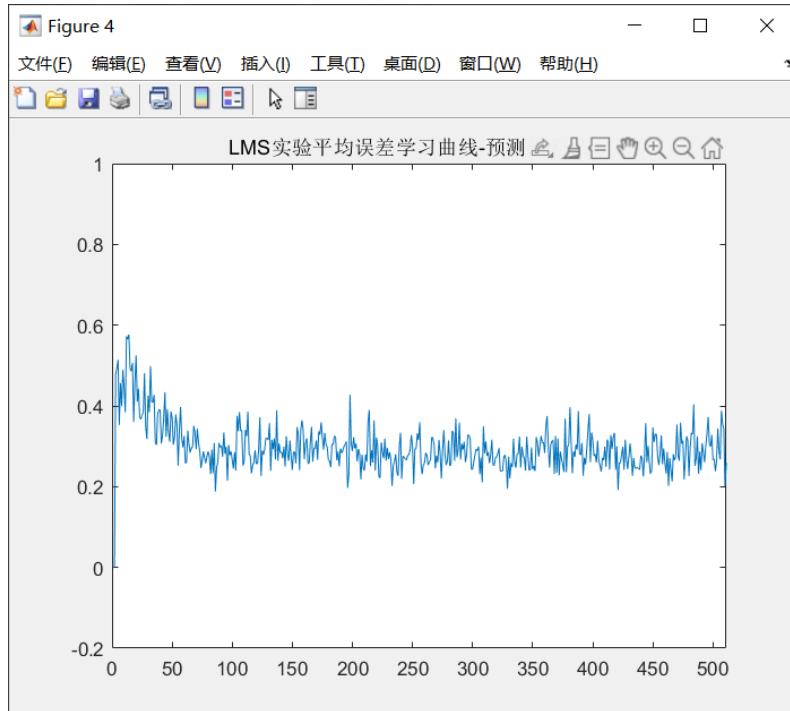
因为 LMS 滤波器的  $E[\alpha_k(n)] = (1 - \delta\lambda_k)^n E[\alpha_k(0)]$ ，系数收敛到最优值  $a_1$ 、 $a_2$  对上式做离散傅里叶变换得

$$S(w) = \frac{E[\alpha_k(0)]}{1 - (1 - \delta\lambda_k)e^{-jw}}$$

易知  $E[\alpha_k(n)]$  具有低通特性，因为  $\alpha(n) = Q^T[H(n) - H_{opt}]$ ，即  $\alpha_i(n)$  是  $h_i(n)$  的线性组合，所以  $h_i(n)$  具有低通特性，从而  $\varepsilon_1(n)$  和  $\varepsilon_2(n)$  具有低通特性。

d) Compute the ensemble average learning curve of the LMS filter by averaging the square value of the prediction error  $f(n)$  over an ensemble of 100 different realization of the filter. Calculate the time constant and residual power according to the experiment results and compare with the corresponded theoretical values.

解：LMS 算法 100 次平均学习曲线如下图所示



理论上 LMS 算法均方收敛的时间常数为  $\tau_k = \frac{1}{2\delta\sigma_x^2} = 10$ ，均方收敛的误差为

$$J(\infty) = \frac{J_{min}}{1 - \frac{\delta}{2} N \sigma_x^2}, \text{ 其中}$$

$$J_{min} = R(0) - H_{opt}^T, \quad H_{opt} = R_{xx}^{-1} R_{yx},$$

$$R_{xx} = \begin{bmatrix} R(0) & R(1) \\ R(1) & R(0) \end{bmatrix}, \quad R_{yx} = [R(1) \quad R(2)]^T$$

解得理论均方收敛误差为  $J_{min}=0.27$ ， $J(\infty)=0.2842$

实验中,  $J(\infty)$ 波动较大, 取收敛后的 200 个点均值 $J(\infty)=0.2828$

取  $J(20)=0.4595$ ,  $J(40)=0.3059$ , 根据公式

$$J(t_1) - J(\infty) = e^{-\frac{t_1-t_2}{\tau}} [J(t_2) - J(\infty)]$$

解得 $\tau=9.8251 \approx 10$ , 实验值与理论值相符。

超量误差 $J_{ex} \approx J(\infty) - J_{min}$ , 则理论超量误差为 $J_{ex} \approx 0.2842 - 0.27 = 0.0142$ , 实

