SPCOM homework 1

Yifan Li(5145147), Yanbin He(5037751)

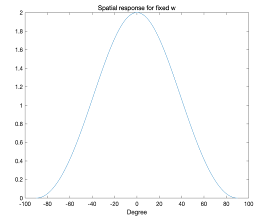
I. Results for Instantaneous model

A. Spatial Response

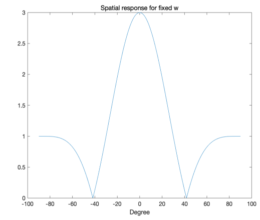
In this section, we will show our results about spatial responses and reproduce the graphs in chapter 1 on the slides. As we can see, the Figure 1 to Figure 3 are the same as the ones shown on the slides.

The discussions are the following:

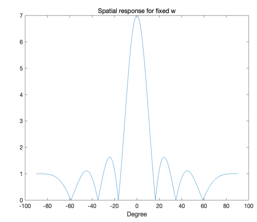
- The pictures shown in Figure 1 are the spatial response for a fixed $\mathbf{w}$. Except for Figure (d), all the $\mathbf{w}$ are set as all one vector. The meaning of these pictures are the ability of receiving energy from different directions instead of there is a source coming from that direction. But the pictures illustrated in Figure 2 and 3 show the detection of two separated sources. As we can see, using beamformer, if sources are well separated, they can be distinguished. However, the resolution is insufficient.
- If we keep Delta fixed and increase the number of antennas, the width of mainlobe decreases. And we will have a better resolution. However, sidelodes appear.
- If we increase the value of Delta and make it greater than 0.5, then there occurs multiple peaks leading to ambiguity. Thus, if we want to get rid of this phenomenon, we should keep Delta less than or equal to 0.5.



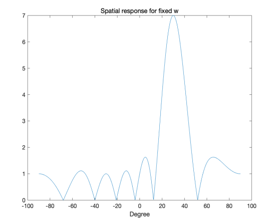
(a) M = 2, Delta = 0.5



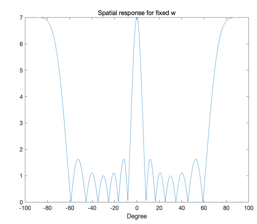
(b) M = 3, Delta = 0.5



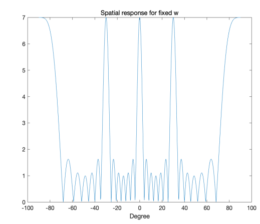
(c) M = 7, Delta = 0.5



(d) $\mathbf{w} = \mathbf{a}(30°)$, M = 7, Delta = 0.5



(e) M = 7, Delta = 1



(f) M = 7, Delta = 2

Fig. 1: Reproduction of single source graphs in chapter 1

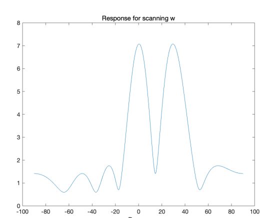


Fig. 2: Reproduction of two separated source graphs M = 7, Delta = 0.5, α = [0 30]

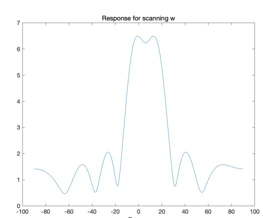


Fig. 3: Reproduction of two separated source graphs M = 7, Delta = 0.5, α = [0 12]

B. SVD of X

In this section, we will show the results of singular values of matrix of $\mathbf{X}$ in terms of varying DOA separation, number of antennas, number of samples and SNR. Here we choose $d = 2$sources, $M = 5$ antennas, $N = 10$ samples.

In our implementation, we keep the noise power fixed. And then we use the SNR to calculate the signal power. Figure 4 shows how the singular values will change when varying the SNR. When SNR gets larger, the signal power will get larger. Due to the fact that the noise power is fixed, the first two singular values will go upward and the gap between the second and the third will also get larger.

Figure 5 shows the results of changing DOA separation. The main result we would like to show is that if the two signals are orthogonal to each other, then the first two singular values tend to be the same. And with the increase of the DOA separation, the first singular will get larger while the second one will get smaller. And the gap between the second and the third value are getting closer as well.

