EE627A

Project Report

Multiple Source Localisation In The Spherical Harmonic Domain

Authors:Christine Evers, Alastair H. Moore and Patrick A. Naylor

April 14, 2019

Department of Electrical Engineering
Indian Institute of Technology(Kanpur)
India

Submitted by:
Kritik Soman (18104052)
Lisha Kumari(18104055)
Nidhi Sen(18104066)
Ravikiran (18104089)
Meeniga Varun (18104413)

Contents

List of Figures

1 Abstract

In this paper, the objective is to evaluate the performance of a low-complexity localisation approach using spherical harmonics in reverberant environments for multiple speakers. As source localisation becomes a challenging problem in reverberant environments and in noisy conditions, Eigen beams of zero and first order are used to estimate the pseudo-intensity vectors pointing in the direction of sound intensity.This paper proposes a clustering approach in which the intensity vectors of active sound sources and strong reflections are extracted, yielding an estimate of the source direction in **azimuth** and **inclination**.

2 Introduction

Source Localisation is prerequisite for many acoustic systems such as scene analysis ,room geometry inference etc.To provide accurate sound source localization and mapping ,accurate data in 3 dimensions are required.Spherical microphone is used in experiment to capture high resolution of data in the spherical harmonic domain(complete set of orthogonal functions used to represent function defined on the surface of the sphere.) as it facilitate analysis and processing of sound with high resolution.Eigen beams which are spherical Fourier Transform of the field is used to compute pseudo-intensity vectors.To find the dominant direction in the set of vectors K-means clustering is employed.Three setup are used in the experiment .In setup 1 two sources with the separation of 45deg are taken for multiple reverberation time. In setup 2 ,2 sources are localized with the separation of {5,10,15....180} with the reverberation time of 0.4 sec. In setup 3 experiment is extended upto 5 simultaneous sound sources.Multiple setup in the experiment shows that clustering approach is independent of separation between the sources.Respective results and their descriptions are included in the below sections.

3 Problem Definition

In this paper, we address the problem of DOA estimation with spherical microphone arrays.This problem is to estimate the azimuth and elevation but not range.

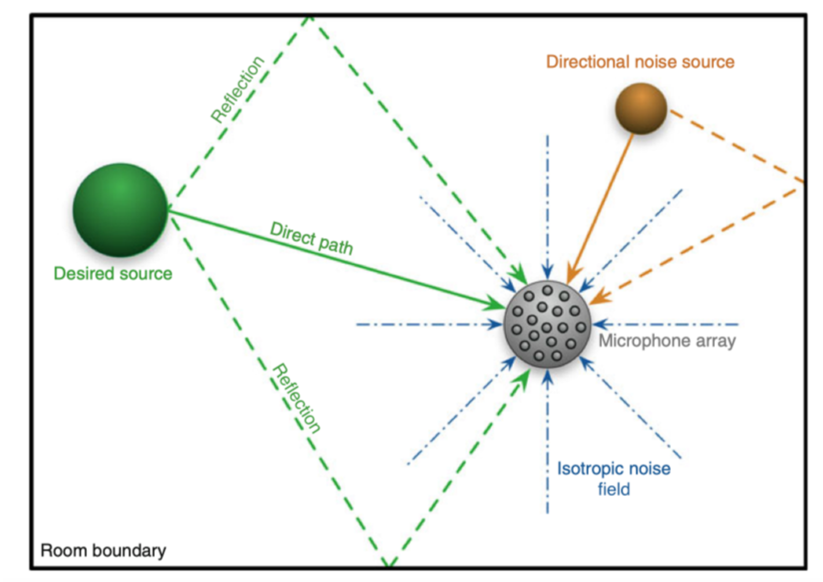


Figure 1: Schematic illustration of scenario in microphone array signal processing

4 Proposed Methodology

4.1 System Model

The eigenbeams, $P_{lm}(k)$, of order l and degree m, measured at a spherical microphone array can be modelled as:

$$P_{lm}(k) = \sum_{n=1}^{N_s+N_{ref}} X_{lm}(k, \Omega_n)S_n(k) + V_{lm}(k)$$

where k = ω/c is the wavenumber, N_s is the number of audio sources,N_{ref} is the number of reflections, X_{lm} is the sound intensity due to a unit amplitude plane wave arriving from angle $\{\Omega_n\}_{n=1}^{N}$; $S_n(k)$ is the amplitude of the nth plane wave; and $V_{lm}(k)$ models sensor noise and late reverberation

4.2 Pseudo-intensity vectors

The sound power per unit area is described by the acoustic intensity vector, I, as a function of sound pressure and particle velocity. Sound intensity consists of an active and a reactive part relating to the stored parts and flow respectively of signal energy.Sound is described by the active sound intensity,expressed as:

$$I = \frac{1}{2}Re\{p(x,\omega)v^*(x,\omega)\}$$

where,$p(x,\omega)$ is the sound pressure;$v(x,\omega)$ is the particle velocity of the sound field at point, x, and angular frequency, $\omega = 2\pi f$; and Re{} is the real part of the argument.
Pseudo-intensity vectors are conceptually similar to the active sound intensity but are calculated from the zero and first-order eigen-beams $P_{lm}(k)$.The pseudo-intensity vector,I(k), is thus defined as

$$I(k) = \frac{1}{2}Re\{\tilde{P}_{00}(k)^*d(k)\}$$

where the omni-directional pressure at the centre of the array,$\tilde{P}_{00}(k)$ is approximated as

$$\tilde{P}_{00}(k) = \left(\frac{P_{00}(k)}{b_0(k)}\right)$$

where $b_0^{-1}(k)$ compensates for the 0-order mode strength, which is a function of the array geometry.

