第五章 AR 谱分析与自适应谱线增强器

张飞 XS06032001

5-1. 假设观测数据由下式产生：

$$x(n)=6\sin(0.4\pi n)+\sin(0.43\pi n)+w(n),w(n)\sim WN(0,1),n=1,2,\cdots 128$$

试用一般的最小二乘法估计观测数据的 AR 模型(模型的阶次分别取 4 和 6)，并分别用 DFT 和 AR 谱估计方法估计正弦波的频率和统计结果(均值和方差)。

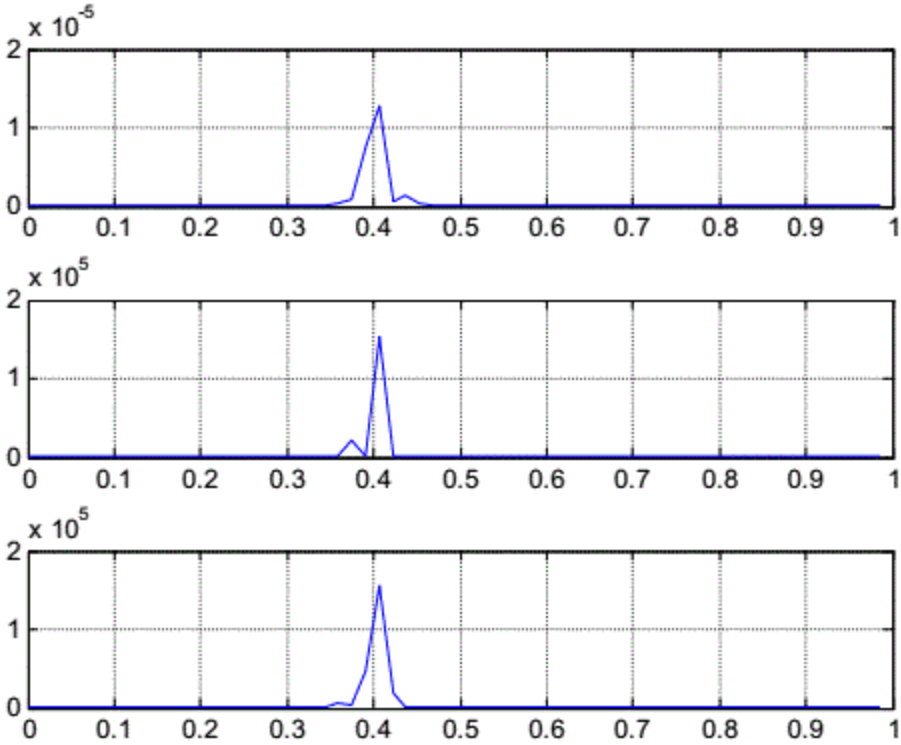
解：根据题意，编出仿真程序如下：

```
N=128;
for num=1:10
Signal=6*sin(0.4*pi*[1:N])+sin(0.43*pi*[1:N]);
Noise=1*randn(1,N);
x=Signal+Noise;
i=1;
z0=fft(x,N);
z0=abs(z0).^2
for k=1:N/2
        if z0(k)==max(z0)
             w11=k;
        end
   end

for k=1:N/2
         if (k > w11 & z0(k)>=z0(k-1)    & z0(k)>=z0(k+1) )
            w12(i)=k;i=i+1;
         end
end
n=1:N/2;z0=z/max(z0);
f11(num)=(w11-1)*2/N;
f12(num)=(w12(1)-1)*2/N;
mean11=mean(f11)
mean12=mean(f12)
var11=std(f11)^2
var12=std(f12)^2
subplot(3,1,1) ;plot((n-1)*2/N,z0(1:N/2));grid;

th0=ar(x',4,'ls');
u=1*randn(N,1);
y=idsim(u,th0);
z1=fft(y,N);
```

```
z1=abs(z1).^2;
i=1;
for k=1:N/2
        if z1(k)==max(z1)
            w21=k;
        end
   end

for k=1:N/2
         if (k > w21 & z1(k)>=z1(k-1)    & z1(k)>=z1(k+1) )
            w22(i)=k;i=i+1;
         end
end
n=1:N/2;
f21(num)=(w21-1)*2/N;    f22(num)=(w22(1)-1)*2/N;
mean21=mean(f21)
mean22=mean(f22)
var21=std(f21)^2
var22=std(f22)^2
subplot(3,1,2) ;plot((n-1)*2/N,z1(1:N/2));grid;

th1=ar(x',6,'ls');
u1=randn(N,1);
y1=idsim(u1,th1);
z2=fft(y1,N);
z2=abs(z2).^2;
i=1;
for k=1:N/2
        if z2(k)==max(z2)
            w31=k;
        end
   end

for k=1:N/2
         if (k > w31 & z2(k)>=z2(k-1)    & z2(k)>=z2(k+1) )
            w32(i)=k;i=i+1;
         end
end
n=1:N/2;
f31(num)=(w31-1)*2/N;
f32(num)=(w32(1)-1)*2/N;
end
mean31=mean(f31)
mean32=mean(f32)
var31=std(f31)^2
```

```
var32=std(f32)^2
subplot(3,1,3) ; plot((n-1)*2/N,z2(1:N/2)); grid;
```