Figure 6 and 7 are about the results of varying the number of antennas and samples, respectively. In both cases, the singular values go higher. We think that this is because more samples and antennas leads to the increase of receiving power.

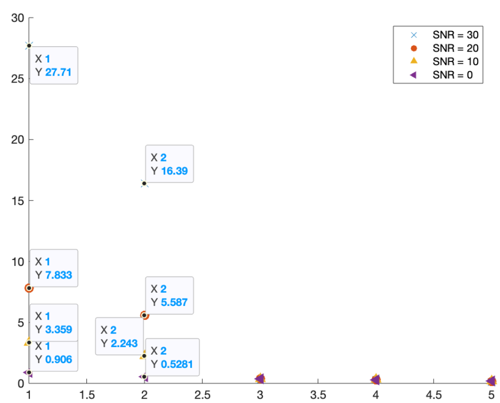


Fig. 4: Singular values of $\mathbf{X}$ when varying SNR

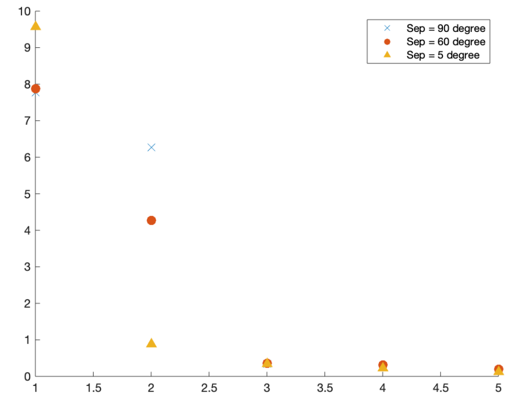


Fig. 5: Singular values of $\mathbf{X}$ when varying DOA separation

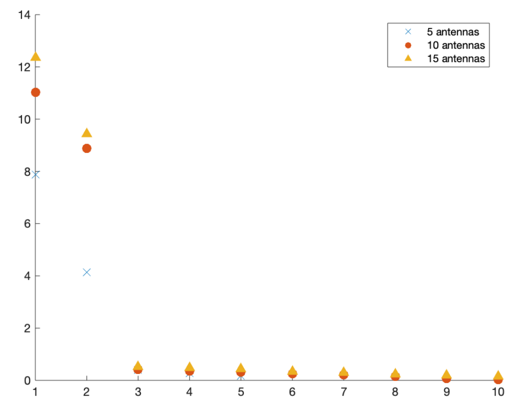


Fig. 6: Singular values of $\mathbf{X}$ when varying the number of antennas

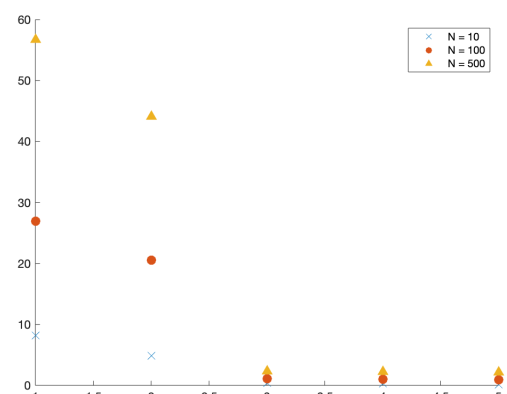


Fig. 7: Singular values of $\mathbf{X}$ when varying the number of samples

II. RESULT FOR CONVOLUTIVE MODEL

Figure 8 shows our plot of the real and imaginary parts of **h** and **x**. For the visual inspection of source **s** from **x**. We did see the characteristics of PSK since the value of the real and imaginary fluctuate greatly around range -1 to 1, but we did not identify any features related to QPSK.