5 Steps of Execution

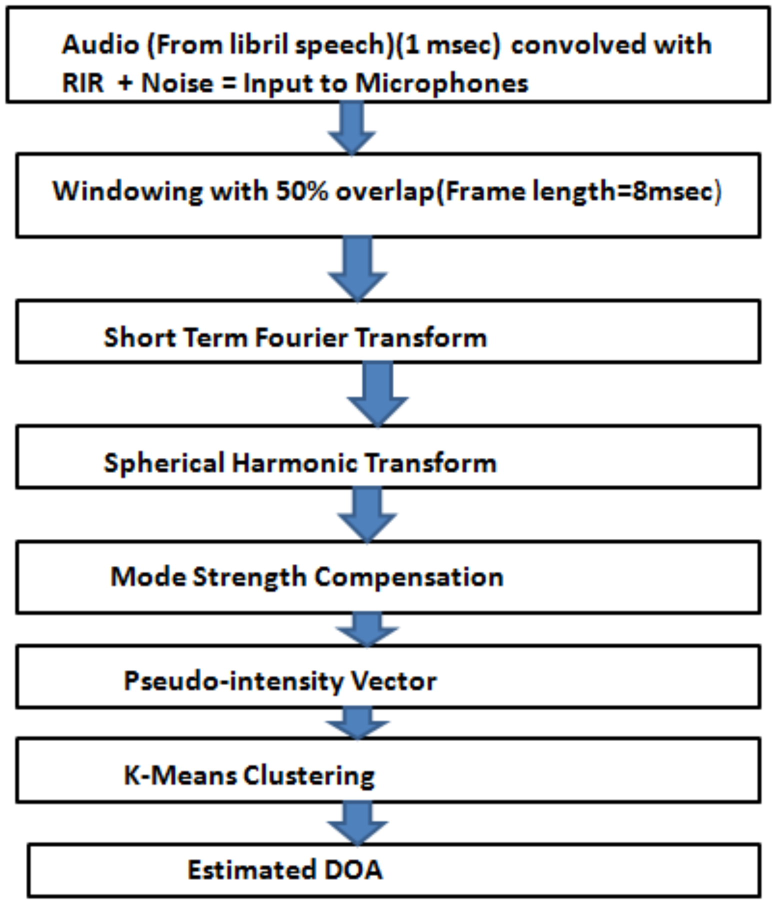


Figure 2: Block Diagram of Steps

Experiment 1: The effect of reverberation time is considered for two sources with a [U+FB01]xed angular separation of 45 deg. The angular error obtained for 2 sources with 45 deg spacing for reverberation time $= 0.2, 0.4, 0.6$ s is computed.
For every value of reverberation time:

1. Two audio files are randomly chosen from the database and placed at 45 degree separation.
2. RMSE is found for the estimated DoA.
3. Above 2 steps is repeated 100 times and then box plot is obtained for the particular reverberation time.

Experiment 2: The effect of the angular separation of two sources is investigated with T 60 = 0.4 s. The boxplots of the angular error obtained for angular spacings of s = 5, 10, 15, 30, 45, 90, 135, 180deg for 2 sources with T 60 = 0.4 s is computed.
For every value of angular separation:

1. Two audio files are randomly chosen from the database and placed at the specific angular separation.
2. RMSE is found for the estimated DoA.

3. Above 2 steps is repeated 100 times and then box plot is obtained for the particular angular separation.

Experiment 3: The results of experiment 2 are extended to include up to 5 simultaneous sound sources. The mean angular error obtained for N s = 2, 3, 4, 5 with the same values of the angular separation as in experiment 2. It can be seen that, increasing the number of sources tends to increase angular error. The lines are truncated as the maximum spacing decreases with increasing number of sources.

1. For every value of angular separation:
   (a) n files are randomly chosen from the database and placed at the specific angular separation.
   (b) RMSE is found for the estimated DoA.
   (c) Above 2 steps is repeated 100 times and then their mean is plotted for the specific angular separation and number of source.
2. The above step is repeated for number of source from 2 to 5.

6 Scripts and functions used in simulation

File Name	Use
run_all.m	Script will generate input for spherical microphone array, find pseudo-intensity vector and show plot of PIV for a particular source location.
run_allexpt1.m	Script will perform experiment 1 as described in paper. The output is the figure 4 as shown in the paper.
run_allexpt2.m	Script will perform experiment 2 as described in paper. The output is the figure 5 as shown in the paper.
run_allexpt3.m	Script will perform experiment 3 as described in paper. The output is the figure 6 as shown in the paper.
est_doa.m	Function will find estimate of DoA using k-means and hungarian assignment on the PIV.
est_doa_2.m	Function will find estimate of DoA using only k-means clustering on the PIV.
gen_db.m	Function will take source location, audio file name and reverberation time as input and generate the input for the spherical microphone array by convolving with room impulse response and adding 10dB noise.
gen_piv.m	Function will take microphone input audio, compute STFT,SHT, perform mode strength compensation, find PIV and return it's histogram.
modeStrength.m	Function that computes the mode strength for the given spherical microphone array.
piv_hist.m	Function that generates the histogram of PIV by binning them into azimuth and inclination bins.
piv.m	Function that returns the PIV when given the compensated eigenbeams of order 0 an 1
sht.m	Function that returns the spherical harmonic transform when given the STFT of microphone signals
stft.m	Function that takes 2D multichannel signal matrix and returns 3D short time fourier transform (STFT).
activlev.m	Function to measure active speech level as in ITU-T P.56
rfft.m	Function that calculates the DFT of real data Y=(X,N,D).
smir_generator.m	Function that gives the room impulse response for the spherical microphone array given the room dim, source pos.,etc.
sphBasis.m	Function that matrix of spherical harmonics evaluated at (inc, az).
distcos.m	Function for finding cosine similarity.

