目录

[作业一：IIR 模拟滤波器 巴特沃夫滤波器 1](#_Toc26889342)

[作业二：IIR 模拟滤波器总结 数字滤波器 5](#_Toc26889343)

[作业三：FIR数字滤波器设计 10](#_Toc26889344)

[作业四：功率谱估计方法 12](#_Toc26889345)

[作业五：现代功率谱估计方法 MUSIC算法 19](#_Toc26889346)

[作业六：维纳滤波原理及LMS自适应算法 21](#_Toc26889347)

[作业七：递归最小二乘（RLS） 29](#_Toc26889348)

[作业八：卡尔曼滤波 34](#_Toc26889349)

作业一：IIR 模拟滤波器

**1.按下表整理复习离散傅立叶变化相关内容**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **FT傅里叶积分变换** | **FS傅里叶级数** | **DTFT离散时间序列的傅里叶变换** | **DFS周期序列的傅立叶级数** | **DFT离散傅里叶变换** |
| **时域** | **连续，非周期** | **连续，周期** | **离散，非周期** | **离散，周期** | **离散，非周期** |
|  |  |  |  |  |
| **频域** |  |  |  |  |  |
| **连续，非周期** | **离散，非周期** | **连续，周期(2π)** | **离散，周期** | **离散，非周期** |
|  | **拉普拉斯变换**  **LT**  **S平面** |  | **Z变换ZT**  **Z平面** |  |  |

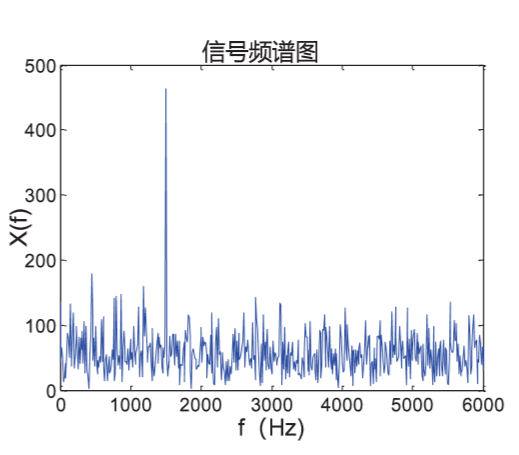
**2假设一音频信号，采样频率为。它收到一窄带（如非常接近于弦波的信号）信号的干扰。其中。**

**注意：**

**1.信号频谱范围在，所以采样后不会出现混叠。**

**2.干扰频率。**

**设计一个简单的数字滤波器来滤除干扰，而不严重影响信号本身。**



程序代码：

clear

clc

fs = 12000; %采样频率

N = 12000; %采样点数

n = 0:N-1;

t = n / fs; %时间序列

s = 10000 \* rand(size(t)); %原始音频信号

w = 10000 \* rand(1) \* sin(2 \* pi \* 1500 \* t); %噪声信号

x = s + w; %混合信号

f = fs \* (0:(N / 2)) / N; %0到6khz频率序列

%混合信号的频谱分析

y1 = fft(x);

P2 = abs(y1/N);

mag1 = P2(1:N/2+1);

mag1(2:end-1) = 2 \* mag1(2:end-1);

%陷波器设计

r=0.9;

w0=2\*pi\*1500/fs; %要滤掉的频率

b=[1 -2\*cos(w0) 1];

a=[1 -2\*r\*cos(w0) r\*r];

[H,w]=freqz(b,a,N);

figure(1)

subplot(221);plot(w,abs(H));title('陷波器的幅频响应');

subplot(222);plot(w,angle(H));title('陷波器的相频响应');

subplot(223);zplane(b,a);title('陷波器的零极点图');

%陷波器滤波

y2=filter(b,a,x);

Y=fft(y2);

P3=abs(Y/N);

mag2=P3(1:N/2+1);

mag2(2:end-1) = 2 \* mag2(2:end-1);

figure(2);

subplot(2,1,1)

plot(f,mag1)

ylim([0 500]);xlabel('频率/Hz');ylabel('振幅')

title('原始信号频谱图')

subplot(2,1,2)

plot(f,mag2)

ylim([0 500]);xlabel('频率/Hz');ylabel('振幅')

title('陷波器滤波后的信号y(n)')

