北京理工大學
Beijing Institute of Technology

本科实验报告

实验名称：MUSIC 线谱估计方法仿真

<table>
<tr><td>课程名称：</td><td>现代信号谱估计导论</td><td>实验时间：</td><td>2019.05.28</td></tr>
<tr><td>任课教师：</td><td>许文龙</td><td>实验地点：</td><td></td></tr>
<tr><td>实验教师：</td><td>许文龙</td><td rowspan="2">实验类型：</td><td rowspan="2">■原理验证
□综合设计
□自主创新</td></tr>
<tr><td>学生姓名：</td><td>刘仕聪</td></tr>
<tr><td>学号/班级：</td><td>1120161380</td><td>第一组：</td><td></td></tr>
<tr><td>学　院　：</td><td>信息与电子学院</td><td>同组搭档：</td><td></td></tr>
<tr><td>专　业　：</td><td>电子信息类（实验班）</td><td>成　绩　：</td><td></td></tr>
</table>

信息与电子学院
SCHOOL OF INFORMATION AND ELECTRONICS

MUSIC 线谱估计方法仿真

一、 实验目的

1. 实现基于 MUSIC 算法的 DoA 估计以及多频率分类
2. 算法性能的简单分析

二、 实验原理

1. 功率谱估计方法简介

谱估计是信号处理中一个重要应用。对于接受到的信号，我们有时并不需要重建出原有的信号，而是需要根据接收到的信号对一些特征进行估计分析。这类应用可以统称为谱分析，其中谱并不仅限于功率谱，还可以是其他参数的谱线。例如在 DoA 估计问题中，谱还可以是线谱，是信号方向的函数。此处以功率谱估计为例，介绍简单的谱估计方法。

一般而言简单的功率谱分析方法包括传统估计方法和现代估计方法。传统方法包括周期图法和相关图法，以及这些方法的扩展方法；现代谱估计方法主要是基于一定信号模型的参数化估计方法，其中典型的模型有 AR 模型，MA 模型和 ARMA 等。

周期图法实际上本质是直接对傅里叶变换结果进行平方处理，表达式如下

$$\hat{\phi}_p(\omega)=\frac{1}{N}\left|\sum_{t=1}^{N}y(t)e^{-j\omega t}\right|^2$$

周期图法由于采用了傅里叶变换的结构，因此可以使用 FFT 快速实现。由于周期图方法依赖于窗函数设计，因此存在较多的窗函数改进方法。

相关图方法本质是对相关函数的傅里叶变换。我们知道相关函数与功率谱是一对傅里叶变换，需要实现估计时，只需要我们对相关函数做出合理估计，就能估计出功率谱。其计算方法为

$$\hat{\phi}_c(\omega)=\sum_{k=-(N-1)}^{N-1}\hat{r}(k)e^{-j\omega k}$$

此处可操作对象是不同的相关函数的估计方法，常用方法有标准无偏估计和标准有偏估计，分别相当于周期图中采用三角窗和矩形窗的方法。可以证明，在采用标准有偏估计时，相关图方法与周期图方法等价。

现代谱估计方法通过计算模型中的参数实现估计。此处以 ARMA 模型为例分析。ARMA 信号模型可以表示为

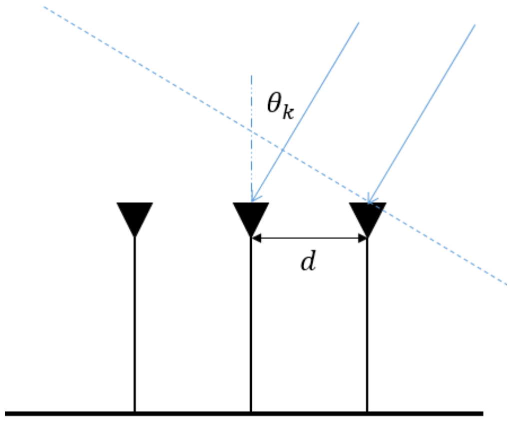
$$y(n)=\frac{B(z)}{A(z)}u(n)$$

若仅采用$A(z)$则为 AR 模型（自回归），仅采用$B(z)$则为 MA 模型（滑动平均）。实际上三种模型可以相互转化。

2. MUSIC 算法简介

经典谱估计方法依赖于窗函数设计来决定分辨率，当需要高分辨率时又会面临方差较大的问题；现代谱估计方法虽然在分辨率上有更大优势，但是需要 ARMA 模型的准确阶数的估计，这需要较为复杂的计算。MUSIC 方法是一种阵列天线信号处理的超分辨率算法，在算法复杂度和分辨率上远超上述传统算法。