7 Experimental Set-Up 1

Speakers(N_s)	2
Spherical Microphone Array	32
Spherical Microphone Array Radius	4.2cm
Spherical Microphone Array Position	(2.54,2.55,4.48 m)
Room Dimensions	5 × 4 × 6 m
Clusters(C)	2
Actual position of Speakers(Ω_q)	Speaker 1(5,90)deg Speaker 2 (50,90)deg
Speech length	1 sec
Sampling rate	8 Khz
Reverberation time	0.4s
Other parameters	As per paper(given in the code)

7.1 Result of Experiment Set Up 1

The clusters are formed at the DOAs of source and strong power field around source localization.

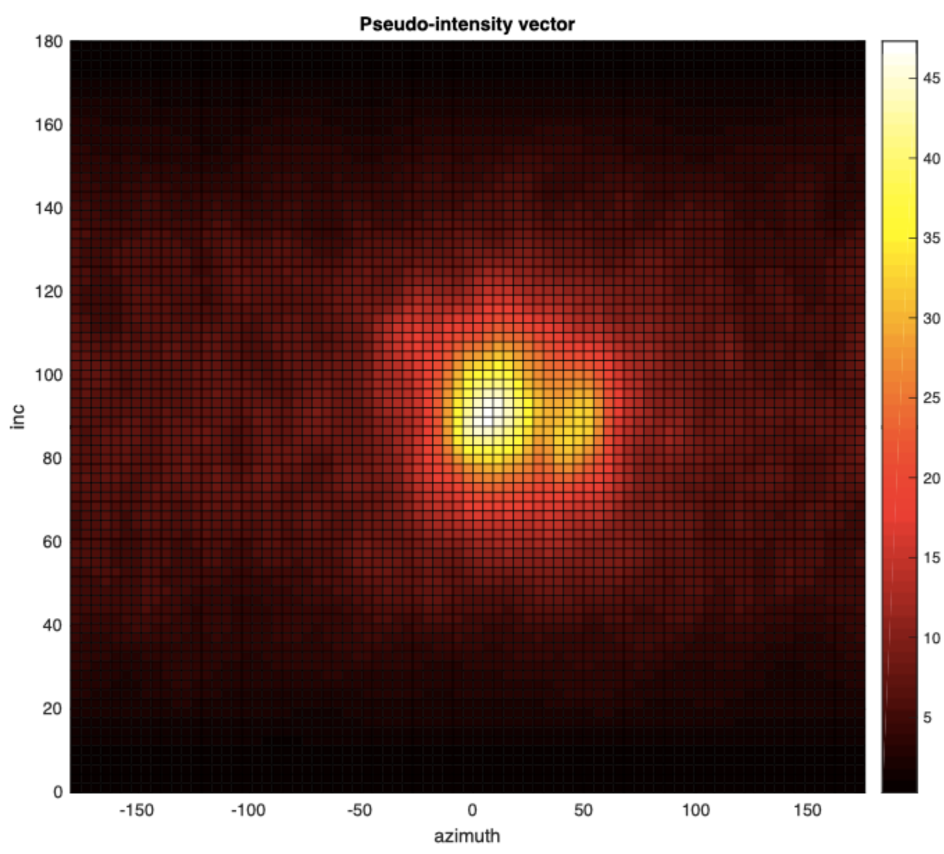


Figure 3: Intensity map of pseudo-intensity vectors

7.2 Result 2 of Experiment Set Up 1

The 2-D plot of intensity vector histogram shows that two peaks can be identified around the source location.

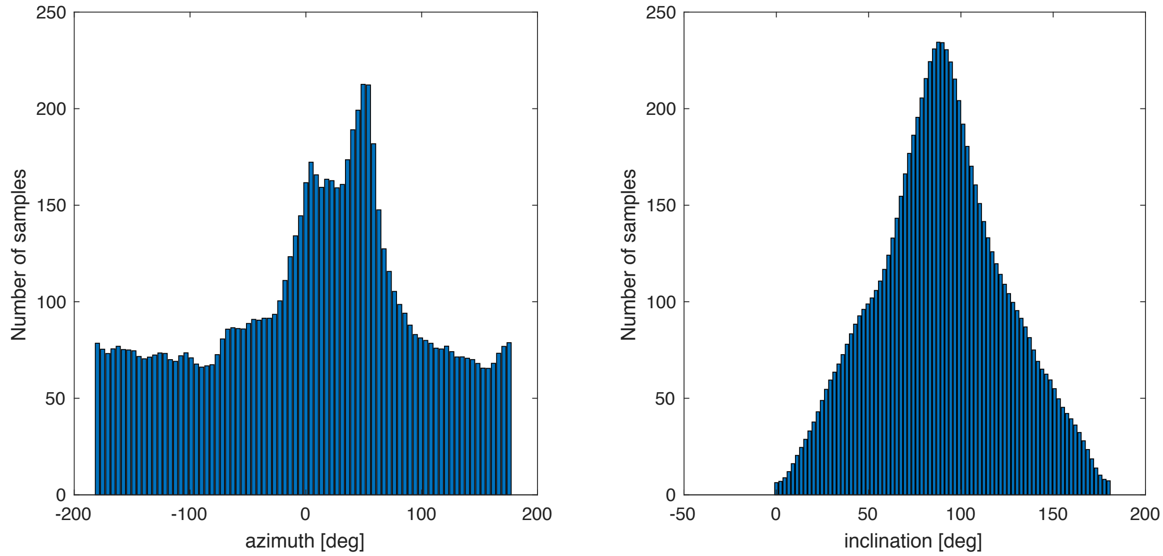


Figure 4: 2D histogram of intensity vectors in azimuth and inclination.