验超量误差为 $J_{ex} \approx 0.2828 - 0.1993 = 0.0835$

2. 设 $x(n) = x_1(n) + x_2(n)$ ,  $x_1(n)$ 是窄带信号, 定义 $x_1(n) = \sin(0.05\pi n + \phi)$ ,  $\phi$ 是在 $[0, 2\pi]$ 区间上均匀分布的随机相位。 $x_2(n)$ 是宽带信号, 它是一个零均值、方差为 1 的白噪音信号  $e(n)$ 激励一个线性滤波器而产生, 其差分方程为 $x_2(n) = e(n) + 2e(n-1) + e(n-2)$ 。

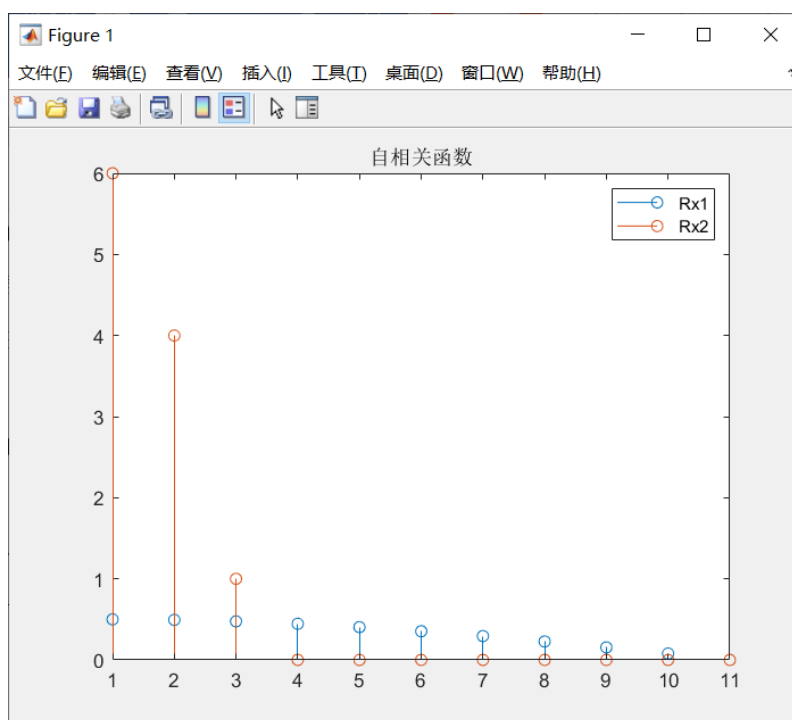
(1) 计算 $x_1(n)$ 和 $x_2(n)$ 各自的自相关函数, 并画出其函数图形。根据此选择合适的延时, 以实现谱线增强。

解:  $R_{x1}(k) = E[x_1(n)x_1(n-k)] = E[\sin(0.05\pi n + \phi) \sin(0.05\pi(n-k) + \phi)] = 0.5\cos(0.05\pi k)$

$$R_{x2} = E[x_2(n)x_2(n-k)] = E\{[e(n) + 2e(n-1) + e(n-2)][e(n-k) + 2e(n-k-1) + e(n-k-2)]\}$$

$$= \delta(k+2) + 4\delta(k+1) + 6\delta(k) + 4\delta(k-1) + \delta(k-2)$$

函数图像如下所示



选择 LMS 算法实现谱线增强, 如上图所示, 当延迟  $D > 3$  时, 就可以保证 LMS 滤波器参考信号中的宽带信号成分与输入信号中的宽带信号成分不相关, 而两者中的窄带信号成分仍然保持一定的相关性, 因此能够有效地起到谱线增强的作用。