运行结果：



图1-1 陷波滤波器的幅频、相频响应与零极点图



图1-2 原始信号和滤波后的幅频图

作业二：IIR 模拟滤波器总结 数字滤波器设计：双线性不变法与脉冲响应不变法（间接法）

**1复习模拟滤波器的设计方法**

①巴特沃斯滤波器

 式中，n为滤波器阶次；为信号圆频；为3dB带宽处的截止频率。

 确定滤波器阶数，

也可直接利用公式确定滤波器阶数

根据阶数查表得到归一化的系统函数。

②切比雪夫滤波器



由通带波纹 dB可以得到波动系数

由阻带衰减，求得

根据，，，求出阶数

，即可求出的值

根据切比雪夫多项式，就可解出阶数

根据阶数查表得到传递函数

**2.用脉冲响应不变法设计数字低通滤波器，要求通带和阻带具有单调下降特性。具体指标参数：，，，.**

解：根据间接设计的基本步骤求解。

1.将数字滤波器设计指标转换为相应模拟滤波器指标。设采样周期为，由式（6.2.9）得到：

，，，

2.设计相应的模拟滤波器，得到模拟系统函数。根据单调下降要求，选择巴特沃思滤波器。首先用式（6.1.5）和式（6.1.6）求出波纹幅度参数为:

，

由式（6.1.8）和（6.1.9）得到：，

再将和代入式（6.1.14）和式（6.1.16）计算得到：

，取整数

取时， 

查表6.1.1得到归一化2阶巴特沃斯多项式为：



将其代入式(6.1.21)得到归一化系统函数为:



其中，

，，

将代入式（6.1.22）去归一化，得到希望设计的低通滤波器的系统函数为：，其中，

3.将代入式（6.2.6），将模拟滤波器系统函数转换成数字滤波器系统函数，即：



**3.用双线性变换法设计上述数字低通滤波器。**

解：根据用双线性变换法设计IIR数字滤波器的设计步骤求解。

1.确定数字滤波器指标。

，，，

2.非线性预畸变校正，将数字滤波器设计指标转换为相应的过渡模拟滤波器指标。

设采样周期，由式（6.2.23）得到：

，

，

3.设计相应的过渡模拟滤波器。

根据单调下降要求，选择巴特沃思滤波器。并仿照例6.2.2计算出,

。查表6.1.1得到归一化1阶巴特沃思模拟滤波器的系统函数为：。去归一化得到：



（4）用双线性变换法将模拟滤波器转换为数字滤波器，即:



程序代码：

clear

clc

T = 0.1; %周期0.1s

fs = 1 / T;

%巴特沃斯滤波器的设计指标参数

Wp = 0.1\* pi / (fs/2);

Ws = 0.3 \* pi / (fs/2);

Rp = 1;

Rs = 12;

[n,Wc] = buttord(Wp,Ws,Rp,Rs); %计算滤波器的阶数n和3dB截止频率

%脉冲响应不变法

[B,A] = butter(n,Wc,'s'); %计算相应的模拟滤波器的系统函数

[Bz,Az] = impinvar(B,A); %用脉冲响应不变法将模拟滤波器转换为数字滤波器H(s)>>H(z)

figure(1)

freqz(Bz,Az,1024,fs)

title('脉冲响应不变法')

%双线法

[A,B] = butter(n,Wc);

figure(2)

freqz(A,B,1024,fs)

title('双线法')

运行结果：



图2-1 滤波器的幅频相频特性（脉冲响应不变法）



图2-2 滤波器的幅频特性（双线法）

**4.对上述两种方法所求结果进行分析、比较**

答：设计两种滤波器结果如下图所示，从图可知：**脉冲响应数字滤波器容易出现频谱混叠失真**，并且其不能设计数字**高通或者带阻滤波器**。而双线性变换法没有频谱混叠现象，并可以设计数字高通滤波器以及带阻滤波器。在实际应用之中，**一般选用双线性变换法**设计数字滤波器。

作业三：FIR数字滤波器设计：窗函数法（直接法）

**1对模拟信号进行低通滤波处理，要求通带，阻带上衰减大于，采样频率。用窗函数法设计满足要求的数字低通滤波器，求出，并画出损耗函数曲线和相频特性曲线。**

解：

1.确定相应的数字滤波器指标：

通带截止频率为： 

阻带截止频率为： 

阻带最小衰减为： 

2.用窗函数法设计FIR数字低通滤波器，选择布莱克曼窗。根据式（7.2.17）计算布莱克曼窗的控制参数为：



程序代码：

clear

clc

wp = 0.4 \* pi;

ws = 0.096 \* pi;

rs = 65;