7.3 Result 3 of Experiment Set Up 1

This figure shows the plot of the true target positions and their DOA estimates.

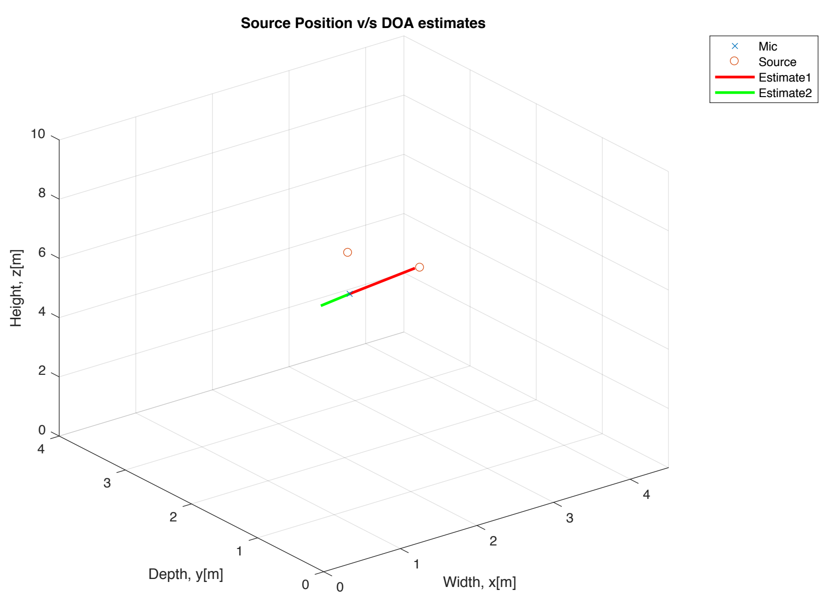


Figure 5: Source Positions vs DOA Estimate

7.4 Result 4 of Experiment Set Up 1

In this,effect of T_{60} is considered for 2 sources with a fixed angular separation of 45deg.This Figure shows boxplots of the angular error obtained for two sources for $T_{60} = \{0.2, 0.4, 0.6\}sec$.It can be seen that the median error with T_{60}.

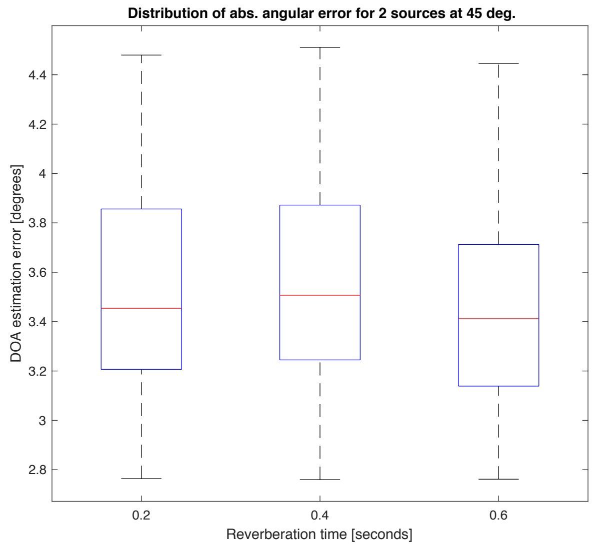


Figure 6: Distribution of absolute angular error vs T_{60} for two sources 45deg apart

8 Experimental Set-Up 2

Speakers(N_s)	2
Spherical Microphone Array	32
Room Dimensions	$5 \times 4 \times 6$ m
Clusters(C)	2
Actual position of Speakers(Ω_q)	Speaker 1:(rand,90)deg, Speaker 2:(rand+$\delta\phi_s$,90)deg
Speech length	1 sec
Sampling rate	8 Khz
Reverberation time	0.4s
Angular Spacings($\Delta\phi_s$)	(5,10,15,30,45,90,135,180)
Other parameters	As per paper(given in the code)

8.1 Result of Experimental Set-Up 2

This graph is plotted by varying angular spacing for 2 sources with reverberation time as 0.4s.The median error is approximately same in all cases, shows that this approach is independent of angular spacings.