运行上述程序可得：



频率 0.4	频率 0.43
mean11 =0.4063	mean12 =0.4375
var11 =0	var12 =0
mean21 =0.4031	mean22 =0.4609
var21 =4.3403e-005	var22 =6.1035e-004
mean31 =0.4016	mean32 =0.4516
var31 =1.1122e-004	var32 =2.9568e-004

分析上述结果可得以下结论：

(1)比较以上结果，可以发现 6 阶 AR 谱估计要比 4 阶 AR 谱估计更接近期望值，这说明 6 阶 AR 模型相对 4 阶 AR 模型能够更好地逼近观测数据模型。

(2) AR 谱估计对 0.4 谱线的估计准确度要高于 DFT 估计方法；原因是 0.4 谱线的信噪比很高，而在高信噪比情况下，AR 谱估计能很好地近似实现皮萨连柯谱估计，估计效果优于 DFT 估计方法。

(3) AR 谱估计对 0.43 谱线的估计准确度低于 DFT 估计方法。原因是 0.43 谱线的信噪比低。这也说明在低信噪比情况下，AR 谱估计不能很好地近似实现皮萨连柯谱估计，这时用 DFT 谱估计更具有优势。

5-2. 构造一组输入为白噪声加三个频率非常接近的正弦波，其信噪比为 10dB，观测长度为 256。试分别用普通 AR 谱估计和采用功率噪声抵消算法的 AR 谱估计估计这组信号的频率，并比较结果。

解：

(1)根据题意，构造信号与噪声数据：

```
N=256;SN=0.56;M=32; %采样点数为 256,信噪比为 10dB
Signal=sin(0.2*pi*[1:N])+sin(0.3*pi*[1:N])+sin(0.4*pi*[1:N]);
Noise=SN*randn(1,N);
x=Signal+Noise;
 %普通 AR 谱估计
 u=randn(N,1);
 th0=ar(x',M ,'ls');
 y=idsim(u,th0);
 z=fft(y, N)
 z=abs(z).^2;
 z=z/max(z);  %频谱归一化
 subplot(2,1,1) ;
 n=1:N/2;plot((n-1)*2/N,z(1:N/2));grid;
title('普通 AR 谱估计');
%功率噪声抵消算法
M=96;mu=0.0005;
W=zeros(1,M);
delta=6;
for k=1:length(x)-M-delta
y(k)=sum(W.*fliplr(x(k+delta:M+k+delta-1)));
    e(k)=d(k)-y(k);
    W=W+2*mu*e(k)*fliplr(x(k+delta:M+k+delta-1));
```

```
end
n=1:N/2;
for l=1:N
    aa=0;
    for k=1:M
        aa=aa+W(k)*exp(-j*k*2*pi*l/N);
    end
    Q(l)=(1/abs(1-aa))^2;
end
Q=Q/max(Q); %频谱归一化
subplot(2,1,2) ;
plot ((n-1)*2/N,Q(1:N/2));grid;
title('功率噪声抵消算法');
```