DB =abs( ws – wp); %计算过渡带宽度

M = ceil(12\*pi/ DB); %计算布莱克曼窗所需阶数 向上取整

N=M+mod(M+1,2); %取余数，确保h（n）长度N为奇数

wc = (wp + ws)/ 2 / pi; %计算理想低通滤波器通带截止频率

hn = fir1(N,wc,’low’,blackman(N)); %调用fir1计算低通FIRDF的h(n)

%波形图

figure(1)

stem(hn);

xlabel('n');

ylabel('h(n)');

title('h(n)波形');

%损耗曲线与相频曲线

figure(2)

freqz(hn,1,512)

运行结果：



图3-1 波形图



图3-2损耗曲线与相频特性曲线

作业四：功率谱估计方法FFT BT法（经典法）周期图法（直接法）AR模型的Levison-Drbin算法

**3.17在计算机上用如下方法产生随机信号的观测样本：首先产生一段零均值、方差为的复高斯白噪声序列；然后在上叠加三个复正弦信号，它们的归一化频率分别是、、和。调整和正弦信号的幅度，使在、和处的信噪比分别为、和。**

**1.令观测样本长度，试用3.1.1节讨论的基于的自相关函数快速计算方法估计自相关函数，并与教材式（3.1.2）估计出的自相关函数作比较。**

程序代码：

clear

clc

%产生零均值、方差为1的复高斯白噪声序列

N = 32;

noise = (randn(1,N) + 1i \* randn(1,N) ) \* sqrt(2);

%产生三个复正弦信号

f1 = 0.15;

f2 = 0.17;

f3 = 0.26; %信号的归一化频率

SNR1 = 30;

SNR2 = 30;

SNR3 = 27; %信号的信噪比

A1 = 10^(SNR1 / 20);

A2 = 10^(SNR2 / 20);

A3 = 10^(SNR3 / 20); %信号的幅度

signal1 = A1 \* exp(1i \* 2 \* pi \* f1 \* (0:N-1));

signal2 = A2 \* exp(1i \* 2 \* pi \* f2 \* (0:N-1));

signal3 = A3 \* exp(1i \* 2 \* pi \* f3 \* (0:N-1)); %产生复正弦信号

%产生观察样本u(n)

un = signal1 + signal2 + signal3 + noise;

%基于FFT的自相关函数快速计算方法

Uk = fft(un,2\*N); %对un进行2N点的FFT

Sk = (1/N)\*abs(Uk).^2; %计算功率谱估计Sk

r0 = ifft(Sk); %对功率谱估计Sk求IFFT

r1 = [r0(N+2:2\*N),r0(1:N)]; %根据教材（3.1.8）求得自相关函数

figure(1)

stem(real(r1)); %提取实部

xlabel('m');

ylabel('实部');

figure(2)

stem(imag(r1)); %提取虚部

xlabel('m');

ylabel('虚部');

%教材式（3.1.2）估计自相关函数

r = xcorr(un,N-1,'biased'); %自相关函数 矢量长度un，N-1，长度不一则补零

figure(3)

stem(real(r)); %提取实部

xlabel('m');

ylabel('实部');

figure(4)

stem(imag(r)); %提取虚部

xlabel('m');

ylabel('虚部');

运行结果：



图4-1 FFT实部提取



图4-2 FFT虚部提取



图4-3 自相关函数实部提取



图4-4 自相关函数虚部提取

比较发现，基于FFT的自相关函数快速计算方法与教材式（3.1.2）估计自相关函数提取的实部和虚部无明显区别，几乎一致。

**2.令观测样本长度，试分别用法和周期图法估计的功率谱，这里设法中所用自相关函数的单边长度。**

程序代码：

clear

clc

%产生零均值、方差为1的复高斯白噪声序列

N = 256;

noise = (randn(1,N) + 1i \* randn(1,N)) \* sqrt(2);

%产生三个复正弦信号

f1 = 0.15;

f2 = 0.17;

f3 = 0.26; %信号的归一化频率

SNR1 = 30;

SNR2 = 30;

SNR3 = 27; %信号的信噪比

A1 = 10^(SNR1 / 20);

A2 = 10^(SNR2 / 20);

A3 = 10^(SNR3 / 20); %信号的幅度

signal1 = A1 \* exp(1i \* 2 \* pi \* f1 \* (0:N-1));

signal2 = A2 \* exp(1i \* 2 \* pi \* f2 \* (0:N-1));

signal3 = A3 \* exp(1i \* 2 \* pi \* f3 \* (0:N-1)); %产生复正弦信号

%产生观察样本u(n)

un = signal1 + signal2 + signal3 + noise;