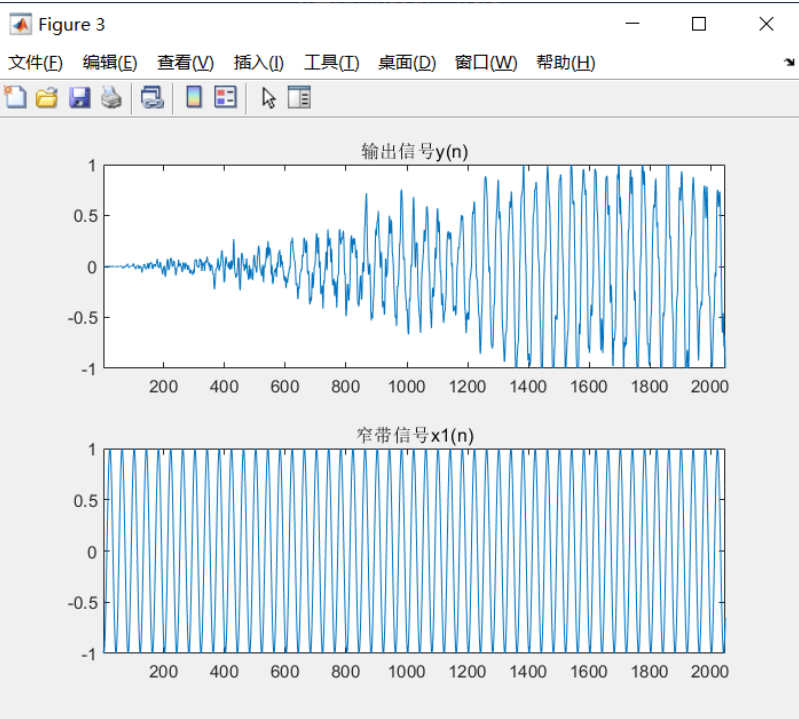
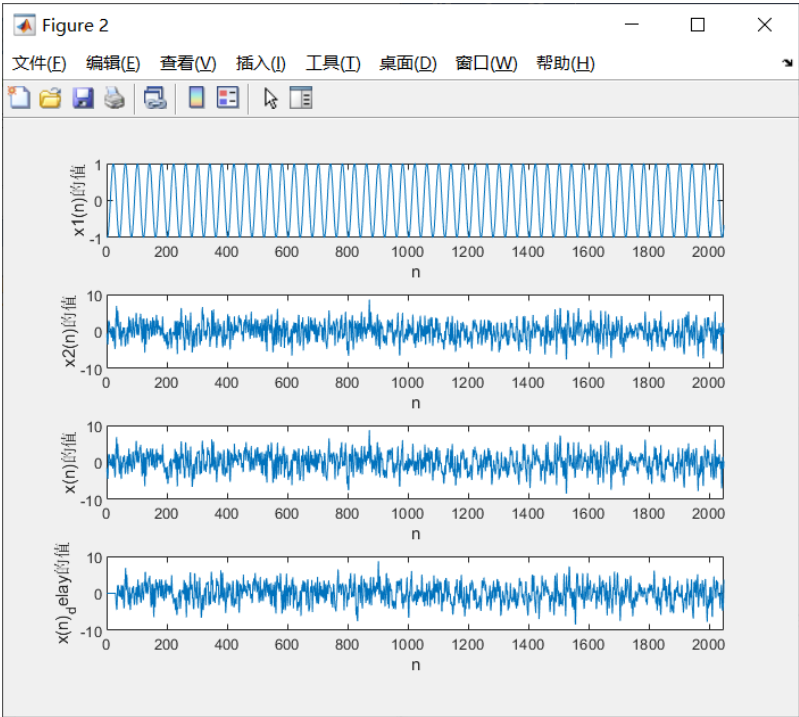
(2) 产生一个 $x(n)$ 序列。选择合适的 $\delta$ 值。让 $x(n)$ 通过谱线增强器。画出输出信号 $\tilde{y}(n)$ 和误差信号  $e(n)$ 的波形, 并分别与 $x_1(n)$ 和 $x_2(n)$ 比较。

