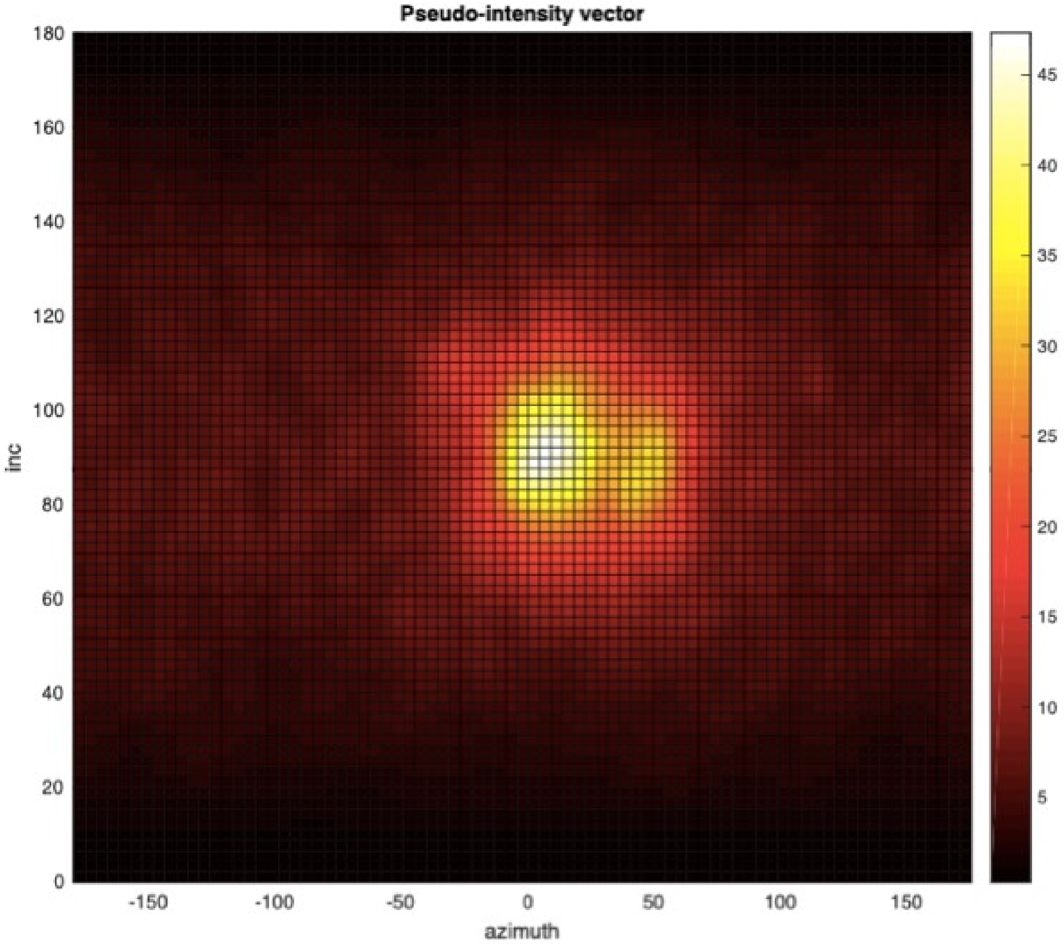
```
clear all;
close all;
clc;
addpath('utils');
plotting=0;
% rng default;
omega=[5,90;50,90];
source_file_path = fullfile('flac',
{'84-121123-0000.flac','84-121123-0025.flac'});
tic
nclusters=length(omega);
gen_db(omega,source_file_path,0.2);%generates input for the mics given