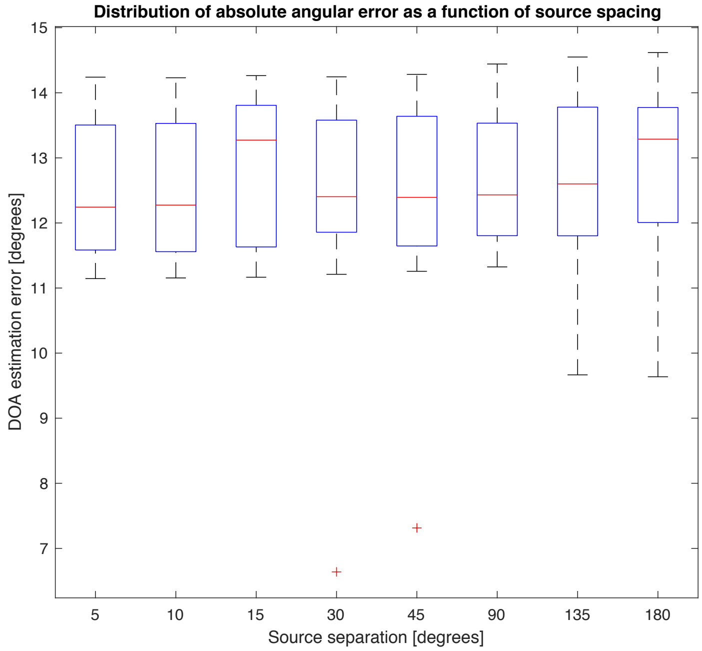


Figure 7: Distribution of absolute angular error as a function of source spacing for two sources

9 Experimental Set-Up 3

Speakers(N_s)	2-5
Spherical Microphone Array	32
Room Dimensions	5 × 4 × 6 m
Clusters(C)	2-5
Actual position of Speakers(Ω_q)	Speaker 1:(rand,90)deg, Speaker 2:(rand+$\delta\phi_s$,90)deg, Speaker 3:(rand+2$\delta\phi_s$,90)deg, Speaker 4:(rand+3$\delta\phi_s$,90)deg, Speaker 5:(rand+4$\delta\phi_s$,90)deg
Reverberation time	0.4s
Angular Spacings($\delta\phi_s$)	(5,10,15,30,45,90,135,180)
Other parameters	As per paper(given in the code and above experiments)

9.1 Result of Experiment 3

The result is extended to 5 sound sources.It can be seen that increasing number of sources tends to increase angular error and also lines are truncated as the maximum spacing decrease with increasing number of sources.As the spacing widens to about 45deg the error tends to that obtained with 2 sources and clustering approach is independent of $\Delta\phi_s$ for two sources.

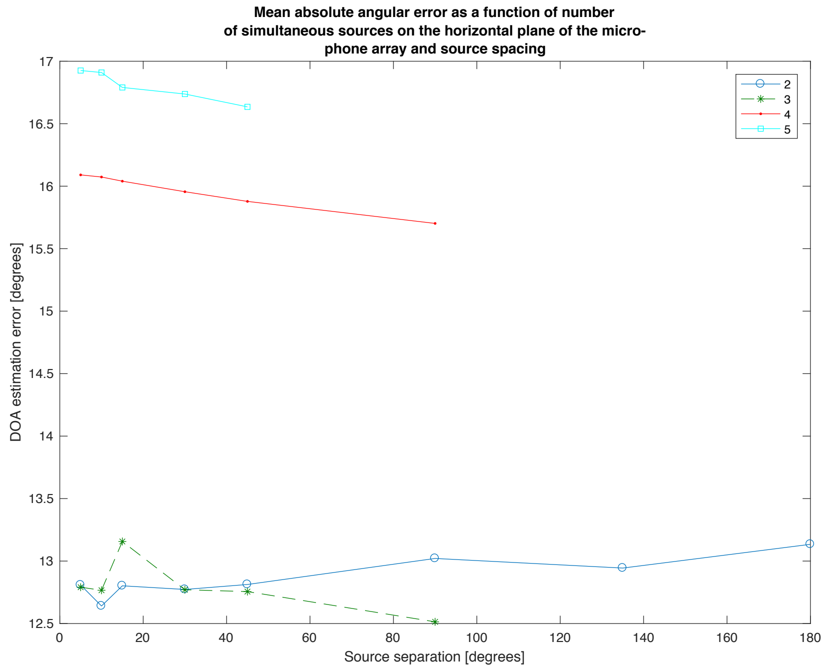


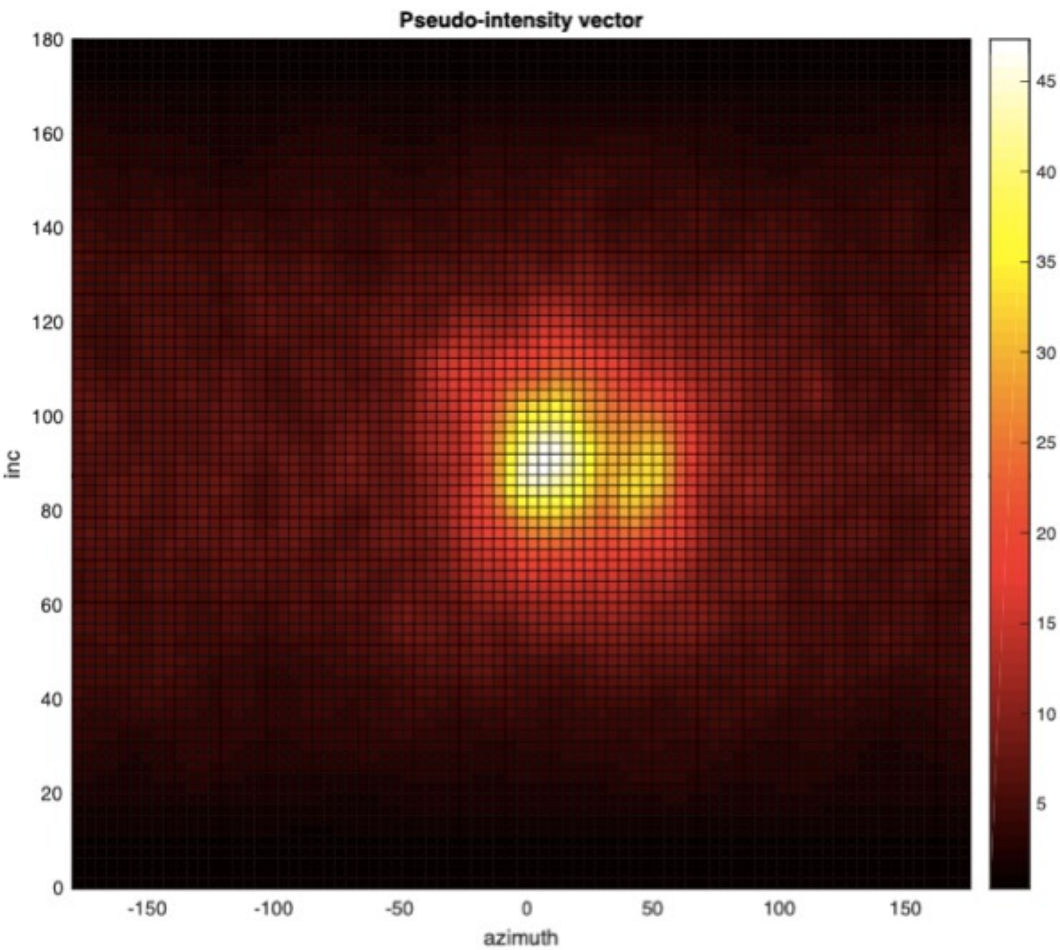
Figure 8: Mean absolute angular error as a function of number of simultaneous sources