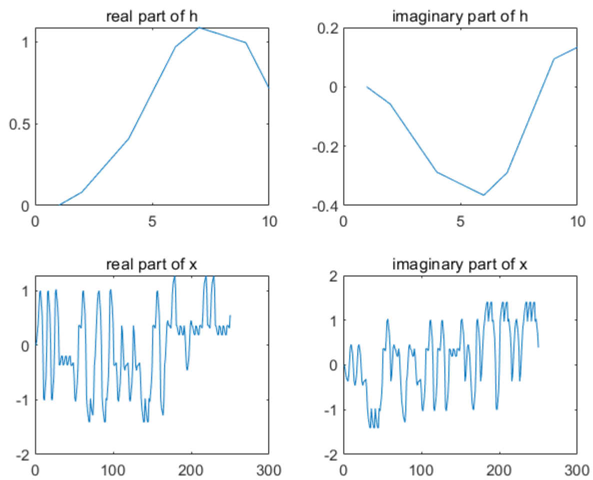


Fig. 8: Plot of the real and imaginary part of **h** and **x**

In this testing phase, we chose $P = 5$, $N = 50$, $L = 2$.the rank of corresponding data matrix is 2. The number of columns of channel response **h** is from $\mathbf{h}(0)$ to $\mathbf{h}(L-1)$. In this testing phase there are only two columns in matrix **H**, according to equations in Figure 9. Hence the rank for **X** is 2.

Another interesting point we found during the experiment was that we tried to change the value of L to verify whether the rank of the matrix **X** would be changed. For L equals 1, 2 and 3, the rank of matrix **X** is the same value as L. However, for any value that is larger than 3, the rank of matrix **X** remains 3. Our preliminary inference is that since the pulse signal length is 2 s, the value $\boldsymbol{\tau}$ we set is 0.1 and 0.6, so the channel response has value in the range of τ_1 to $2 + \tau_2$. For any L-1 is greater than 2, the channel response remains zero which no longer affects the rank of matrix **X**.

To give a better view, we can use equation showed in Figure 9 again. Because $h(t)$ has a limited length, the size of matrix **H** would remain when L becomes larger than a certain value, as we said before. In our case, this **H** matrix would only have 3 non-zero columns when L is greater than 3. Besides, the column of **X** belongs to the $ran(\mathbf{H})$. Thus, the rank stays at 3.

To verify our idea, we tried to increase the present τ value to include more valid channel response value. For example, we increased τ_2 to 1.6. By doing so, the right end of the interval of **h** is $2 + \tau_2 = 3.6$. Hence there is one more column appearing in the column of **H**. The simulation results turned out to be it did increase the rank of matrix **X** to 4.

$$
\mathbf{X} = \mathbf{H}\mathcal{S}_L
$$

$$
= \begin{bmatrix} h(0) & h(1) & \cdots & h(L-1) \\ h\left(\frac{1}{P}\right) & h\left(1+\frac{1}{P}\right) & \cdots & h\left(L-1+\frac{1}{P}\right) \\ \vdots & \vdots & & \vdots \\ h\left(\frac{P-1}{P}\right) & h\left(1+\frac{P-1}{P}\right) & \cdots & h\left(L-1+\frac{P-1}{P}\right) \end{bmatrix} \begin{bmatrix} s_0 & s_1 & \dots & s_{N-1} \\ 0 & s_0 & \dots & s_{N-2} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & s_{N-L} \end{bmatrix}
$$