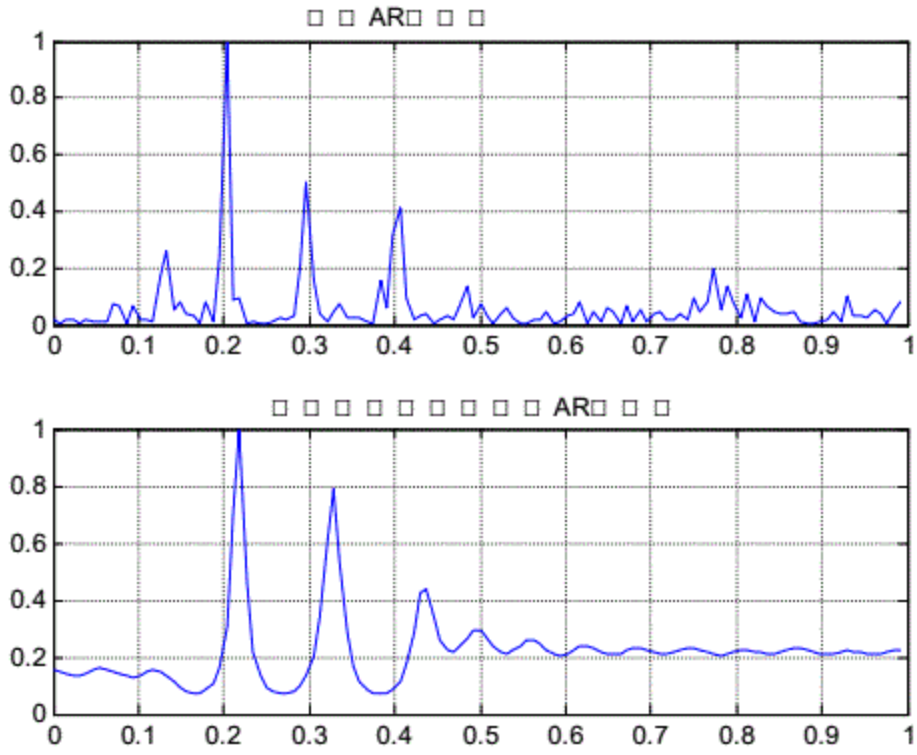



图-1　delta=6

3.仿真结果分析:

从图-1 的结果来看，无论从谱线估计精度还是从对噪声谱线的抑制作用来看，

功率噪声抵消算法的 AR 谱估计都不如普通 AR 谱估计。原因是信号延迟 delta=6 过小，没达到噪声相关时间，改为 delta=128 重新运行，得结果如图-2，此时功率噪声抵消算法的谱线估计精度有了明显提高。此外，为了更好的抑制噪声，可进行多次 LMS 迭代，图-3 给出了迭代 3 次后的结果，可见此时功率噪声抵消算法较普通 AR 谱估计的估计结果已有了明显的优势。