10 Discussions and Conclusion

Above demonstrations, proposed an approach for localization of multiple sound sources using spherical arrays in reverberant environments.Pseudo-intensity vectors were estimated and clustered to extract the DOAs of the direct path sound waves and strong reflections.Multiple setup in the experiment shows that clustering approach is independent of separation between the sources.It is advantageous to use pseudo-intensity vectors in terms of computational requirements and it offers fast source localization.

```
clear all;
close all;
clc;
addpath('utils');
plotting=0;
% rng default;
omega=[5,90;50,90];
source_file_path = fullfile('flac',
{'84-121123-0000.flac','84-121123-0025.flac'});
tic
nclusters=length(omega);
gen_db(omega,source_file_path,0.2);%generates input for the mics given
 position of source
gen_piv(plotting);%generates PIV from mic input
% errMat=est_doa(plotting);%using hungarian assignment
errMat=est_doa_2(plotting,nclusters);%using just k-means
% disp(rms(errMat));
toc
```

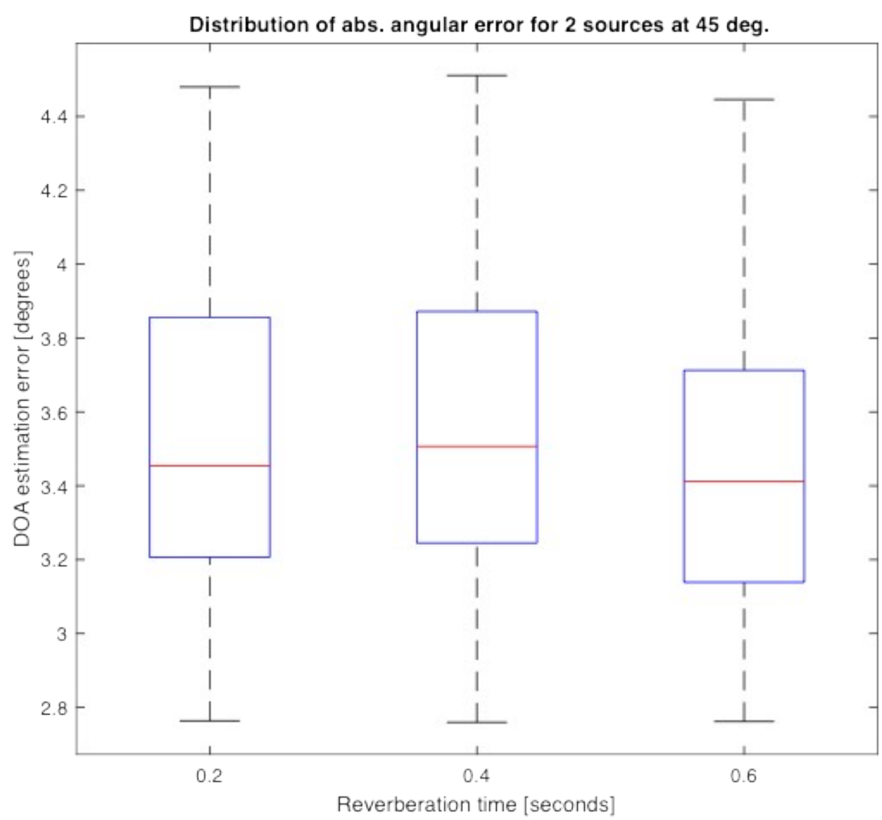
Elapsed time is 18.041823 seconds.