 position of source
gen_piv(plotting);%generates 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
```

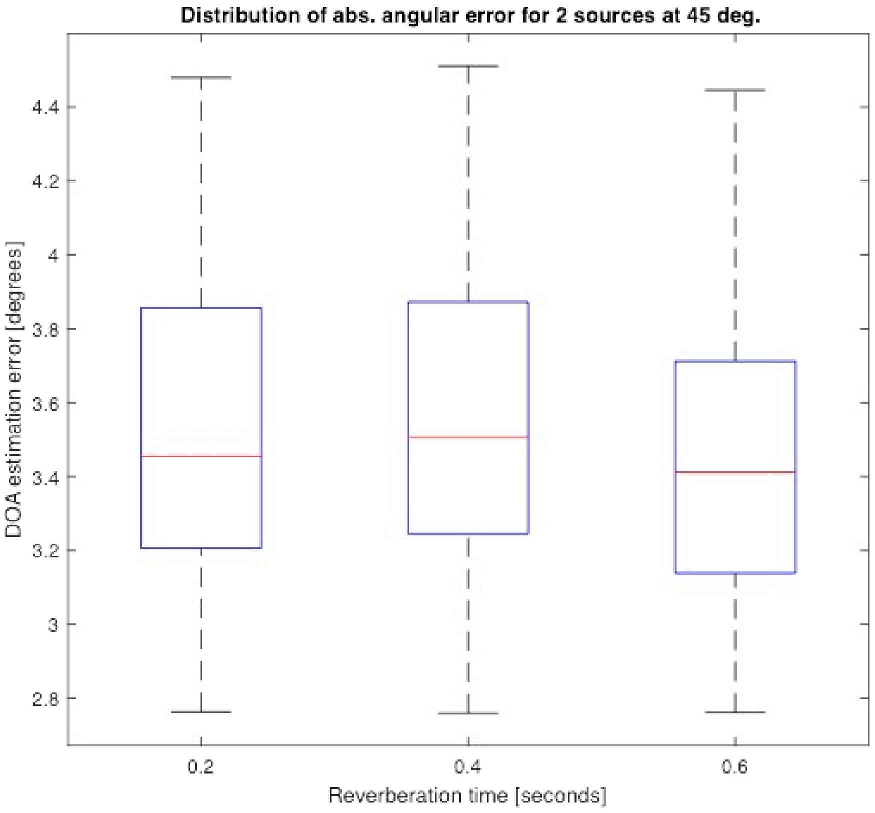
```
Elapsed time is 18.041823 seconds.
```



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.');
```

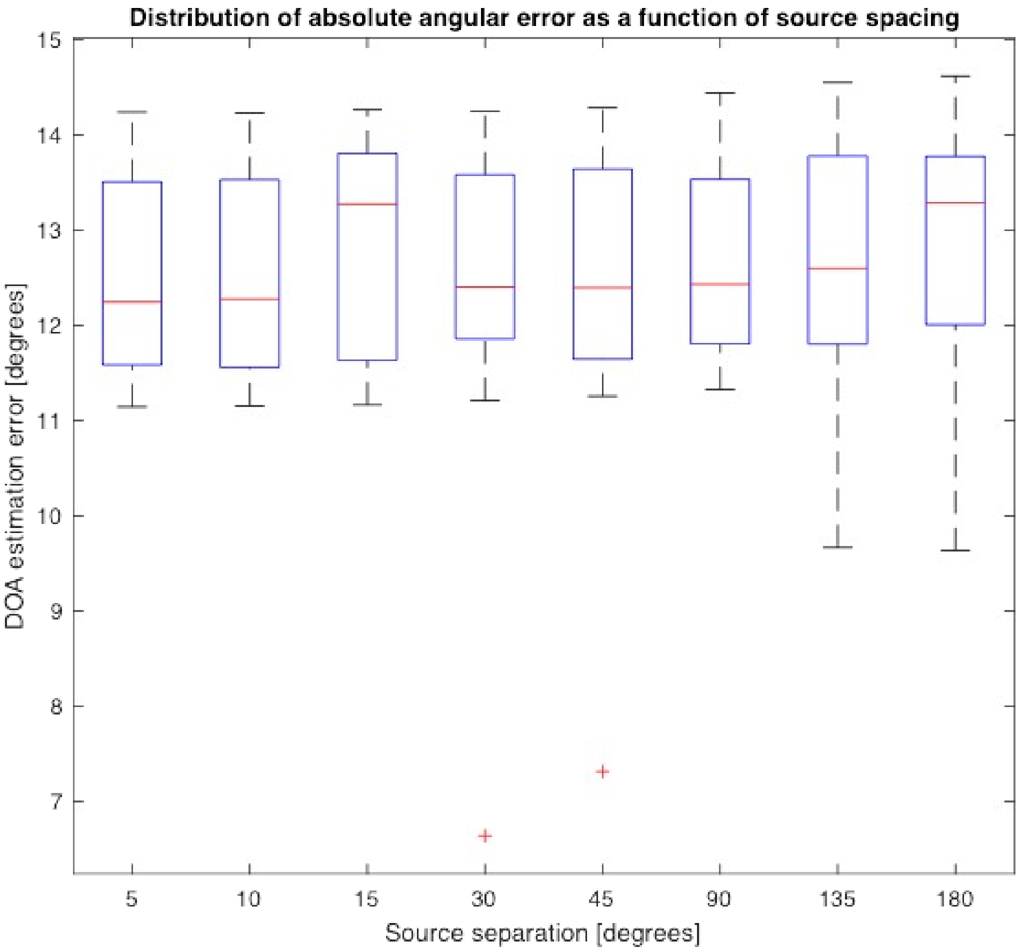