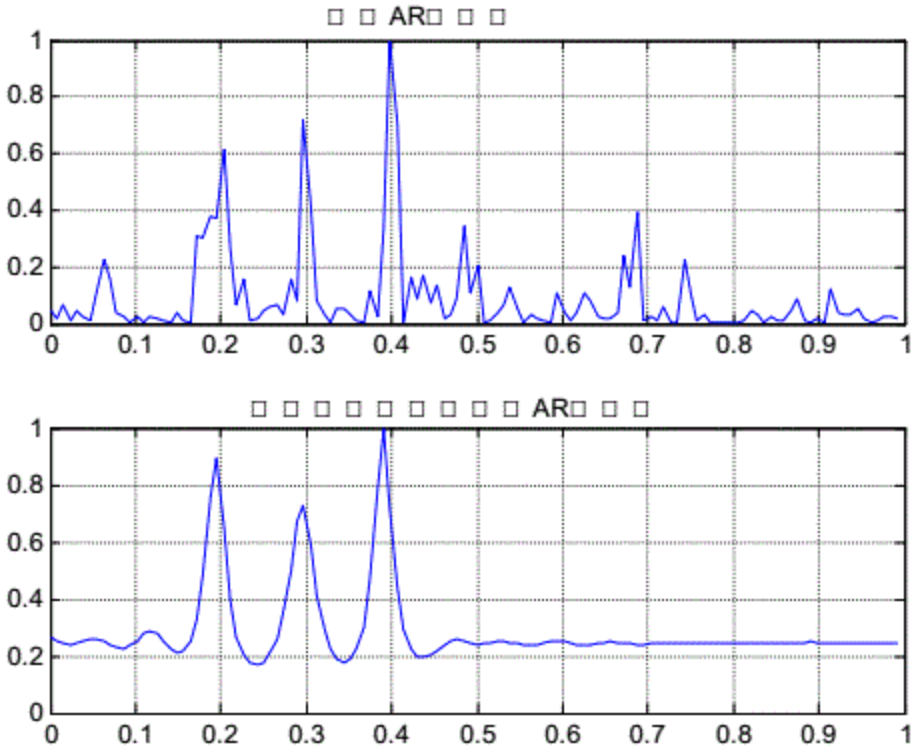


图-2　delta=128

5-3. 在自适应谱线增强器中，假设输入由两个正弦波组成，其中一个是每周期为 16 个样点，另一个是每周期 17 个样点，叠加上信噪比为 0.25 的白噪声。试采用 n=64(权系数个数)，$\mu = 0.04$，多次运行后，分别计算出 ALE 权向量谱估计器和 ALE 输出谱估计器的检验统计量，并讨论仿真结果的物理意义。

解：（1）根据题意，构造信号与噪声数据：

```
N=512; SN=2;        %采样 512 点，信噪比为 0.25
f1=17; f2=16; fs=40;
n=1:N;
Signal=(sin(2*pi*f1*n/fs)+sin(2*pi*f2*n/fs))*0.01;
d=Signal;  Noise=SN*randn(1,N)*0.01;
```

```
x=Signal+Noise;
```

2.求两种算法进行功率谱估计的检验统计量，程序如下：

```
N=512; SN=2;      %采样 512 点，信噪比为 0.25
f1=17; f2=16; fs=40;
n=1:N;
Signal=(sin(2*pi*f1*n/fs)+sin(2*pi*f2*n/fs))*0.01;
d=Signal;  Noise=SN*randn(1,N)*0.01;
x=Signal+Noise;
M=64; mu=0.04;
W=zeros(1,M);  delta=20;
for k=1:length(x)-M-delta
    y(k)=sum(W.*(fliplr(x(k+delta:M+k+delta-1))));
    e(k)=d(k)-y(k);
    W=W+2*mu*e(k)*fliplr(x(k+delta:M+k+delta-1));
end
a=-W;
Sigma=fft(W,M);          %ALE 权向量谱估计
for k=1:M
    Qxk(k)=1/(abs(1+Sigma(k)))^2;
end
Ax=mean(Qxk)
y1=y(length(x)-M-delta-M+1:length(x)-M-delta);
Sigma=fft(y1,M);         %ALE 输出谱估计
for k=1:M
    Qyk(k)=1/(abs(1+Sigma(k)))^2;
end
Ay=mean(Qyk)
```

（2）仿真结果分析

四次运行上述程序分别得：

Ax = 0.9994， Ay =0.9990； Ax = 0.9990， Ay =0.9993；

Ax = 1.0002， Ay = 1.0001； Ax = 1.0001， Ay =0.9990；

由此可见本题中的两种谱估计器，在 LMS 算法收敛到维纳解后，基本上是等效的。ALE 权向量谱估计相当于对自适应滤波器的单位脉冲响应作 DFT，得到的是滤波器的频谱，而 ALE 输出谱估计是对自适应滤波后的数据进行 DFT，得到的是滤波器输出结果的频谱，二者反映的频谱分布规律是一样的。