Published with MATLAB® R2018b

```
clc; clear all; close all;
addpath('utils');
%% (Fig.4) Script to generate Distribution of absolute angular error as a
function of T60 for two sources 45,ó¶ apart on the horizontal plane of the
microphone array.

error_04=zeros(100,3);
rand_flac1=randi([3,2679],1,100);
rand_flac2=randi([3,2679],1,100);
rand_omega=-pi+ (2*pi)*rand(1,100);
files=dir('flac');
plotting=0;
beta_array=[0.2,0.4,0.6];
for j=1:length(beta_array)
    for i=1:100
        omega=[rand_omega(i),90;rand_omega(i)+pi/4,90];
        source_file_path =
fullfile('flac',{files(rand_flac1(i)).name,files(rand_flac2(i)).name});
        nclusters=length(omega);
        gen_db(omega,source_file_path,beta_array(j));%generates input for the
mics given position of source
        gen_piv(plotting);%generates PIV from mic input
        errMat=est_doa(plotting);%using hungarian assignment
%         errMat=est_doa_2(plotting,nclusters);%using just k-means
        error_04(i,j)=rms(errMat);
    end
end
boxplot(error_04,'Labels',{'0.2','0.4','0.6'});
xlabel('Reverberation time [seconds]');
ylabel('DOA estimation error [degrees]');
title(' Distribution of abs. angular error for 2 sources at 45 deg.');
```

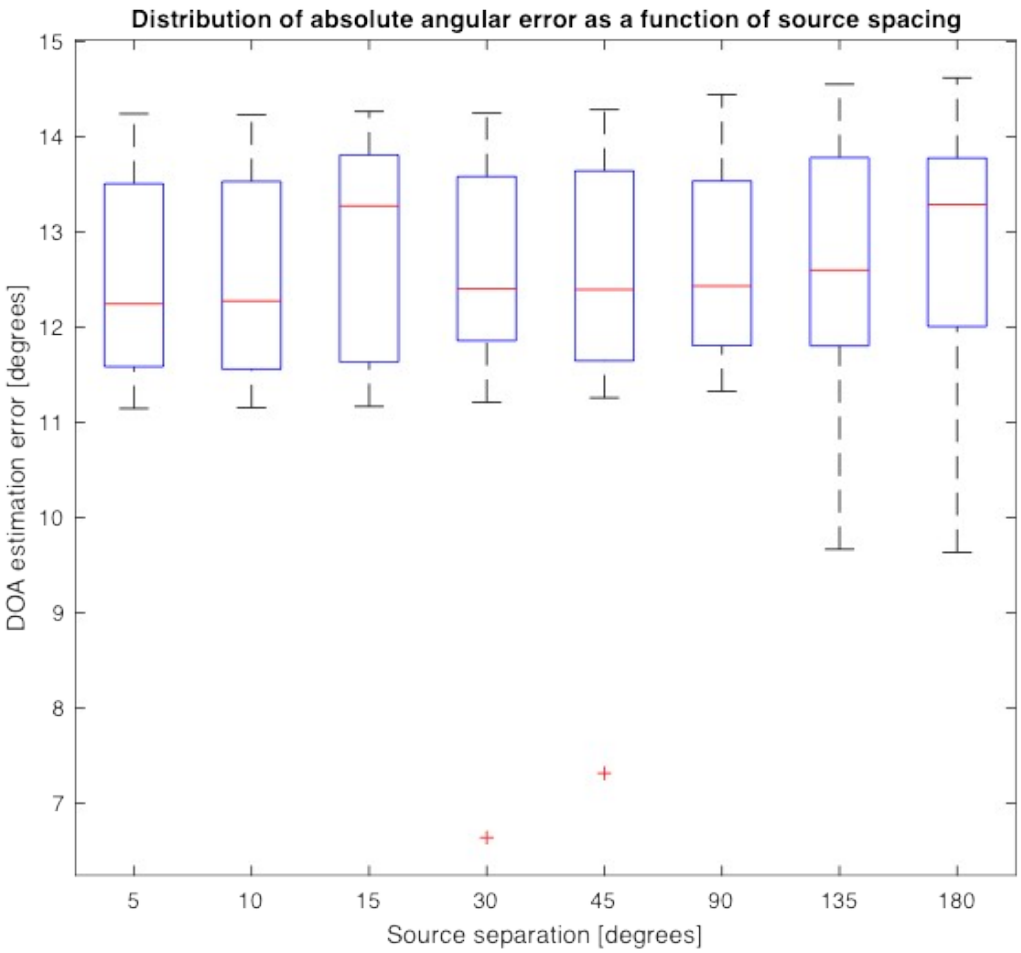


Published with MATLAB® R2018b

```
clear all;clc;close all
addpath('utils');
%% (Fig.5) Script to generate : Distribution of absolute angular error as a
function of source spacing for two sources on the horizontal plane of the
microphone array (T60=0.4 s).
plotting=0;
rand_flac1=randi([3,2679],1,100);
rand_flac2=randi([3,2679],1,100);
rand_omega=-pi+ (2*pi)*rand(1,100);
files=dir('flac');
saperation=[5 10 15 30 45 90 135 180];
error=zeros(100,numel(saperation));
beta=0.4;
tic
for j=1:numel(saperation)
    for i=1:100
        omega=[rand_omega(i),90;(rand_omega(i))+saperation(j)*pi/180,90];
        source_file_path =
fullfile('flac',{files(rand_flac1(i)).name,files(rand_flac2(i)).name});

        nclusters=length(omega);
        gen_db(omega,source_file_path,beta);%generates input for the mics
given position of source
        gen_piv(plotting);%generates PIV from mic input
%         errMat=est_doa(plotting);%using hungarian assignment
        errMat=est_doa_2(plotting,nclusters);%using just k-means
        error(i,j)=rms(errMat);
    end
end
toc
% boxplot(error)

boxplot(error,'Labels',{'5','10','15','30','45','90','135','180'});
xlabel('Source separation [degrees]');
ylabel('DOA estimation error [degrees]');
title('Distribution of absolute angular error as a function of source
spacing');
```