Published with MATLAB® R2018b

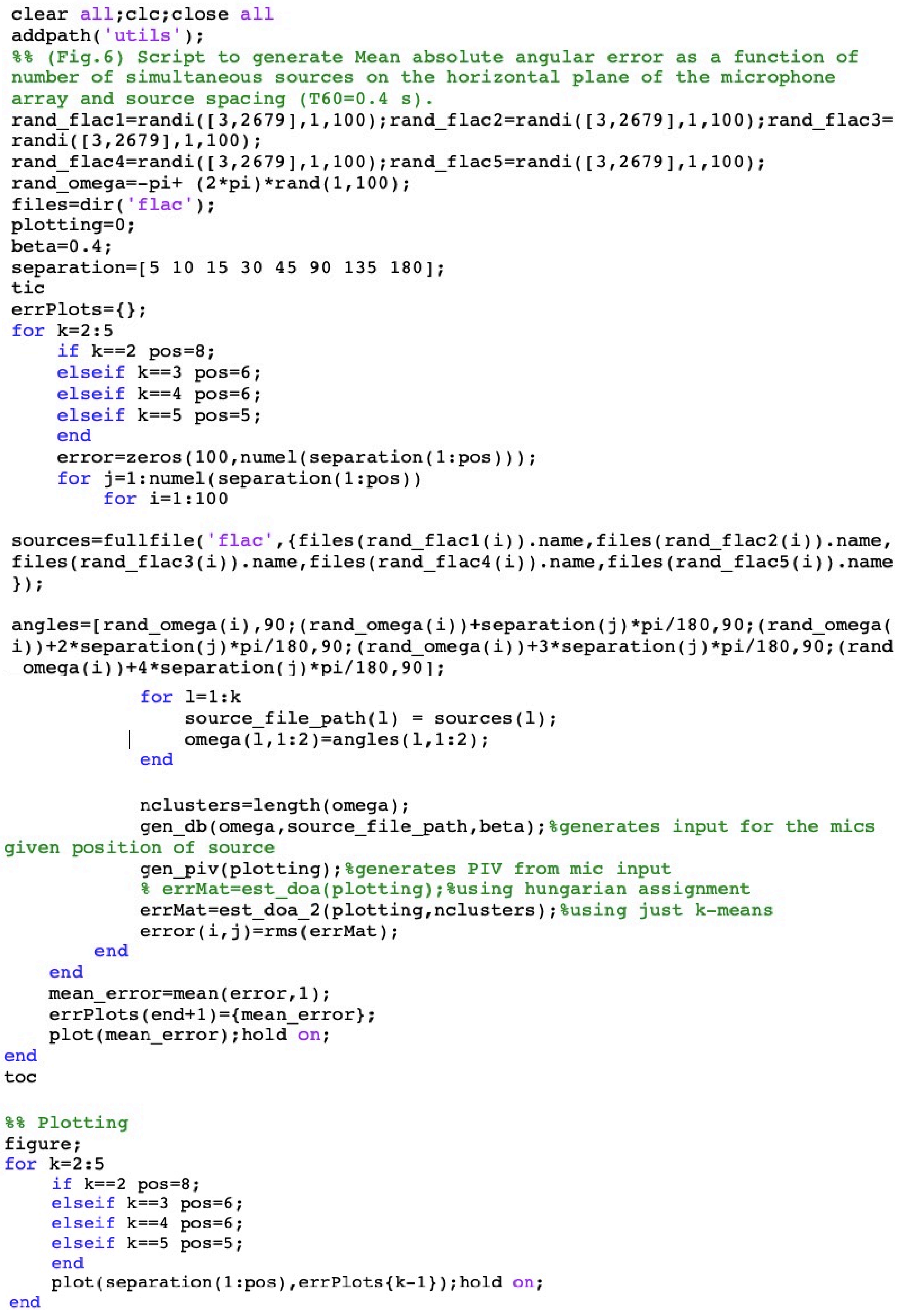
```
clear all;clc;close all
addpath('utils');
%% (Fig.5) Script to generate : Distribution of absolute angular error as a
function of source spacing for two sources on the horizontal plane of the
microphone array (T60=0.4 s).
