

阵列信号处理**MATLAB**编程

阵列信号处理**MATLAB**编程

例8. 1： 传统波束成形

%Classical_Beamforming_Method

```
clear;  
clc;  
close all;  
ns=1; %nombre de sources  
nc=8; %nombre de capteurs  
nd=200; %nombre d' échantillons  
% snr=10; %le rapport signal sur bruit  
theta(1)=60; %le fisement exprimé en degré  
% theta(2)=7;  
% theta(3)=7;
```

```
snr=-20:30;  
cal=1:100;
```

```
est_m=zeros(length(cal),length(snr));  
est_r=zeros(length(cal),length(snr));  
est_e=zeros(length(cal),length(snr));
```

```

for iii=1:length(snr)
    for iiii=1:length(cal)
        Xc=gene_source(ns,nc,nd,theta,snr(iiii));
        %calcul de la matrice decovariance
        R=zeros(nc,nc);
        for i=1:nd
            R=R+Xc(:,i)*Xc(:,i)';
        end
        R=R/nd;
        [R_e_vec,R_e_val]=eig(R);
        [rds,i]=sort(diag(R_e_val));
        R_e_vec1=R_e_vec(:,i);
        V_noise=R_e_vec1(:,(1:nc-ns));
        theta1=[0:90];          %d_theta
        fs=sind(theta1)/2;
        ii=[0:nc-1]';
        d_theta=exp(j*2*pi*ii*fs); %position errors between -0.05
        Pmusic=d_theta'*V_noise*V_noise'*d_theta;
        P=abs(diag(Pmusic));
        for k=1:length(P)
            P(k)=inv(P(k));
        end
    end
end

```

```

P1=max(P);
P2=10*log10(P/P1);
est_m(iii,iiii)=(find(P2==0)-1);
    %k=0:(nc-1);
[U,D]=eig(R);
[rds,i]=sort(abs(diag(D))); %l'ordre croissant!!attention
D1=D(i,i);
U1=U(:,i);
Us=U1(:,(nc-ns+1):nc); %matrice de l'espace bruit
order1=1:(nc-1);
order2=order1+1;
Us1=Us(order1,:);
Us2=Us(order2,:);
psi=inv(Us1'*Us1)*Us1'*Us2;
[ksi,phi]=eig(psi);
phi=diag(phi);
ys=asind(angle(phi)/pi);
est_e(iii,iiii)=ys;
end
end

```

```
function [Xc] = gene_source(ns,nc,nd,theta,snr)
```

```
    k=0:(nc-1);
```

```
    for ii=1:nd      % nd:nombre d' échantillons
```

```
        fs=sind(theta)/2;
```

```
        Xc(:,ii)=exp(2j*pi*k'*fs)*(randn(ns,1)*sin(rand*pi)); % dimension (15,1000)
```

```
    end
```

```
    Xc=Xc+(10^((-snr)/20))*randn(nc,nd);
```

```
end
```