对于一般阵列模型



同一信号在两个阵元上的波程差为

$$\Delta x=dsin\theta$$

时延为

$$\Delta t=\frac{\Delta x}{c}$$

如果入射波波长λ已知，容易得到同一信号在不同阵元上接收到的结果是

$$
\begin{bmatrix} x_1(t) \\ x_1(t)e^{-j\omega\tau} \\ x_1(t)e^{-j\omega 2\tau} \\ \cdots \\ x_1(t)e^{-j\omega(m-1)\tau} \end{bmatrix} = \begin{bmatrix} y_1(t) \\ y_1(t) \\ y_2(t) \\ \cdots \\ y_m(t) \end{bmatrix} = \begin{bmatrix} \\ \\ a(\theta_1) \\ \\ \\ \end{bmatrix} x_1(t)
$$

对于不同入射信号，我们可以写成矩阵的形式（同时考虑接收机上的等效总噪声）

$$
Y = AX + N
$$

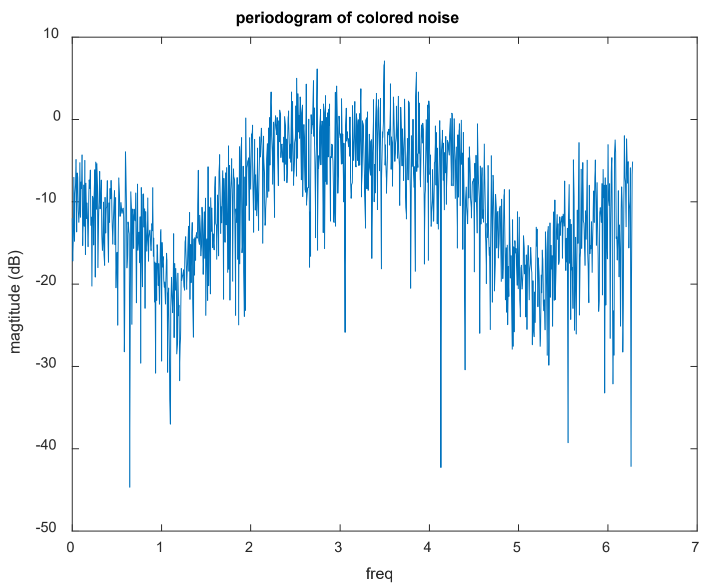
$$
\begin{bmatrix} Y_1 \\ Y_2 \\ \cdots \\ \cdots \\ Y_M \end{bmatrix} = \begin{bmatrix} & & & \\ & & & \\ a(\theta_1) & a(\theta_2) & \ldots & a(\theta_D) \\ & & & \\ & & & \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \cdots \\ \cdots \\ X_D \end{bmatrix} + \begin{bmatrix} N_1 \\ N_2 \\ \cdots \\ \cdots \\ N_M \end{bmatrix}
$$

以上是简单的阵列信号模型，MUSIC 算法输入的信号是向量Y，我们要根据信号Y对信号特征做出估计。MUSIC 算法的一个开创性思想是采用了特征值分解的方法，借助信号子空间与噪声特征向量的正交关系，对信号方向、频率、数量等作出渐近无偏估计。作者通过对特征向量之间的关系的证明，证明了方法的正确性，并得到了远超传统方法的结果。