plotting=0;
rand_flac1=randi([3,2679],1,100);
rand_flac2=randi([3,2679],1,100);
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



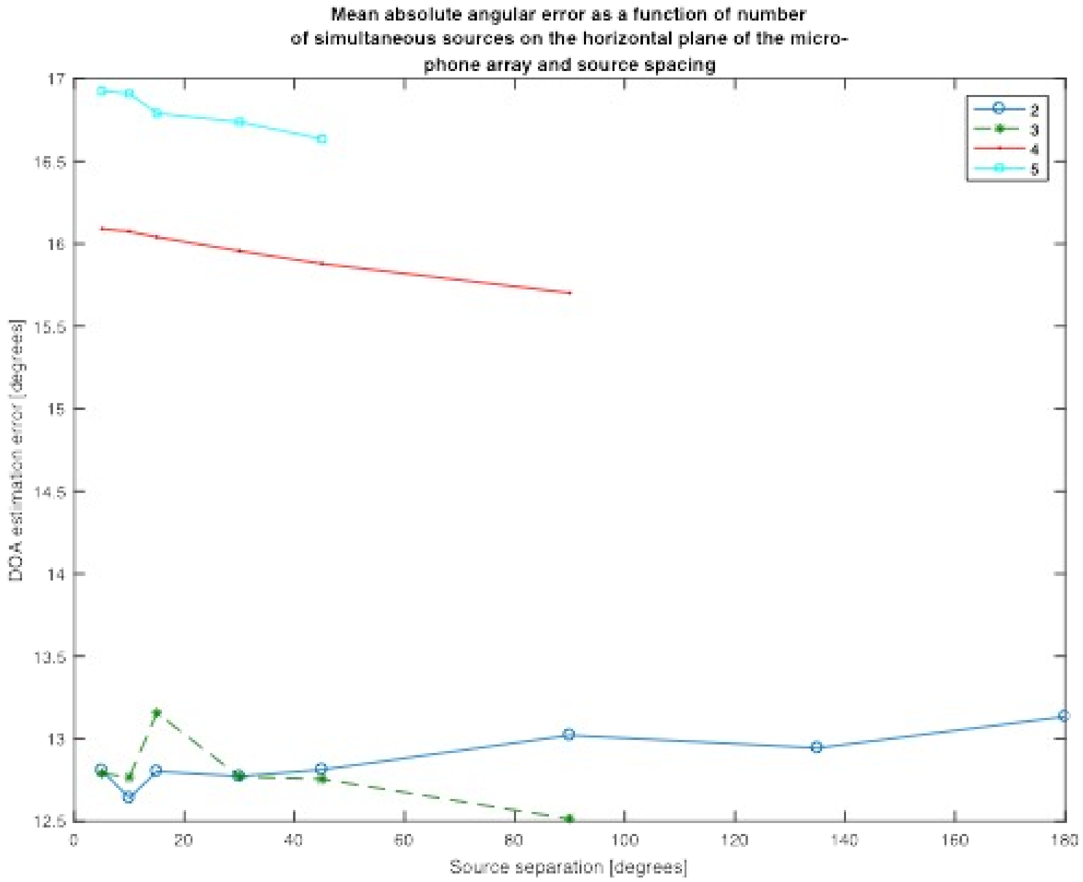
```
clear all;clc;close all
addpath('utils');
%% (Fig.6) Script to generate Mean absolute angular error as a function of
number of simultaneous sources on the horizontal plane of the microphone
array and source spacing (T60=0.4 s).
rand_flac1=randi([3,2679],1,100);rand_flac2=randi([3,2679],1,100);rand_flac3=
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