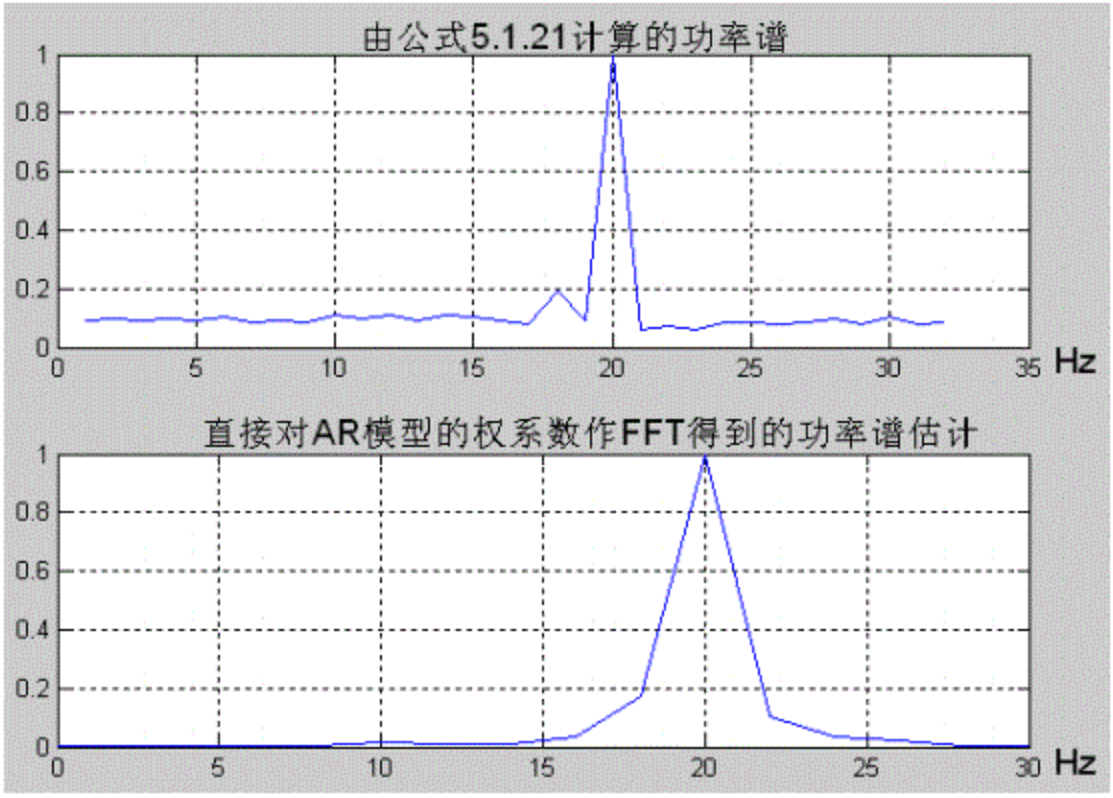
例 5-1 讨论在 AR 谱估计过程中，直接用 LMS 的权系数作 FFT 的功率谱与按式 5.1.21 计算的功率谱间的异同。

程序如下：

```
%构造信号与噪声数据
N=64; M=32;mu=0.005;
W=zeros(1,M);
Q=zeros(1,N);
f=20;S=sin(2*pi*f*[0:1/N:1-1/N]);
d=S;noise=2*randn(1,N);
x=S+noise;
%LMS 算法
delta=1;
for k=1:length(x)-M-delta
    y(k)=sum(W.*fliplr(x(k+delta:M+k+delta-1)));
    e(k)=d(k)-y(k);
    W=W+2*mu*e(k)*fliplr(x(k+delta:M+k+delta-1));
end
%按公式 5.1.21 计算的功率谱
n=1:N/2;
for l=1:N
    aa=0;
    for k=1:M
        aa=aa+W(k)*exp(-j*k*2*pi*l/N);
    end
    Q(l)=(1/abs(1-aa))^2;
```

```
end
Q=Q/max(Q);
subplot(2,1,1) ;
plot(n,Q(1:N/2));grid;

%直接用 LMS 的权系数作 FFT 的功率谱
sigma=fft(W,M);
k=1:M/2;
pp=(abs(sigma(1:M/2))).^2;
pp=pp/max(pp);
subplot(2,1,2) ;plot(2*k-2,pp);grid;
```