%周期图法

NF = 1024; %周期图法中FFT的点数

Spr = fftshift((1/NF) \* abs(fft(un,NF)).^2); %fftshift将零频率的分量移到频谱的中心

A = 10 \* log10(Spr);

f = (-length(A)/2 + 1):(length(A)/2);

figure(1)

plot(f/NF,A);

xlabel('w/2pi');

ylabel('归一化功率谱/dB');

title('周期图法');

%BT法

M = 64; %自相关函数的单边长度

r = xcorr(un,M,'biased'); %计算自相关函数

NF = 1024; %BT法中FFT的点数

BT = fftshift(fft(r,NF)); %BT法计算功率谱

B = 10 \* log10(BT);

f = (-length(B)/2 + 1):(length(B)/2);

figure(2)

plot(f/NF,B);

xlabel('w/2\pi');

ylabel('归一化功率谱/dB');

title('BT法');

运行结果：



图4-5 周期图法功率谱



图4-6 BT法功率谱

比较发现，周期图法与BT法估计功率谱很相似，但是BT法左部功率谱更平滑。

**3.令观测样本长度，试用迭代算法求解模型的系数并估计的功率谱，模型阶数取为。**

程序代码：

clear

clc

%产生零均值、方差为1的复高斯白噪声序列

N = 256;

noise = (randn(1,N) + 1i \* randn(1,N) / sqrt(2));

%产生三个复正弦信号

f1 = 0.15;

f2 = 0.17;

f3 = 0.26; %信号的归一化频率

SNR1 = 30;

SNR2 = 30;

SNR3 = 27; %信号的信噪比

A1 = 10^(SNR1 / 20);

A2 = 10^(SNR2 / 20);

A3 = 10^(SNR3 / 20); %信号的幅度

signal1 = A1 \* exp(1i \* 2 \* pi \* f1 \* (0:N-1));

signal2 = A2 \* exp(1i \* 2 \* pi \* f2 \* (0:N-1));

signal3 = A3 \* exp(1i \* 2 \* pi \* f3 \* (0:N-1)); %产生复正弦信号

%产生观察样本u(n)

un = signal1 + signal2 + signal3 + noise;

%计算自相关函数值

p = 16; %AR模型的阶数

r0 = xcorr(un,p,'biased'); %直接计算自相关函数

r = r0(p + 1:2 \* p + 1); %提取r(0),r(1),...,r(p)

%计算一阶AR模型的系数与输入方差 迭代算法初值准备

a(1,1) = -r(2)/r(1); %1阶AR模型的系数

signal(1) = r(1) - (abs(r(2)^2)/r(1)); %1阶AR模型的输入方差

%Levinsion-Durbin迭代算法

for m = 2:p

k(m) = -(r(m+1) + sum(a(m-1,1:m-1) .\* r(m:-1:2)))/signal(m-1);

a(m,m) = k(m);

for i = 1:m-1

a(m,i) = a(m-1,i) + k(m) \* conj(a(m-1,m-i));

end

signal(m) = signal(m-1) \* (1-abs(k(m))^2);

end

%计算十六阶AR模型的功率谱

NF = 1024; %AR方法中FFT的点数

Par = signal(p)./fftshift(abs(fft([1,a(p,:)],NF)).^2); %p阶AR模型的功率谱

C = 10\*log10(Par);

f = (-length(C)/2 + 1):(length(C)/2);

figure(1)

plot(f/NF,C);

xlabel('w/(2\*pi)');

ylabel('归一化功率谱/dB');

title('16阶AR模型');

运行结果：



图4-7 16阶AR模型

作业五：现代功率谱估计方法MUSIC算法

**3.19复正弦加白噪声随机过程同题3.18中所给。试用方法进行信号频率估计的仿真实验。（要求：信号样本数取1000，估计的自相关矩阵为8阶。）画出相应的频率估计谱线。**

解：根据AIC准则进行单次信号源估计的结果为2。采用AIC准则估计的信号源个数进行MUSIC谱估计。

MUSIC算法计算步骤：

1.根据样本值，估计自相关矩阵；

2.对进行特征分解，得到个最小特征值对应的归一化特征向量，即得到噪声子空间的一组基向量，并构造矩阵；

3.在内改变，计算，其峰值位置就是信号频率的估计值。

程序代码：

clear

clc

%产生零均值、方差为1的复高斯白噪声序列v(n)