三、 实验内容

a. 噪声信号的生成与绘制

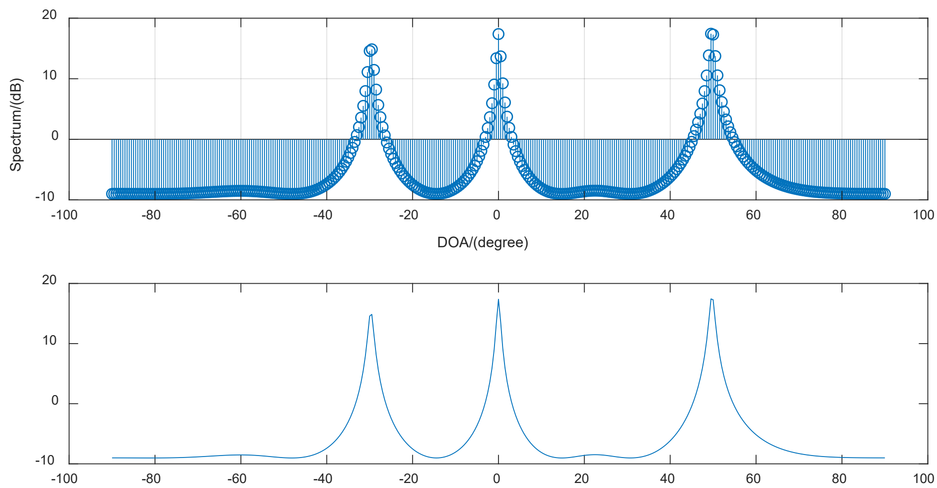
首先生成模型中较为简单的噪声项。由于作者在论文中介绍过，对于不同的噪声模型，本算法都有足够的鲁棒性，因此采用色噪声代替白噪声。色噪声产生我在其他报告中已经详细介绍，此处不加赘述。



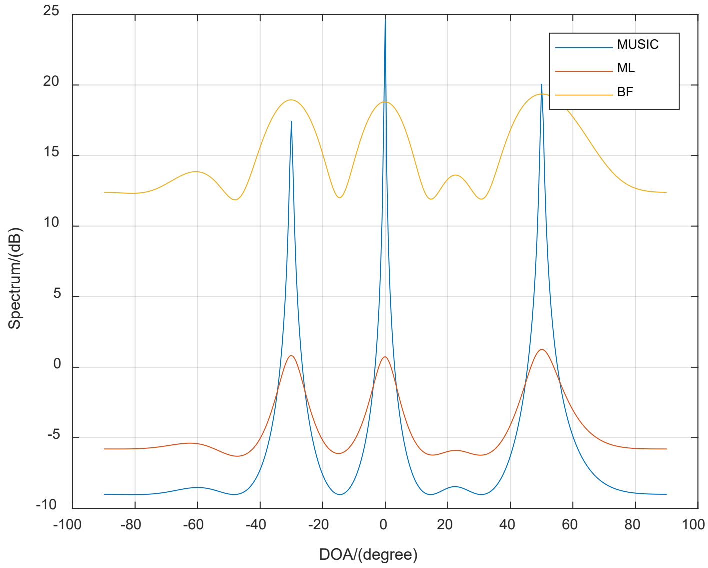
噪声功率谱如图。

b. MUSIC 算法 DOA 估计

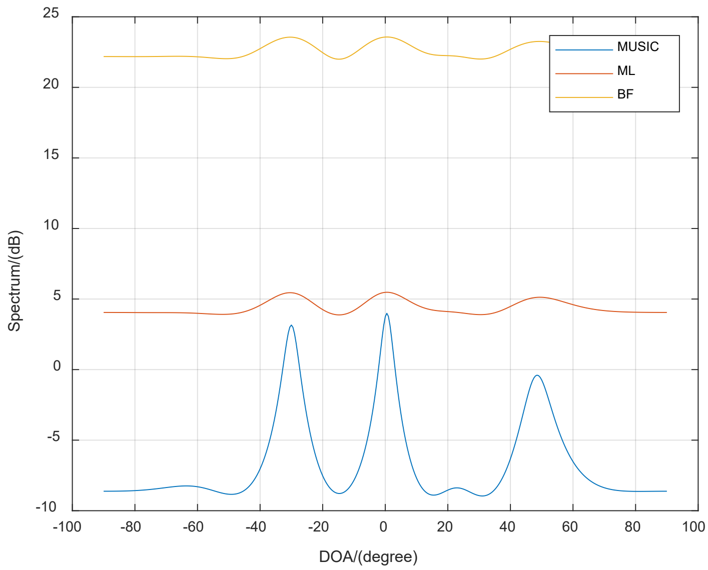
假设信号方向分别为-30 度、0 度和 50 度，幅度相同，在 MUSIC 算法下得到线谱及其包络为