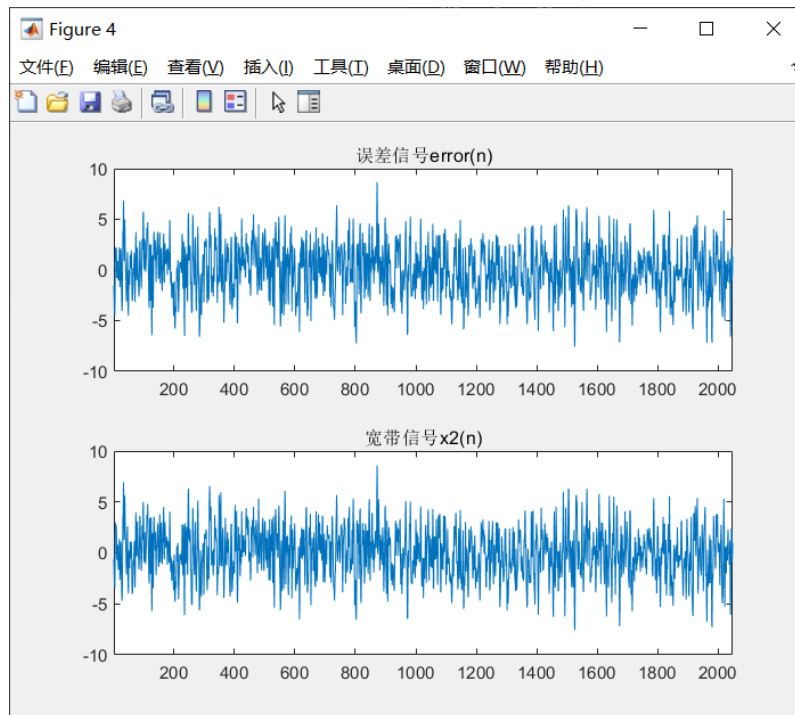
解: 考虑到噪声功率远大于信号功率, 所以为了实现谱线增强需要选择合适的滤

波器阶数和迭代步长。

当滤波器阶数较小时，误差信号  $e(n)$  可以收敛到  $x_2(n)$ ，但  $\hat{y}(n)$  不能很好地收敛到  $x_1(n)$ 。当迭代步长小时，精度大但收敛慢，当迭代步长大时，精度小但收敛快。

通过实验比较与理论分析，选择滤波器阶数  $R=250$ ，迭代步长  $\delta=0.00001$ ，延迟  $D=30$ ，采样点  $N=2048$ ，信号波形如下图所示。







## 代码附录

### 第一题

```
1 —   clc;
2 —   clear all;
3 —   close all;
4
5 —   N=512;%采样点数
6 —   epoch=100;%迭代次数
7 —   a1=0.1;a2=-0.8;%默认参数
8 —   step=0.05;%LMS算法步长
9 —   R=2;%滤波器阶数
10 —  h_all=zeros(2,N,epoch);%存放每次实验的a1,a2
11 —  en_all=zeros(1,N,epoch);%存放每次实验的误差
12 —  Jn_all=zeros(1,N,epoch);%100次实验的J(n)
13 —  figure;
14 —  for i=1:2
15 —      v=sqrt(0.27)*randn(1,N);%方差为0.27,均值为0的白噪声
16 —      x=zeros(1,N);
17 —      x(1)=v(1);
18 —      x(2)=-a1*x(1)+v(2);
19 —      for j=3:N
20 —          x(j)=-a1*x(j-1)-a2*x(j-2)+v(j);
21 —      end
22 —      plot(x);axis([0 N -3 3]); hold on;
23 —  end
24 —  legend('第1次生成x(n)过程','第2次生成x(n)过程');
25
26 —  for i=1:epoch
27 —      v=sqrt(0.27)*randn(1,N);
28 —      x=zeros(1,N);
29 —      h=[0;0];
30 —      x(1)=v(1);
31 —      x(2)=-a1*x(1)+v(2);
32 —      for j=3:N
33 —          x(j)=-a1*x(j-1)-a2*x(j-2)+v(j);
34 —      end
35 —      for j=3:N
36 —          e=x(j)-h'*x(j-1:-1:j-2)';
37 —          h=h+step*e*x(j-1:-1:j-2)';
38 —          h_all(1,j,i)=h(1,1);
39 —          h_all(2,j,i)=h(2,1);
40 —          en_all(:,j,i)=e;
41 —          Jn_all(1,j,i)=e^2;
42 —      end
43 —  end
44
```

```

45 — h_average=zeros(2,N);
46 — E=zeros(2,N);
47 — en_average=zeros(1,N);
48 — Jn_average=zeros(1,N);
49 — for i=1:N
50 —     h_average(1,i)=sum(h_all(1,i,:))/epoch;
51 —     E(1,i)=-a1-h_average(1,i);
52 —     h_average(2,i)=sum(h_all(2,i,:))/epoch;
53 —     E(2,i)=-a2-h_average(2,i);
54 —     en_average(1,i)=sum(en_all(1,i,:))/epoch;
55 —     Jn_average(1,i)=sum(Jn_all(1,i,:))/epoch;
56 — end
57
58 — en_fft=fft(en_average(1,:));
59 — E1_fft=fft(E(1,:));
60 — E2_fft=fft(E(2,:));
61
62 — figure;
63 — plot(1:N,h_average(1,:),1:N,h_average(2,:));axis([0 N-1 -0.2 1]);
64 — title('迭代100次取平均');legend('a1平均值','a2平均值');
65 — figure;
66 — plot(1:N,abs(en_fft),1:N,abs(E1_fft),1:N,abs(E2_fft));
67 — title('功率谱');axis([0 N-1 0 10]);
68 — legend('预测误差的功率谱','a1误差的功率谱','a2误差的功率谱');
69 — figure;
70 — plot(1:N,Jn_average(1,:));
71 — title('LMS实验平均误差学习曲线-预测误差平方');axis([0 N-1 -0.2 1]);