N = 1000;

noise = (randn(1,N) + 1i \* randn(1,N)) \*sqrt(2);

%产生带噪声的信号样本u(n)

signal1 = exp(1i \* 0.5 \* pi \* (0:N-1) + 1i \* 2 \* pi \* rand(1)); %产生第一个信号

signal2 = exp(1i \* -0.3 \* pi \* (0:N-1) + 1i \* 2 \* pi \* rand(1)); %产生第二个信号

un = signal1 + signal2 + noise; %产生带噪声的信号

%计算自相关矩阵

M = 8; %自相关矩阵阶数

for k = 1:N-M

xs(:,k) = un(k + M - 1: -1 : k); %构造样本矩阵

end

R = xs \* xs' / (N-M); %计算自相关矩阵

%自相关矩阵特征值分解

[U,E] = svd(R);

ev = diag(E); %提取对角元素上的特征值

%根据AIC准则进行信号源个数估计

for k = 1:M

dec = prod(ev(k:M) .^ (1/(M - k + 1)));

nec = mean(ev(k:M));

lnv = (dec/nec)^((M - k + 1) \* N);

AIC(k) = -2 \* log(lnv) + 2 \* (k-1) \* (2 \* M - k + 1);

end

[Amin,K] = min(AIC); %寻找使AIC准则最小的索引

%计算MUSIC谱

En = U(:,K:M); %噪声子空间的向量组成的矩阵

NF = 2048; %MUSIC的扫描点数

for n = 1:NF

Aq = exp(-1i \* 2 \* pi \* (n-1) / NF \* (0:M-1)');

Pmusic(n) = 1/(Aq' \* En \* En' \* Aq); %MUSIC谱

end

S = 10 \* log10(Pmusic);

m1 = -1023:1024;

plot(m1/2048,S);

xlabel('w/2pi');

ylabel('归一化功率谱/dB');

title('MUSIC谱估计');

运行结果：



图5-1 MUSIC谱估计

作业六：维纳滤波原理及LMS自适应算法

**1.考虑过程****，其差分方程，其中是零均值、方差为的加性白噪声。参数****，。**

**1.产生点的样本序列。**

解：1.产生点的()样本序列

程序代码：

clear

clc

%产生样本序列

a1 = -0.975;

a2 = 0.95;

s = 0.0731; %方差

trials = 100; %随机试验次数

data\_length = 512;

n = 1:data\_length;

v = sqrt(s) \* randn(data\_length,1);

u0 = [0 0 0];

num = 1; %分子系数

den = [1 a1 a2]; %分母系数

Zi = filtic(num,den,u0);

u = filter(num,den,v,Zi);

figure(1)

plot(u);

xlabel('n'); ylabel('(n)--y(n)');

grid;

**2.令为二阶线性预测器的输入，在****、的情况下用滤波器来估计****和。**

用LMS算法，迭代估计和；

①初始化，

权向量：

估计误差：

输入向量：

②对

权向量的更新：

期望信号估计：

估计误差：

③令，转到步骤2

%LMS迭代算法

h1 = 0.05; %步长因子h1 h2

h2 = 0.005;

w1 = zeros(2,data\_length); %在不同步长因子下的w1 w2权向量初始值、存储空间

w2 = zeros(2,data\_length);

e1 = zeros(data\_length,1); %估计误差的初始值、存储空间大小

e2 = zeros(data\_length,1);

d1 = zeros(data\_length,1);

d2 = zeros(data\_length,1); %期望信号的估计同上

for n = 3:data\_length-1 %LMS迭代

w1(:,n+1) = w1(:,n) + h1 \* u(n-1:-1:n-2) \* conj(e1(n));

w2(:,n+1) = w2(:,n) + h2 \* u(n-1:-1:n-2) \* conj(e2(n));

d1(n+1) = w1(:,n+1)' \* u(n:-1:n-1);

d2(n+1) = w2(:,n+1)' \* u(n:-1:n-1);

e1(n+1) = u(n+1) - d1(n+1);

e2(n+1) = u(n+1) - d2(n+1);

end

figure(2)

plot(1:512,w1(1,:),'r');

hold on;

plot(1:512,w1(2,:),'b');

hold off;

xlabel('迭代次数');

ylabel('抽头权值');

title('步长0.05 ');

figure(3)

plot(1:512,w2(1,:),'r');

hold on;

plot(1:512,w2(2,:),'b');

hold off;

xlabel('迭代次数');

ylabel('抽头权值');

title('步长0.005');