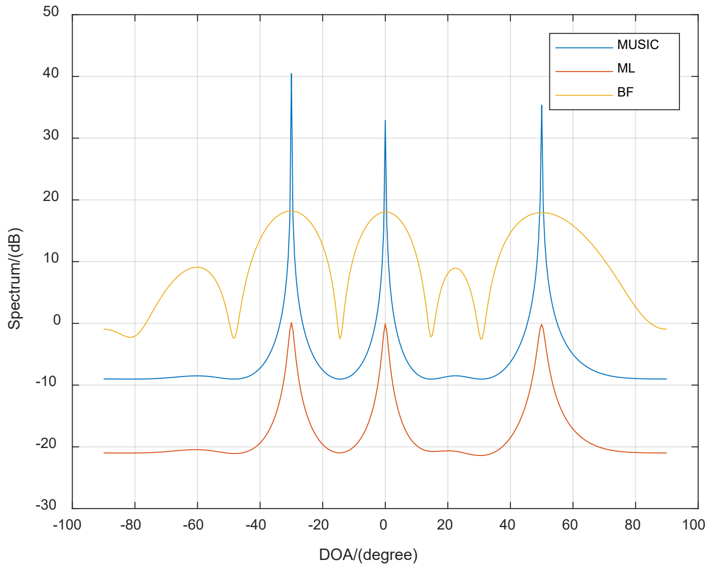
图线中可以明显看出，对方向的估计十分准确。将 MUSIC 方法与其他传统方法进行比较如下



该结果是在信噪比为 0dB 的条件下测得，以上仿真也均为 0dB 信噪比。实际应用中接收信号的信噪比可能更小，此时 MUSIC 算法以外的普通方法可能就不能得到较好的估计：



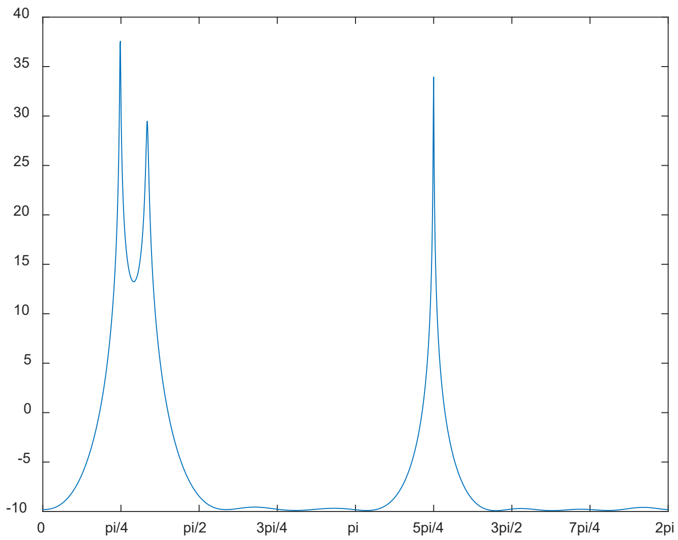
以上估计的信噪比-10dB，此时只有 MUSIC 算法还能提供估计结果。可以发现，MUSIC 算法不仅仍可以检测到信号方向，而且还能提供偏差较小的估计。在典型的通信场景中，以 15dB 为参考信噪比，仍然是 MUSIC 算法效果最优