程序运行结果如下图所示：



结果分析如下：

（1）由于按公式 5.1.21 计算的功率谱使用了 N 点数据，而直接用 LMS 的权系数作 FFT 的功率谱使用了 M 点数据，所以的到的谱分辨率前者要高于后者。

（2）从上图可以发现按公式 5.1.21 计算的功率谱中，噪声的功率并未得到彻底的抑制，　　而直接用 LMS 的权系数作 FFT 的功率谱对噪声的抑制效果较为理想。原因是 LMS 算法的到的权系数还未收敛于维纳解。解决办法就是利用每次 LMS 算法的到的结果作初值并减小 mu 值，重新迭代，然后再按公式 5.1.21 计算的功率谱。

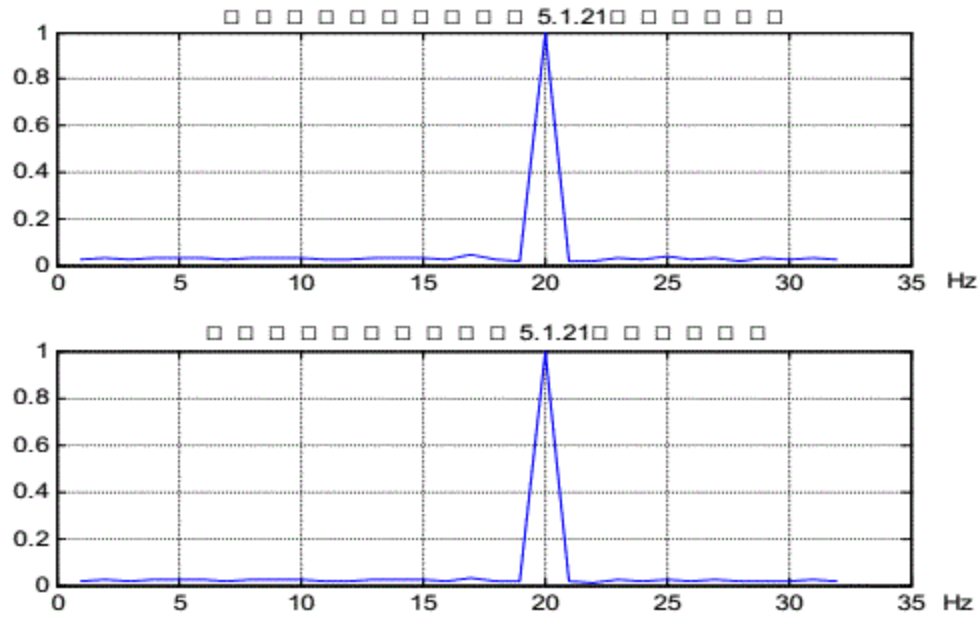
（3）实际上如果不考虑频谱幅度的差异，当 LMS 算法收敛于维纳解时，二者的频谱应该是一样的，因为这时自适应滤波器已变成只允许期望信号通过的理想带通滤波器。