运行结果：



图6-1步长为0.05时的w1、w2



图6-2 步长为0.005时的w1、w2

**3.在2的参数条件下，滤波器100次独立实验，通过平均预测误差的平均值，计算剩余均方误差和失调参数，并画出学习曲线。**

通过100次独立实验计算剩余均方误差和失调参数

① 剩余均方误差为：

当时，；

当时，；

② 失调参数为：

当时，失调参数为0.0504；

当时，失调参数为0.0044；

程序代码：

%进行100次独立实验

trials = 100;

data\_length = 512;

wopt = zeros(2,trials);

Jmin = zeros(1,trials);

sum\_eig = zeros(trials,1);

w11 = zeros(trials,data\_length);

w12 = zeros(trials,data\_length);

w21 = zeros(trials,data\_length);

w22 = zeros(trials,data\_length);

e1 = zeros(trials,1);

e2 = zeros(trials,1);

%步长

h1 = 0.05;

h2 = 0.005;

%计算最小均方误差

for m = 1:trials

%产生样本序列

a1 = -0.975;

a2 = 0.95;

s = 0.0731;

n = 1:data\_length;

v = sqrt(s)\*randn(data\_length,1);

u0 = [0 0 0];

num = 1;

den = [1 a1 a2];

Zi = filtic(num,den,u0);

u = filter(num,den,v,Zi);

%LMS迭代算法

w1 = zeros(2,data\_length);

w2 = zeros(2,data\_length);

e1 = zeros(data\_length,1);

e2 = zeros(data\_length,1);

d1 = zeros(data\_length,1);

d2 = zeros(data\_length,1);

for n = 3:data\_length-1

w1(:,n+1) = w1(:,n) + h1 \* u(n-1:-1:n-2) \* conj(e1(n));

w11(m,n+1) = w1(1,n+1);

w12(m,n+1) = w1(2,n+1);

w2(:,n+1) = w2(:,n) + h2 \* u(n-1:-1:n-2) \* conj(e2(n));

w21(m,n+1) = w2(1,n+1);

w22(m,n+1) = w2(2,n+1);

d1(n+1) = w1(:,n+1)' \* u(n:-1:n-1);

d2(n+1) = w2(:,n+1)' \* u(n:-1:n-1);

e1(n+1) = u(n+1) - d1(n+1);

e2(n+1) = u(n+1) - d2(n+1);

end

e1(m) = mean(e1);

e2(m) = mean(e2);

rm = xcorr(u,'biased');

R = [rm(512),rm(513);rm(511),rm(512)];

rm512(m) = rm(512);

rm513(m) = rm(513);

rm511(m) = rm(511);

p = [rm(511);rm(510)];

wopt(:,m) = R \ p;

[v,d] = eig(R);

Jmin(m) = rm(512) - p' \* wopt(:,m);

sum\_eig(m) = d(1,1) + d(2,2);

end

%100次平均误差

sJmin = sum(Jmin) / trials;

%100次平均特征值之和

sum\_eig\_100trials = sum(sum\_eig)/100;

Jexfin1 = h1 \* sJmin \* (sum\_eig\_100trials / (2 - h1 \* sum\_eig\_100trials));

Jexfin2 = h2 \* sJmin \* (sum\_eig\_100trials /(2 - h2 \* sum\_eig\_100trials));

%计算失调参数

M1 = Jexfin1 / sJmin;

M2 = Jexfin2 / sJmin;

%计算100次的系数平均

q1 = mean(w11(:,:));

q2 = mean(w12(:,:));

q3 = mean(w21(:,:));

q4 = mean(w22(:,:));

rm11 = mean(rm511);

rm22 = mean(rm512);

rm33 = mean(rm513);

figure(3)

subplot(2,1,1)

plot(1:512,q1,'r');

hold on;

plot(1:512,q2,'b');

hold off;

xlabel('迭代次数');

ylabel('抽头权值');

title('100次步长0.05');

subplot(2,1,2)

plot(1:512,q3,'r');

hold on;

plot(1:512,q4,'b');

hold off;

xlabel('迭代次数');

ylabel('抽头权值');

title('100次 步长0.005');

w1=mean(wopt(1,:));

w2=mean(wopt(2,:));

%计算均方误差

for t=1:data\_length

J1(t)=sJmin+([q1(t) q2(t)]-[w1 w2])\*[rm22,rm33;rm11,rm22]\*([q1(t) q2(t)]-[w1 w2])';