c. 进行多正弦信号频率估计

MUSIC 算法的应用较多，此处给出论文中的频率估计仿真。

假设信号归一化频率为 pi/4、pi/3 和 5pi/4，幅度分别为 4、2 和 3。估计结果如下



对估计结果进行处理，计算幅度得到估计幅度为

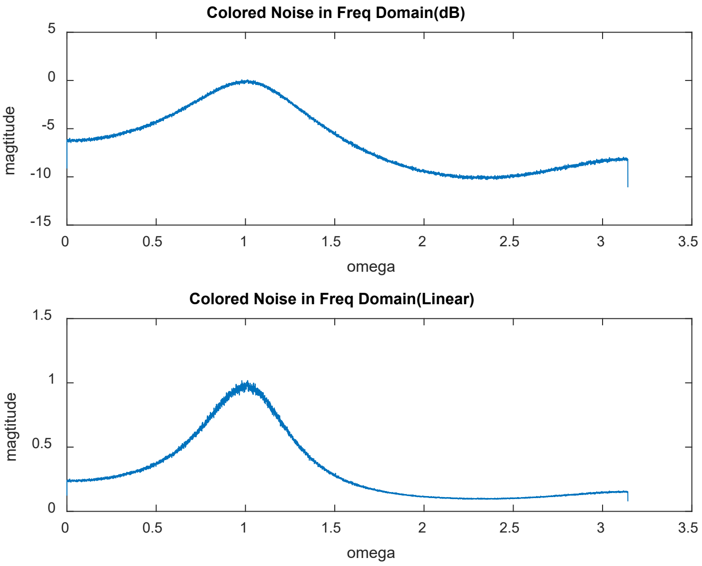
$A =$

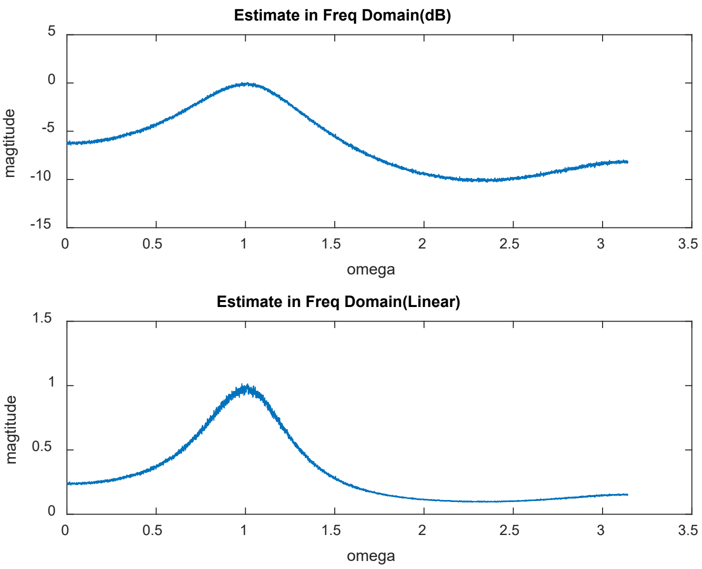
$4.0464 \quad 2.1434 \quad 2.9966$

四、 问题与思考

1. 传统估计方法

此处采用的对比是传统方法的估计结果，而非真实谱。后采用统计平均方法求得真实谱进行对比，其结果如下





此部分代码在附录，运行时间较长。

2. ARMA 模型很依赖于对阶数的估计准确程度。当阶数较为精确的时候，我么得到的结果较好；当阶数估计不准确的时候，估计结果也较差。这一点是 ARMA 模型的缺陷，也是 MUSIC 等线谱估计算法的优势所在。

五、 附录

本次仿真的代码如下

```
clc; clear; close all;
L=1024;  % length
c = [1 -0.5];
nc = length(c) - 1;
xik=zeros(nc,1);  % init
xi=randn(L,1);  % 0 mean sigma=1

for k=1:L
    e(k)=c*[xi(k);xik];  %colored
    % update
    for i=nc:-1:2
        xik(i)=xik(i-1);
    end
    xik(1)=xi(k);
end
%============================================
derad = pi/180;
N = 8;               % Arrays
M = 3;               % Signals
theta = [-30 0 50];  % Directions
snr = 15;            % SNR
K = 1024;            % Quick shots

dd = 0.5;            % array distance/lambda, d=lambda/2
d=0:dd:(N-1)*dd;      % d here is md/lambda
A=exp(-1i*2*pi*d.'*sin(theta*derad));

S=randn(M,K);
X=A*S;
sigma2=(var(S(1,:))+var(S(2,:))+var(S(3,:)))/3/(10^(snr/10))
var(S(3,:))
WN=wgn(N,K,sigma2,'linear','complex');
e=zeros(N,K);
for i=1:N
    for j=3:K
        e(i,j)=WN(i,j)-0.7*WN(i,j-1)+0.7*WN(i,j-2);
```

```
    end
end

% X1=awgn(X,snr,'measured'); % X=AF+W
X1=X+e;
Rxx=X1*X1'/K;
%InvS=inv(Rxx);
[EV,D]=eig(Rxx);    % eigval vec
EVA=diag(D)';
[EVA,I]=sort(EVA);  % small -> large
%EVA=fliplr(EVA);   % flip
EV=fliplr(EV(:,I)); % flip in column
for iang = 1:361    % iteration
       angle(iang)=(iang-181)/2;
       phim=derad*angle(iang);
       a=exp(-1i*2*pi*d*sin(phim)).';
       a2=exp(-1i*2*pi*d*cos(phim)).';
       L=M;
       En=EV(:,L+1:N);
       %SP(iang)=(a'*a)/(a'*En*En'*a);
       SP(iang)=1/(a'*En*En'*a);
       b=[a';a2'];
       SP4=1./b*En*En'*b';
       SP4_1(iang)=SP4(1,1);