程序如下：

```
mu=0.002;
%迭代第二次
delta=1;
for k=1:length(x)-M-delta
    y(k)=sum(W.*fliplr(x(k+delta:M+k+delta-1)));
    e(k)=d(k)-y(k);
    W=W+2*mu*e(k)*fliplr(x(k+delta:M+k+delta-1));
end
%迭代第三次
delta=1;
for k=1:length(x)-M-delta
    y(k)=sum(W.*fliplr(x(k+delta:M+k+delta-1)));
    e(k)=d(k)-y(k);
    W=W+2*mu*e(k)*fliplr(x(k+delta:M+k+delta-1));
end
%迭代第三次后，按公式 5.1.21 计算的功率谱
n=1:N/2;
for l=1:N
    aa=0;
    for k=1:M
        aa=aa+W(k)*exp(-j*k*2*pi*l/N);
    end
    Q(l)=(1/abs(1-aa))^2;
```

```
end
Q=Q/max(Q);
subplot(2,1,1) ;
plot(n,Q(1:N/2));grid;
%迭代第四次
delta=1;
for k=1:length(x)-M-delta
    y(k)=sum(W.*fliplr(x(k+delta:M+k+delta-1)));
    e(k)=d(k)-y(k);
    W=W+2*mu*e(k)*fliplr(x(k+delta:M+k+delta-1));
end
%迭代第四次后，按公式 5.1.21 计算的功率谱
n=1:N/2;
for l=1:N
    aa=0;
    for k=1:M
        aa=aa+W(k)*exp(-j*k*2*pi*l/N);
    end
    Q(l)=(1/abs(1-aa))^2;
end
Q=Q/max(Q);
subplot(2,1,2) ;
plot(n,Q(1:N/2));grid;
```

