```
close all; clear all; clc;
%% System parameter
L = 2; % Number of path L0 is LOS
BW = 100; % Bandwidth
N = 1; % Number of subcarrier
Ts = 1/BW; % Sampling Period
Nt = 32; % Number of Tx
Nr = Nt; % Number of Rx
Nb=Nt*2; % Number of beams in dictionary;
Ns = 20; % Number of beams sent
posTx = [0 0]'; % position of Tx
posRx = [3 1]'; % position of Rx
c = 300; % Speed of light meter/us
sigma=0.1; % Noise standard deviation
%% Scatter point
SP = [2 2]; % random points uniformly placed in a 20 m x 20 m area
%% Channel parameter
TOA(1) = norm(abs(posRx))/c; % t = s/v, LOS Time of arrival
AOD(1) = atan2(posRx(2), posRx(1)); % LOS Angle of departure
AOA(1) = atan2(posRx(2), posRx(1)) - pi; % LOS Angle of arrival
for i=1:L-1
  AOD(i+1) = atan2(SP(i, 2), SP(i, 1));
  AOA(i+1) = atan2(abs(SP(i, 2)-posRx(2)), abs(SP(i, 1)-posRx(1)));
  TOA(i+1) = (norm(SP(i,:)-posTx)+norm(posRx-SP(i,:)))/c;
end
h=10*ones(1,L); % Channel coefficient
%% Channel
H=zeros(Nr,Nt,N);
for i=1:N % i-th subcarrier
  g = zeros(L, L); % gamma
  At = []; % angular domain correlation matries of tramsmitter
  Ar = []; % angular domain correlation matries of arriver
  for k=1:L % k-th path
      g(k, k) = sqrt(Nt*Nr) * h(k) * exp(-j*2*pi*(i-1)*TOA(k)/(N*Ts));
      At = [At, getResponse(Nt, sin(AOA(k)))];
      Ar = [Ar, getResponse(Nr, sin(AOD(k)))];
  end
  At = At';
  H(:,:,i) = Ar * g * At;
end
%% Create Dictionary
Ut=zeros(Nt,Nb);
Ur=zeros(Nr,Nb);
aa=-Nb/2:Nb/2-1;
aa=2*aa/Nb; % dictionary of spatial frequencies
for m=1:Nb
  Ut(:,m)=getResponse(Nt,aa(m))*sqrt(Nt);
```

```
  Ur(:,m)=getResponse(Nr,aa(m))*sqrt(Nr);
end
%% Visualize the beamspace channel for 1 subcarrier in AOA/AOD space
figure;
subplot(1, 2, 1);
Hb=Ur'*H(:,:,1)*Ut;
mesh(asin(aa),asin(aa),abs(Hb));
xlabel('AOD'); ylabel('AOA');
%% Generate the observation and beamformers
y=zeros(Nr,Ns,N);
F=zeros(Nt,Ns,N);
for k=1:Ns
  for n=1:N
     F(:,k,n)=exp(1j*rand(Nt,1)*2*pi); % beamformers
     y(:,k,n)=H(:,:,n)*F(:,k,n)+sigma/sqrt(2)*(randn(Nr,1)+1j*randn(Nr,1)); % y = HFx + N
  end
end
%% Vectorize and generation of the basis
Omega=zeros(Nr*Ns,Nb*Nb,N);
yb=zeros(Nr*Ns,N);
h = Hb(:); % Vecterize
for n=1:N
  Omega(:,:,n) = kron((Ut'*F(:,:,n)).', Ur);
  yb(:,n) = Omega(:,:,n)*h + sigma/sqrt(2); % * (randn(Nr*Ns, 1)+1j*randn(Nr*Ns, 1)); %
y=Omega*h
end
%% OMP
% [Omega_k, x_k] = OMP(yb, Omega, L);
% hb = zeros(size(Hb));
% for i=1:length(Omega_k)
%     hb(Omega_k(i)) = x_k(i);
% end
% subplot(1, 2, 2);
% mesh(asin(aa),asin(aa),abs(hb));
% xlabel('AOD'); ylabel('AOA');
[xk, ~, ~] = solveLASSOProblem(Omega,yb,1000);
hh = zeros(Nb, Nb);
for i=1:length(hh)
  for j=1:length(hh)
     hh(i, j) = xk(i+Nb*(j-1));
  end
end
subplot(1, 2, 2);
mesh(asin(aa),asin(aa),abs(hh));
xlabel('AOD'); ylabel('AOA');
```