Published with MATLAB® R2018b

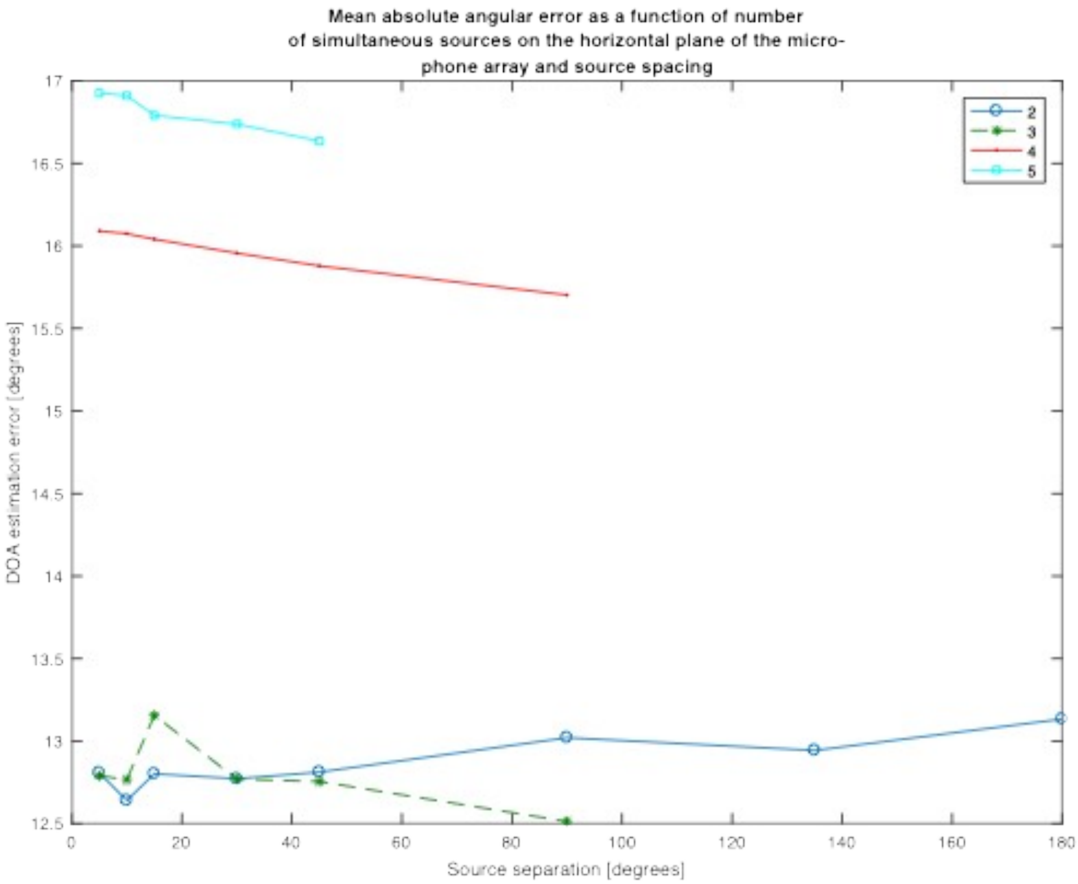
```
clear all;clc;close all
addpath('utils');
%% (Fig.6) Script to generate Mean absolute angular error as a function of
number of simultaneous sources on the horizontal plane of the microphone
array and source spacing (T60=0.4 s).
rand_flac1=randi([3,2679],1,100);rand_flac2=randi([3,2679],1,100);rand_flac3=
randi([3,2679],1,100);
rand_flac4=randi([3,2679],1,100);rand_flac5=randi([3,2679],1,100);
rand_omega=-pi+ (2*pi)*rand(1,100);
files=dir('flac');
plotting=0;
beta=0.4;
separation=[5 10 15 30 45 90 135 180];
tic
errPlots={};
for k=2:5
    if k==2 pos=8;
    elseif k==3 pos=6;
    elseif k==4 pos=6;
    elseif k==5 pos=5;
    end
    error=zeros(100,numel(separation(1:pos)));
    for j=1:numel(separation(1:pos))
        for i=1:100

sources=fullfile('flac',{files(rand_flac1(i)).name,files(rand_flac2(i)).name,
files(rand_flac3(i)).name,files(rand_flac4(i)).name,files(rand_flac5(i)).name
});

angles=[rand_omega(i),90;(rand_omega(i))+separation(j)*pi/180,90;(rand_omega(
i))+2*separation(j)*pi/180,90;(rand_omega(i))+3*separation(j)*pi/180,90;(rand
_omega(i))+4*separation(j)*pi/180,90];
            for l=1:k
                source_file_path(l) = sources(l);
                omega(l,1:2)=angles(l,1:2);
            end

            nclusters=length(omega);
            gen_db(omega,source_file_path,beta);%generates input for the mics
given position of source
            gen_piv(plotting);%generates PIV from mic input
            % errMat=est_doa(plotting);%using hungarian assignment
            errMat=est_doa_2(plotting,nclusters);%using just k-means
            error(i,j)=rms(errMat);
        end
    end
    mean_error=mean(error,1);
    errPlots(end+1)={mean_error};
    plot(mean_error);hold on;
end
toc

%% Plotting
figure;
for k=2:5
    if k==2 pos=8;
    elseif k==3 pos=6;
    elseif k==4 pos=6;
    elseif k==5 pos=5;
    end
    plot(separation(1:pos),errPlots{k-1});hold on;
end
```



Published with MATLAB® R2018b