Fig. 9: Equation

Appendix A
Test for question 1

```
%% test 1
%% the results in this part are about the spatial response for fixed w
% results are in 20/21/23/24 pages of the slide
theta_range = [-0.5*pi:0.001*pi:0.5*pi]';
M1 = 2;
M2 = 3;
M3 = 7;
Delta = 0.5;
w1 = ones(M1,1);
w2 = ones(M2,1);
w3 = ones(M3,1);
w4 = gen_a(M3,Delta,pi/6);
w5 = gen_a(M3,Delta,pi/15);
y1 = spat_response(w1,Delta,theta_range);% page 20
y2 = spat_response(w2,Delta,theta_range);% page 21 left
y3 = spat_response(w3,Delta,theta_range);% page 21 right
y4 = spat_response(w4,Delta,theta_range);% page 24
y5 = spat_response(w3,1,theta_range);%page 23 left-down
y6 = spat_response(w3,2,theta_range);%page 23 right_down
y7 = spat_response(w3+w4,Delta,theta_range);% page 25 left
y8 = spat_response(w3+w5,Delta,theta_range);% page 25 right
x_axis = [-90:0.18:90];
figure(1)
plot(x_axis,abs(y1))
title('Spatial response for fixed w')
xlabel('Degree');
figure(2)
plot(x_axis,abs(y2))
xlabel('Degree');
title('Spatial response for fixed w')
figure(3)
plot(x_axis,abs(y3))
xlabel('Degree');
title('Spatial response for fixed w')
figure(4)
plot(x_axis,abs(y4))
xlabel('Degree');
title('Spatial response for fixed w')
figure(5)
plot(x_axis,abs(y5))
xlabel('Degree');
title('Spatial response for fixed w')
figure(6)
plot(x_axis,abs(y6))
xlabel('Degree');
title('Spatial response for fixed w')
figure(7)
plot(x_axis,abs(y7))
xlabel('Degree');
title('Response for scanning w')
figure(8)
plot(x_axis,abs(y8))
xlabel('Degree');
title('Response for scanning w')
%% generate X, and then do SVD
```

```
% 2 sources, 5antennas, 10 samples
M = 5;
N = 10;
Delta = 0.5;
index = [1:1:M];
theta1 = [-pi/4,pi/4]';
theta2 = [-pi/6,pi/6]';
theta3 = [-pi/144,pi/144]';
%% this is for DOA
SNR = 20;
X1 = gen_data(M,N,Delta,theta1,SNR);
s1 = svd(X1);
X2 = gen_data(M,N,Delta,theta2,SNR);
s2 = svd(X2);
X3 = gen_data(M,N,Delta,theta3,SNR);
s3 = svd(X3);
%% this is for number of samples
SNR = 20;
N1 = 10;
N2 = 100;
N3 = 500;
X1 = gen_data(M,N1,Delta,theta1,SNR);
s1 = svd(X1);
X2 = gen_data(M,N2,Delta,theta1,SNR);
s2 = svd(X2);
X3 = gen_data(M,N3,Delta,theta1,SNR);
s3 = svd(X3);
%% this is for SNR
SNR1 = 30;
X1 = gen_data(M,N,Delta,theta1,SNR1);
s1 = svd(X1);
SNR2 = 20;
X2 = gen_data(M,N,Delta,theta1,SNR2);
s2 = svd(X2);
SNR3 = 10;
X3 = gen_data(M,N,Delta,theta1,SNR3);
s3 = svd(X3);
SNR4 = 0;
X4 = gen_data(M,N,Delta,theta1,SNR4);
s4 = svd(X4);
%

figure(9)
scatter(index,s1,80,'x');
hold on
scatter(index,s2,80,'filled');
hold on
scatter(index,s3,80,'filled','^');
hold on
scatter(index,s4,80,'filled','<');

%% this is for different numbers of antenna
theta = [-pi/6,pi/6]';
M1 = 5;
index1 = [1:1:min(M1,N)];
X1 = gen_data(M1,N,Delta,theta,SNR);
s1 = svd(X1);

```

```
M2 = 10;
index2 = [1:1:min(M2,N)];
X2 = gen_data(M2,N,Delta,theta,SNR);
s2 = svd(X2);

M3 = 15;
index3 = [1:1:min(M3,N)];
X3 = gen_data(M3,N,Delta,theta,SNR);
s3 = svd(X3);
%
figure(10)
scatter(index1,s1,80,'x');
hold on
scatter(index2,s2,80,'filled');
hold on
scatter(index3,s3,80,'filled','^');
hold on
```

APPENDIX B
GEN A FUNCTION

```
function a = gen_a(M,Delta,theta)
% generate the array response a(  ) of a uniform linear array with M elements
% and spacing     wavelengths to a source coming from direction    degrees;
a = zeros(M,1);
for i = 1:M
    a(i)=exp(1j*2*pi*(i-1)*Delta*sin(theta));
end
end
```

APPENDIX C
SPAT RESPONSE FUNCTION

```
function y = spat_response(w,Delta,theta_range)
% plot the spatial response of a given beamformer w as a function of the
% direction    of a source with array response a(  );
[M,~] = size(w);
[b,~] = size(theta_range);
y = [];
for i = 1:b
  a = gen_a(M,Delta,theta_range(i));
  y(i)=w'*a;
end
end
```

