```
%%%Coherent Source Direction-Finding using a monostatic MIMO array with velocity receive sensors
%%%%    2019 7 15 Edited by Yuwei Song

clc;
clear all;
close all;
M=6;                    %transmit antennas number
N=6;                     %receive antennas number
snap=100;           % snapshots
monte=100;         %MONTE KALOR
a=[0:M-1]';         %transmit array vector
b=[0:N-1]' ;        %receive array vector
c=3e8;
f=300e6;
lamda=c/f;              %wavelength
dr=lamda/2 ;              %inter-element spacing at receiver
dt=lamda/2 ;              %inter-element spacing at transmitter


theta=[10 20]*pi/180;phi=[15    25]*pi/180; %the parameters of source
  % theta=[80 60]*pi/180;phi=[90    0]*pi/180; %the parameters of source

P=length(theta);                  %nmber of source

u=(sin(theta).*cos(phi));
v=(sin(theta).*sin(phi));
w=cos(theta);
p=[u; v ;w];
a_t=exp(j*2*pi/lamda*dt*a*u); %transmitter steering vector      x
a_r=exp(j*2*pi/lamda*dr*b*v);      %receiver steering vector      y

A=[kron(p(:,1),kron(a_t(:,1),a_r(:,1))) kron(p(:,2),kron(a_t(:,2),a_r(:,2)))];

s=exp(j*rand(1,snap)*2*pi);    %signals vector
s=[s ;s];
tic

SNR=[0:2:30];        %signal to noise ratio
for ii=1:monte
    ii
    noise=(randn(3*M*N,snap)+j*randn(3*M*N,snap))/sqrt(2);
    for kk=1:length(SNR)
        x = 10^(SNR(kk)/20)*A*s+noise;      % observation vector
```

```
R_xx = x(1:M*N,:)*x(1:M*N,:)'/snap;              % the covariance matrix
R_yy = x(M*N+1:2*M*N,:)*x(M*N+1:2*M*N,:)'/snap;             % the covariance matrix
R_zz = x(2*M*N+1:3*M*N,:)*x(2*M*N+1:3*M*N,:)'/snap;      % the covariance matrix

R_x = (R_xx+R_yy+R_zz)/3;
%                   R_x=x*x'/snap;
[V,D] = eig(R_x);           % eigenvalue decomposition
[D,index] = sort(diag(D));    %%%
save eig80 D

en = V(:, index(1:M*N-P));      %%
es = V(:, index(M*N-P+1:M*N));      % noise and signal subspace

%% Paper name--- Joint DOD-DOA estimation using combined esprit_music approach in MIMO radar

Es1=es((1:M*N-N),:);%%% TLS-ESPRIT
Es2= es(N+1:M*N,:);
RzzEs1Es2 = [Es1'; Es2'] * [Es1 Es2];
[EVec, EVal] = eig(RzzEs1Es2);
[EVal, Indx] = sort(diag(EVal).');
for Col = 1 : 2*P
        E(:,Col) = EVec(:,Indx(2*P+1-Col));
end;
E12 = E(1:P,P+1:2*P);
E22 = E(P+1:2*P,P+1:2*P);
[Vt,THETA_u] = eig(-1*E12/E22);
THETA_u=diag(THETA_u);          %%%%end estimation u

for Row=1:M
        frst = (Row-1)*(N-1) + 1;
        last = Row*(N-1);
        EsTop(frst:last,:) = es(frst+(Row-1):last+(Row-1),:);
        EsBot(frst:last,:) = es(frst+Row:last+Row,:);
end;
RzzEs1Es2 = [EsTop'; EsBot'] * [EsTop EsBot];
[EVec, EVal] = eig(RzzEs1Es2);
[EVal, Indx] = sort(diag(EVal).');
for Col = 1 : 2*P
        E(:,Col) = EVec(:,Indx(2*P+1-Col));
end;
E13 = E(1:P,P+1:2*P);
E23 = E(P+1:2*P,P+1:2*P);
[Vr,THETA_v] = eig(-1*E13/E23);
```

```
        THETA_v=diag(THETA_v);    %%%%end estimation v

        pair=abs(inv(Vt)*(Vr));

        u_est = angle(THETA_u)/pi;    %paired processing

        for iii=1:P
            index=find(pair(iii,:)==max(pair(iii,:)));
            v_est(iii)=angle(THETA_v(index))/pi;
        end

        theta_est=asin(sqrt(u_est.'.^2+v_est.^2))*180/pi;

        phi_est=angle(u_est.'+j*v_est)*180/pi;


        err_theta(:,ii,kk)=abs(sort(theta_est,'descend') -sort(theta*180/pi,'descend')).^2;
        err_phi(:,ii,kk)=abs(sort(phi_est,'descend') -sort(phi*180/pi,'descend')).^2;
    end
end



err_theta_RMSE=squeeze(sqrt(sum(err_theta,2)/monte));
err_phi_RMSE=squeeze(sqrt(sum(err_phi,2)/monte));



figure;axes('Fontsize',14)
semilogy(SNR,err_theta_RMSE(1,:),'*-',SNR,err_theta_RMSE(2,:),'^-',...
    SNR,err_phi_RMSE(1,:),'s-',SNR,err_phi_RMSE(2,:),'o-',...
    'MarkerSize',9,'LineWidth',1.5);hold on;
xlabel('SNR (dB)');ylabel('RMSE    (\circ)');grid on;box off;
legend('Azimuth of source 1',' Azimuth of source 2',' Elevation of source 1',' Elevation of source 2');grid off
% ylim([2*10^-4 1])

save rlarge err_theta_RMSE err_phi_RMSE
```