J2(t)=sJmin+([q3(t) q4(t)]-[w1 w2])\*[rm22,rm33;rm11,rm22]\*([q1(t) q2(t)]-[w1 w2])';

end

e1\_100trials\_ave=J1;

e2\_100trials\_ave=J2;

%计算剩余均方误差

Jex1=e1\_100trials\_ave-sJmin;

Jex2=e2\_100trials\_ave-sJmin;

运行结果：



图6-3 迭代100次不同步长的抽头权值

**4.改变，其它参数不变，计算剩余均方误差和失调参数，并画出学习曲线，比较和二者学习曲线的区别。**

程序代码：

%学习曲线

figure(4)

plot(1:512,J1,'r');

hold on;

plot(1:512,J2,'b');

hold off;

xlabel('迭代次数');

ylabel('均方误差');

title('100次 步长 0.05红色 0.005蓝色');

grid on;

运行结果：



图6-5 不同步长的学习曲线

结论：当步长因子越小时，LMS稳态性能越好，但收敛速度慢；当步长因子越大时，LMS稳态性能越差，但收敛速度变快。

作业七：递归最小二乘（RLS）

**6.15、考虑一阶模型：的线性预测。假设白噪声的方差为。使用抽头数为的滤波器，用和算法实现的线性预测。并对两种方法进行分析、比较。**

解：应用RLS算法迭代求解最优权向量：

①初始化：，是小的正数，，遗忘因子；

②当时，完成以下迭代运算：

增益向量 

先验估计误差 





③令，转第2步。

程序代码：

clear

clc

%产生AR模型的输出信号

a1=0.99;

seta=0.995;

N=1024;

for i=1:600

y=randn(1,N)\*sqrt(seta); % noise sig

num=1; %分子系数

den=[1 a1]; %分母系数

u0=zeros(length(den),1)'; %初始数据

xic=filtic(num,den,u0); %初始条件

un=filter(num,den,y,xic); %产生数据

%产生期望响应信号和观测数据矩阵

n0=1; %需要实现n0步线性预测

m=2; %滤波器阶数 抽头数

b=un(n0+1:N); %预测期望响应d(n)

l=length(b);

un1 = [zeros(m-1,1), un]; %扩展数据

a=zeros(m,l); %u(n)

for k=1:l-1

a(:,k)=un1(m-1+k:-1:k); %观测数据矩阵A矩阵

end

%应用RLS算法迭代求最有权向量

delta=0.004; %调整参数

lambda=0.98; %遗忘因子

w=zeros(m,l+1); %存储权向量

epsilon=zeros(l,1); %先验估计误差存储

p1=eye(m)/delta; %相关矩阵的逆

for k=1:l

pin=p1\*a(:,k);

denok=1+a(:,k)'\*pin;

kn=pin/denok;

epsilon(k)=b(k)-w(:,k)'\*a(:,k);

w(:,k+1) = w(:,k) + kn \* conj(epsilon(k));

p1=p1/lambda-kn\*a(:,k)'\*p1/lambda;

end

mse=abs(epsilon).^2;

JFWC(i,:) =mse;

W1(i,:) = w(1,:);

W2(i,:) = w(2,:);

end

%求800次均值

mse = mean(JFWC);

w1 = mean(W1); %两个抽头的权值

w2 = mean(W2);

figure(1)

plot(1:800,mse(1:800),'r');

xlabel('迭代次数');

ylabel('MSE');

title('均方误差');

grid on;

%权值

figure(2)

plot(1:800,w1(1:800),'r');

hold on;

plot(1:800,w2(1:800),'b');

hold off;

xlabel('迭代次数');

ylabel('权值');

title('计算权值');

grid on;

%产生样本序列LMS法

clear

clc

%产生AR模型的输出信号

a1=0.99;

seta=0.995;

data\_length=2000;

h1 = 0.05;

for i=1:1000

y=randn(1,data\_length)\*sqrt(seta); % noise sig

num=1; %分子系数

den=[1 a1]; %分母系数

u0=zeros(length(den),1)'; %初始数据

xic=filtic(num,den,u0); %初始条件

u=filter(num,den,y,xic); %产生数据



图7-1 均方误差



图7-2 计算权值

%%运行节

%100次试验

W1 = zeros(100,4500); %在不同步长因子下的w1 权向量初始值、存储空间

W2 = zeros(100,4500); %估计误差的初始值、存储空间大小

for m = 1:100

%产生样本序列

L = 5000;

a1 = 0.99;

s = 0.995;

n = 1:L;

v = sqrt(s) \* randn(L,1);

u(1) = v(1);

for k = 2:L

u(k) = -a1 \* u(k-1) + v(k);

end

u=u(500:end);

