MUSIC Algorithm Project

Dept. of Electrical engineering, Shiraz University

Instructor: Dr. Alireza Masnadi-Shirazi

Introduction

MUSIC stands for MUltiple SIgnal Classification. It is a subspace-based algorithm used for estimating the direction of arrivals (DOA) of same-frequency narrowband sources arriving to a sensor array. For simplicity we assume that the sensors form an array and are positioned in a linear fashion, although more complicated array structures is possible. For additional reading beyond what we have taught in class, you can go to the two references that comes with this assignment.

General Assignment and Matlab Walkthrough

In this project you will do the following.

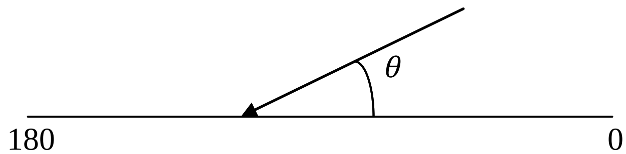
Step a: Simulate the narrowband sources

Lets assume in general there are *M*sensors and *N* sources and *p* being the number of time snapshots. First you will construct the narrowband complex exponential source signals. It is assumed that the frequency of all sources is 10^6 Hz and the sampling frequency at the sensors is 10^7 Hz. Lets for now assume each source has a random amplitude which is Gaussian distributed with mean zero and variance 1. Also assume that all sources are independent of one another. Lets start with M=10 and N=5 and p=100. In Matlab these *N* sources along with their *p* snapshots can be constructed in matrix form as following.

```
p=100;% number of Snapshots
fs=10^7; %sampling Frequency
fc=10^6 ;%center Frequency of narrow band sources
M=10;%Number of array elements
N=5; %Number of sources
s_var=1; %variance of the amplitude of the sources
s=sqrt(svar)*randn(N,p).*exp(j*(2*pi*fc*repmat([1:p]/fs,N,1)));% p
snapshots of N narrowband sources with random amplitude of mean zero
and covariance I
```

Step b: Mix the sources using the Steering Matrix and get the sensor signals

Assume that the sensors are antennas and the sources travel with the speed of light. The Linear array of antennas are each spaced 150 meters apart (you can experiment later that with increased frequency of the sources you can afford the decrease the antenna spacing). We assume that incident DOAs of the sources are `20,50,85,110,145` where each angle is with respect to the reference shown below where the horizontal line is the line were are antennas are linearly placed.



Now we can construct what is received at the antennas by multiplying the source signals by the steering matrix. Also for now we assume that the variance of the noise is 1.

```
doa=[20;50;85;110;145]; % DOAs
c_speed=3*10^8 ; % speed of light
dist=150;  %antenna spacing
A=zeros(M,N); %To create a matrix with M row and N column
for k=1:N
     A(:,k)=exp(-j*2*pi*fc*dist*cosd(doa(k))*(1/c_speed)*[0:M-1]'); %NOTE: 
cosd(.) is in angles and cos(.) is in radians!
end
noisecoeff=1;  %variance of added noise
x=A*s+sqrt(noisecoeff)*randn(M,p);
```

Step c: Estimate the covariance matrix of the sensor array

So far from the code above we have 100 data snapshots of the 10 arrays stored as 10x100 matrix in "x". It is expected that the columns of x have a mean of zero since it was originally reconstructed with random variable of mean zeros. One way to estimate the covariance is through the summation $R_x = \frac{1}{100}\sum_{i=1}^{100} x^{(i)} {x^{(i)}}^H$ where $x^{(i)}$ is each snapshot data received at the antennas and i is the snapshot index. In other words in our code $x^{(i)}{'s}$ are the columns of the matrix x. instead of using the summation we can find R_x in one step using the matrix x as following

```
R=(x*x')/p %Empirical covariance of the antenna data
```

Step d: <CODE NOT PROVIDED> Find the noise subspace and estimate the DOAs

So we now have the data observed at the antennas concentrated in matrix R. The next step which the code is not provided, is for you to perform the MUSIC algorithm in order to figure out the DOAs. In summary here is what you need to do:

1- It is assumed that you know the number of sources, the array geometry and the center frequency of the narrowband sources, but you do not know the DOAs and they need to be estimated. Use the function svd or eig in matlab to find the noise subspace U_2 of R. The noise subspace is the subspace that corresponds to the M-N smallest eigen values of R.

2- Do a sweep on θ from 0 to 180 degrees with steps of 1 degree. For every θ construct a steering vector as following $a(\theta) = [1\ e^{-j2\pi f_c d\cos(\theta)/c}\ \ldots\ e^{-j2\pi f_c (M-1) d\cos(\theta)/c}\ ]^T$. In the case where there is no added noise, the inner product of $a(\theta)$ and the noise subspace is zero for the correct DOAs. Now for the case where noise is present, the norm of the inner product of the $M \times 1$ steering vector $a(\theta)$ and the $M \times (M-N)$ noise subspace Matrix U_2 should be minimum for the correct N DOAs. We can reformulate it into a maximum finding approach as follows

$$1 - do\ a\ sweep\ on\ \theta$$

$$2 - find\ \theta's\ corresponding\ to\ N\ largest\ peaks\ of\ \ \frac{1}{||a(\theta)^H U_2||^2}$$

Step e: Repeat the experiment for different parameters

Repeat the experiment for different parameter sets as following

1- Repeat the experiment for noise variances being 1, 5, 10, 20, 35, 50 and answer the following.
   a) How does noise effect the performance of the algorithm?
   b) for low noises How can the number of Sources be estimated ? Hint: look at the eigen values.

2- For noise variance being 5, repeat the experiment for inter-spacing between antennas of 50 100 150 200 250 meters. What do you observe?

3- For noise variance being 5, repeat the experiment for the variance of the amplitude of sources being 0.01, 0.1, 1, 5, 10. What do you observe?

What to turn in

All code should be submitted as one single matlab file. Avoid using inter linked functions. Your report should be typed up (in farsi or english) and in .doc or .pdf format. Everything should be on a CD and project number written on the CD.

DO IT YOURSELF: All Codes and reports will be cross checked, any similar code or report will be given a grade of zero for both sides :(