```

## 第二题

```
1 —   clc;
2 —   clear all;
3 —   close all;
4
5 —   N=2048;%采样点数
6 —   R=250;%滤波器阶数
7 —   step=0.00001;%迭代步长
8 —   delay=30;%延迟D
9 —   h=zeros(R,N);
10 —  n=1:N;
11 —  x1n=sin(0.05*pi*n+2*pi*rand(1));
12 —  en=randn(1,N);
13 —  x2n=zeros(1,N);
14 —  x2n(1)=en(1);
15 —  x2n(2)=en(2)+2*en(1);
16 —  for i=3:N
17 —      x2n(i)=en(i)+2*en(i-1)+en(i-2);
18 —  end
19
20 —  x=x1n+x2n;
21 —  x_delay=zeros(1,N);
22 —  for i=delay+1:N
23 —      x_delay(i)=x(i-delay);
24 —  end
25
26 —  y=zeros(1,N);
27 —  e=zeros(1,N);
28 —  for i=1:R-1
29 —      y(i)=h(1:i)*x_delay(i:-1:1)';
30 —      e(i)=x(i)-y(i);
31 —      h(1:i)=h(1:i)+step*e(i)*x_delay(i:-1:1);
32 —  end
33
34 —  for i=R:N
35 —      y(i)=h(:,i-1)'*x_delay(i:-1:i-R+1)';
36 —      e(i)=x(i)-y(i);
37 —      h(:,i)=h(:,i-1)+step*e(i)*x_delay(i:-1:i-R+1)';
38 —  end
39
```

```

40 — figure;
41 — m=0:1:10;
42 — rx1=0.5*cos(0.05*pi*m);
43 — rx2=1*(m== -2)+4*(m== -1)+6*(m==0)+1*(m==2)+4*(m==1);
44 — stem(rx1);hold on;stem(rx2);
45 — title('自相关函数');legend('Rx1','Rx2');
46 — figure;
47 — subplot(4,1,1);plot(x1n);xlabel('n');ylabel('x1(n)的值');axis([0 N -1 1]);
48 — subplot(4,1,2);plot(x2n);xlabel('n');ylabel('x2(n)的值');axis([0 N -10 10]);
49 — subplot(4,1,3);plot(x);xlabel('n');ylabel('x(n)的值');axis([0 N -10 10]);
50 — subplot(4,1,4);plot(x_delay);xlabel('n');ylabel('x(n)_delay的值');axis([0 N -10 10]);
51 — figure;
52 — subplot(2,1,1);plot(y);title('输出信号y(n)');axis([1 N -1 1]);
53 — subplot(2,1,2);plot(x1n);title('窄带信号x1(n)');axis([1 N -1 1]);
54 — figure;
55 — subplot(2,1,1);plot(e);title('误差信号error(n)');axis([1 N -10 10]);
56 — subplot(2,1,2);plot(x2n);title('宽带信号x2(n)');axis([1 N -10 10]);

```