%LMS迭代算法

M = 2;

w(1,:) = zeros(1,M);

e(1) = u(1);

mu = 0.001;

uu = zeros(1,M);

w(2,:) = w(1,:) + mu \* e(1) \* uu;

uu = [u(1),uu(1:M-1)];

dd = (w(2,:)\*uu')';

e(2) = u(3) - dd;

for k = 3:4501

w(k,:) = w(k-1,:) + mu \* e(k-1) \* uu;

uu = [u(k-1),uu(1:M-1)];

dd = (w(k,:)\*uu')';

e(k) = u(k)-dd;

end

W1(m,:) = w(1:4500,1)';

W2(m,:) = w(1:4500,2)';

end

W11 = mean(W1);

W22 = mean(W2);

plot(1:4500,W11,'r');

hold on;

plot(1:4500,W22,'b');

xlabel('迭代次数');

ylabel('抽头权值');

title('步长0.05');

grid on;

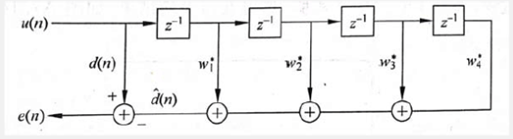


图7-3 两种方法比较图

比较可得，LMS计算权值时收敛相较RLS算法收敛非常慢，实际应用中更多的运用到RLS算法。

作业八：卡尔曼滤波

**设信号u(n)由一个四阶AR模型u(n)-1.6u(n-1)+1.46u(n-2)-0.616u(n-3)+0.1525u(n-4)=v(n)产生，其中v(n)为零均值的高斯白噪声过程，噪声方差为。如图是以该序列作为输入的四阶线性预测模型，试用卡尔曼滤波算法估计该模型中的最优权值，并与原模型参数进行比较。**



程序代码：

clear

clc

%产生给定方差的高斯白噪声

N=3000;

seta=0.0332; %噪声方差

y=randn(1,N)\*sqrt(seta); % noise sig

%AR模型u1(n-1)的参数

a1=-1.6;

a2=1.46;

a3=-0.616;

a4=0.1525;

%根据AR模型产生u（n）序列

N=3000;

u1=zeros(1,N); %初始化u1（n）n时刻输入信号的向量

for i=1:N-4

u1(i+4)=-a1\*u1(i+3)-a2\*u1(i+2)-a3\*u1(i+1)-a4\*u1(i)+y(i+4); %产生序列u（n）---z(n)观测向量

end

%卡尔曼滤波

e=0.005; %设定观测噪声方差

N=2000;

for i=5:N

U(:,i)=[u1(i-1); u1(i-2) ;u1(i-3); u1(i-4)]; %构成观测矩阵c(n)'前期准备

end

P=eye(4); %状态误差自相关矩阵 初始E矩阵

W=zeros(4,N); %最优权值 状态向量x(n)初始化1step

for i=1:2000

P\_pre=P; %预测状态误差自相关矩阵3step

A=(U(:,i))'\*P\_pre\*U(:,i)+e; %新息过程自相关矩阵4step

K=P\_pre\*U(:,i)/A; %卡尔曼增益5step

alpha(i)=u1(i)-(U(:,i))'\*W(:,i); %由观测信号计算新息过程2step

W(:,i+1)=W(:,i)+K\*alpha(i); %状态估计6step

P=P\_pre-K\*(U(:,i))'\*P\_pre; %状态估计误差自相关矩阵7step

end

t = 1:1000;

plot(t,W(1,t),'b',t,W(2,t),'r',t,W(3,t),'y',t,W(4,t),'g');

%红色线最优化估算结果滤波后的值，绿色线观测值，蓝色线预测值

b=W(1,600)

r=W(2,600)

y=W(3,600)

g=W(4,600)

%原参数 b=1.6;

% r=-1.46;

% y=0.616;

% g=-0.1525;

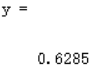
运行结果：



图8-1 卡尔曼滤波权值计算

**与原模型参数对比**：

原参数： b=1.6 卡尔曼滤波计算：

 r=-1.46

y=0.616

g=-0.1525

可以看到卡尔曼滤波算出的权值与题中相近。