程序运行结果如下图：



可见多次迭代后，按公式 5.1.21 计算的功率谱已取得了比较理想的效果。

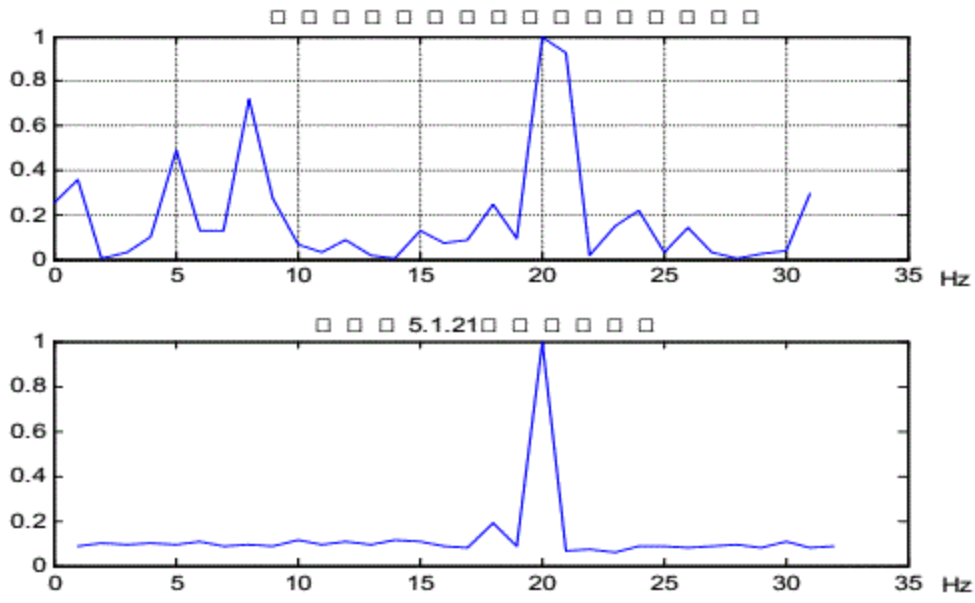
例 5—2 谱线增强器的 Matlab 程序实现

程序如下：

```
%构造信号与噪声数据
N=64; M=32;mu=0.005;
W=zeros(1,M);
Q=zeros(1,N);
f=20;S=sin(2*pi*f*[0:1/N:1-1/N]);
d=S;noise=2*randn(1,N);
x=S+noise;
%直接计算被噪声污染的数据的功率谱
z=abs(fft(x,N)).^2;z=z/max(z);
n=1:N/2;
subplot(2,1,1) ;plot(n-1,z(1:N/2));grid;
%LMS 算法
delta=6;
for k=1:length(x)-M-delta
```

```
    y(k)=sum(W.*fliplr(x(k+delta:M+k+delta-1)));
    e(k)=d(k)-y(k);
    W=W+2*mu*e(k)*fliplr(x(k+delta:M+k+delta-1));
end
%按公式 5.1.21 计算的功率谱
n=1:N/2;
for l=1:N
    aa=0;
    for k=1:M
        aa=aa+W(k)*exp(-j*k*2*pi*l/N);
    end
    Q(l)=(1/abs(1-aa))^2;
end
Q=Q/max(Q);
subplot(2,1,2) ;
plot(n,Q(1:N/2));grid;
```

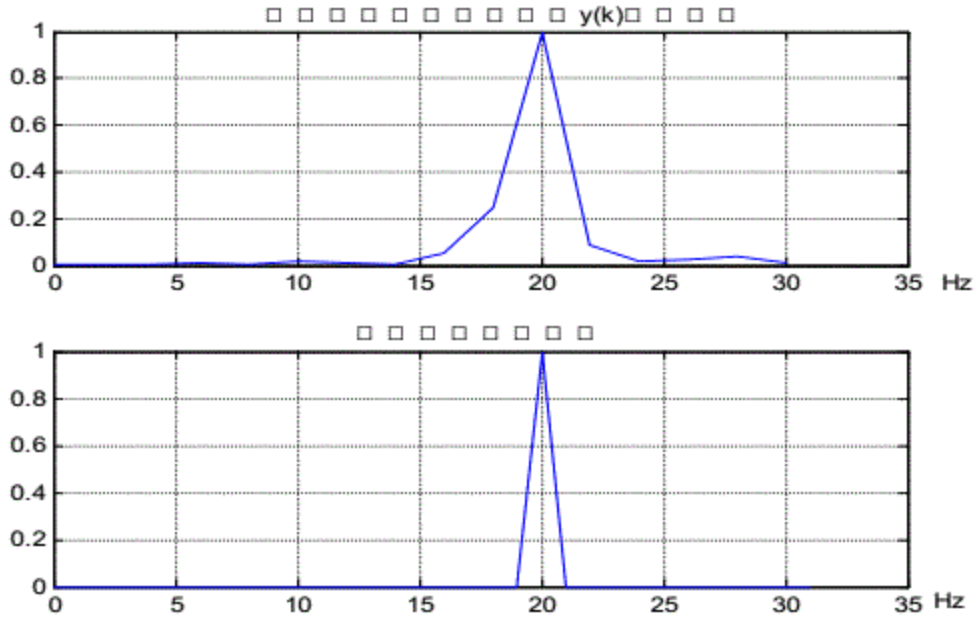
程序运行结果如下图：




```
%自适应谱线增强器输出 y(k)的功率谱
yy=[y,0,0,0,0,0,0]; yyy=(abs(fft(yy))).^2;
```

```
yyy=yyy/max(yyy);
 k=1:16; subplot(2,1,1) ;
plot(2*k-2,yyy(1:16)); grid;
%期望信号的功率谱
dd= (abs(fft(d))).^2;
dd=dd/max(dd);
n=1:N/2; subplot(2,1,2) ;
plot(n-1,dd(1:N/2)); grid;
```


程序运行结果如下图：



结论：

（1）直接计算被噪声污染的数据的功率谱，可以发现噪声的谱线与信号谱线的幅度较为接近，从而很难分辨出信号的频谱。而按公式 5.1.21 计算的功率谱中，噪声和信号谱线区别鲜明，达到了谱线增强的效果。

（2）直接计算自适应谱线增强器输出 y(k)的功率谱，可以发现其噪声和信号谱线区别也很鲜明，而且其噪声谱得到了很好的抑制，这正是 LMS 自适应滤波器所具有的功能。然而由于 y(k)的点数为 length(x)-M-delta（在本程序中为 64－32－6＝26），经补零得 32 点再进行 FFT，所以得到的谱分辨率不高。