APPENDIX D
GEN DATA FUNCTION

```
function X = gen_data(M,N,Delta,theta,SNR)
% generate a data matrix X = AS+N as function of the directions
%    = [   1              d ] T , number of antennas M, number of samples N,
% and signal-to-noise ratio (SNR) in dB (the SNR is  d e ned  as the ratio
% of the source power of a single user over the noise power). S and N
% are respectively a d     N and M    N random zero-mean complex Gaussian matrix;
% transfer SNR into the ratio of power
sigma_square_noise = 0.01;% power of noise
snr_decimal = 10^(SNR/10);% transform dB in to linear
sigma_square_user = sigma_square_noise*snr_decimal;% derive the power of the users'
    signal
%% generate A matrix
```

```
% first generate the columns of A using gen_a(M,Delta,theta(single value)) function
    and then combine them
[d,~]=size(theta);
A = zeros(M,d);
for k = 1:d
   a = gen_a(M,Delta,theta(k));
   A(:,k) = a;
end
%% generate S matrix and N matrix
% S is d*N, Noise is M*N, and each entry of these two matirx is generated
% by a complex Gaussian random variable with sigma square, which is related
% to the power of the signal
% sqrt(var/2)*(randn(1,N)+i*randn(1,N))
S = sqrt(sigma_square_user/2)*(randn(d,N) + (1i * randn(d,N)));
Noise = sqrt(sigma_square_noise/2)*(randn(M,N) + (1i * randn(M,N)));

X = A*S+Noise;
end
```

APPENDIX E
TEST FOR QUESTION 2

```
%% test 2
%% question 2
tau = [0.1,0.6]';
% generate beta vector
ph1 = rand()*2*pi;
ph2 = rand()*2*pi;
beta = [exp(1i*ph1),0.7*exp(1i*ph2)]';
%% generate x and X
P = 5;
N = 50;
L = 2;
h = channel(tau,beta,L,P);
s = source(N);
x = gen_data1(h,s,P,N);
X = zeros(P,N);
for i =1:N
    X(:,i) = x([(i-1)*P+1:i*P]);
end
r = rank(X);
```

APPENDIX F
G FUNCTION

```
function gt = g_t(t)
%
if t<0 || t >2
    gt = 0;
end
if t>=0&& t<=1
    gt=t;
elseif t>1&&t<=2
    gt=2-t;
end
end
```

APPENDIX G
PULSE FUNCTION

```
function g = pulse(tau,L,P)
% here we define a g(t) function to retreive the value of
% g(t) such that we don't have to calculate the value of the
% entries of g(tau)
g = zeros(P*L,1);
tag = 1;
for i = 1:L
    for j = 1:P
        g(tag) = g_t(i-1+(j-1)/P-tau);
        tag = tag + 1;
    end
end
end
```

APPENDIX H
CHANNEL FUNCTION

```
function h = channel(tau,beta,L,P)
% constructchannel response resulting from r paths with delays tau
% and gains beta
% here I assume that we only have one antenna at our receiver.
% Thus we just assign 1 to a(theta) vector.
% we can use g function
[t,~]=size(tau);
one = ones(t,1);
in = zeros(t,L*P);
for i = 1:t
    in(i,:)=pulse(tau(i),L,P)*beta(i);
end
h = in.'*one;
end
```

APPENDIX I
SOURCE FUNCTION

```
function s = source(N)
% construct a source sequence s
s = zeros(N,1);
for i = 1:N
    seed = rand(1);
    if seed>=0&&seed<0.25
        s(i)=1;
    elseif seed>=0.25&&seed<0.5
        s(i)=1i;
    elseif seed>=0.5&&seed<0.75
        s(i)=-1;
    elseif seed>=0.75&&seed<=1
        s(i)=-1i;
    end
end
end
```

APPENDIX J
GEN DATA1 FUNCTION

```
function x = gen_data1(h,s,P,N)
% construct a sampled version of the output x(t)
% which is the convolution between h(t) and s-delta(t)
% first we extend s to s_ext
s_est = kron(s,[1;zeros(P-1,1)]);
```

```
% then do conv
x = conv(h,s_est);
% pick the first N*P samples up
x = x([1:N*P]);
end
```