       SP4_2(iang)=SP4(1,2);
       SP4_3(iang)=SP4(2,1);
       SP4_4(iang)=SP4(2,2);

       SP2(iang)=1/(a'*inv(Rxx)*a);
       SP3(iang)=a'*Rxx*a;
end

SP=abs(SP);
%SPmax=max(SP);
%SP=10*log10(SP/SPmax);  % normalized
SP=10*log10(SP);
h=plot(angle,SP);
set(h,'Linewidth',0.5);
xlabel('DOA/(degree)');
ylabel('Spectrum/(dB)');
%axis([-100 100 -40 60]);
set(gca, 'XTick',[-100:20:100]);
grid on
hold on;
```

```
%%
SP2=abs(SP2);
%SPmax=max(SP);
%SP=10*log10(SP/SPmax);  % normalized
SP2=10*log10(SP2);
h2=plot(angle,SP2);
set(h2,'Linewidth',0.5);
xlabel('DOA/(degree)');
ylabel('Spectrum/(dB)');
%axis([-100 100 -40 60]);
set(gca, 'XTick',[-100:20:100]);
grid on
hold on;

SP3=abs(SP3);
%SPmax=max(SP);
%SP=10*log10(SP/SPmax);  % normalized
SP3=10*log10(SP3);
h3=plot(angle,SP3);
set(h3,'Linewidth',0.5);
xlabel('DOA/(degree)');
ylabel('Spectrum/(dB)');
%axis([-100 100 -40 60]);
set(gca, 'XTick',[-100:20:100]);
grid on
legend('MUSIC','ML','BF')

%========================
figure
[y1,f1] = periodogram(e(1,:));
p1 = 10*log10(y1);
plot(f1,p1)
xlabel('freq')
ylabel('magtitude (dB)')
title('periodogram of colored noise')
```

以下是频率估计

```
clear;
close all;
%Frequency Estimation by Eigendecomposition of Autocorrelation
Matrix
N_x=128; % Length of Signal
```

```
N=10; % Size of Rx Matrix
A=[4 2 3];
w=[pi/4 pi/3 5*pi/4]';
phase=[pi/3*ones(1,N_x);pi/6*ones(1,N_x);pi/5*ones(1,N_x)];
M=3; % Number of Signals
x=randn(1,N_x)+A*exp(j*(w*[0:N_x-1]+phase));

Cx=xcorr(x,'biased');
Rxx=Cx(N_x:N_x+N-1)';
Rx=toeplitz(Rxx);

[V,D] = eig(Rx); %Eigendecomposition  ????
D=sum(D);
Nw=1000;
ww=[0:2*Nw]/Nw*pi;
e=exp(-j*ww'*[0:N-1]);
ev=e*V(:,1:N-M);
Pw=1./real(diag(ev*ev')');
figure
plot(ww,10*log10(Pw));
xlim([0 2*pi])
set(gca,'XTick',0:pi/4:2*pi)
set(gca,'XTickLabel',{'0','pi/4','pi/2','3pi/4','pi','5pi/4','
3pi/2','7pi/4','2pi'})

S=mean(D(1:N-M))
E=exp(j*[0:N-1].'*w');
P=real(E\(Rx-(eye(N).*S))/E');
A=sqrt(sum(P))
```