DAAP Homework #2

Marco Furio Colombo - Student ID: 10537094
Matteo Pettenò - Student ID: 10868930

***Abstract* - This report is intended to comment on the key steps in the MATLAB code and on the results obtained for the second homework of DAAP.**

1 Distance and frequency estimation

To determine the microphone distance that satisfies the anti-aliasing condition we need $d \leq \lambda/2$ and, since we have to cover all possible signal up to the Nyquist frequency ($F_n = F_s/2$), λ is given by c/F_n, therefore the maximum distance possible between each microphone is $d = c/F_s = 4.25\ cm$.
Then the frequencies f_{c_1} and f_{c_2} of the sinusoidal signals emitted by the two sources can be estimated from the analysis of a microphone signal magnitude: since we know that $w_{c_1} \neq w_{c_2}$ the magnitude will present two peaks and the frequencies are given by the indexes at which these peaks occurs. Commands *findpeaks* or *maxk* can be used to find the local maximums of a function but note that the search must be performed on half of the signal due to the frequency spectrum symmetry and $MATLAB$'s indexing must be fixed by adding -1 to the peak index. Fig. 1 confirms our estimation and clearly shows the two peaks at $f_{c_1} = 700\ Hz$ and $f_{c_2} = 500\ Hz$.

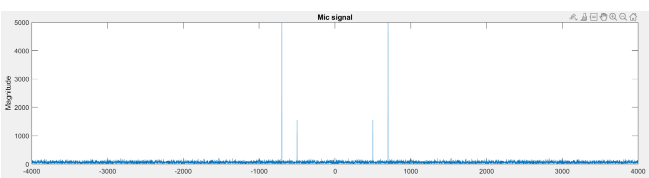

Figure 1: Microphone signal spectrum.

2 Delay-and-Sum beamformer

Once defined the array of candidate angles between $\pi/2$ and $-\pi/2$, with one degree steps, *Delay-And-Sum* method is straightforward applied for both sources and the estimated DOAs are $\overline{\theta}_1 = -70°$ and $\overline{\theta}_2 = 10°$. Fig. 2 shows the resulting pseudo-spectra and beam patterns.

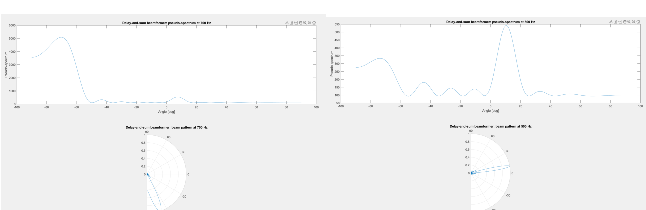

Figure 2: DAS pseudo-spectra and beam patterns.

Defined $l = (M-1) \cdot d$ as the ULA length, the resolution of the system depends on the main lobe's width, which is function of both the wavelength and the estimated DOAs:

$$w(\lambda_i, \overline{\theta}_i) = \frac{\lambda_i}{l \cdot |cos(\overline{\theta}_i)|} \tag{1}$$

For our two sources the wavelengths are $\lambda_1 = c/f_{c_1} \simeq 49\ cm$ and $\lambda_2 = c/f_{c_2} = 68\ cm$ and if we compute the width of the main lobes we get: $w(\lambda_1, \overline{\theta}_1) \simeq 30°$ and $w(\lambda_2, \overline{\theta}_2) \simeq 15°$. So as expected the resolution of the system is much higher around $0°$, but despite being almost double, if we playback the spatial filtered sound for the first source we hear the pure tone much clearer than the second source placed in front: this is due to the fact that the second source has a small SNR - e.g the power of the signal is much closer to the one the noise compared to the first source.

3 MUltiple SIgnal Classification

Next we applied the MUSIC method and Fig.3 shows the estimation results. We can appreciate how the peaks of the pseudo-spectra are far thinner than the ones obtained with *DAS*. This means MUSIC performs better than *DAS* when in presence of noises and its resolution is less afflicted by different signals coming from close angles. Furthermore the resolution is less dependent from $\overline{\theta}$, leading to better results as the angle get closer to $\pm\pi/2$.

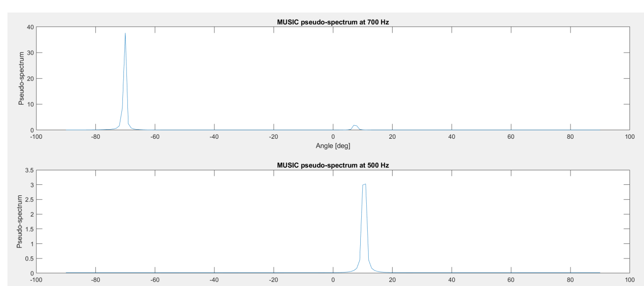

Figure 3: MUSIC pseudo-spectrum of both signal sources.

However, looking at *MUSIC*'s pseudo-spectra, we noticed imperfections in the peaks shape, especially for the source at f_{c_2}. This is caused by the chosen step between the candidate angles, which is large enough to be seen on the graphs. Lowering the step size, we obtained more precise results and understood that the second one is more pronounced because the real value is approximately $10.50°$ (worst case scenario for $1°$ steps), while the other one is about $-70.29°$. Using *MUSIC*, with a $1°$ step-size, we also obtain an angle of $11°$ for the signal at f_{c_2}, which is different from the result obtained with *DAS* (10°). This is due to the mentioned approximations that happen with bigger step-sizes.

4 Questions

4.1 DOA estimation methods

Delay-and-Sum (*DAS*) belongs to the family of non-parametric methods for DOA estimation also known as spatial filtering methods. The main advantage of DAS is that it is data independent: in fact it is designed around the assumption that no information about the covariance of the signals captured by the microphones is known.
To define *DAS* pseudo-spectrum function the spatial response $h(\overline{\theta})$ for each candidate direction $\overline{\theta}$ has to be computed first. Each $h(\overline{\theta})$ is built meeting two conditions:

1. **Distortionless condition:** leave undistorted the signals coming from the current candidate angle $\overline{\theta}$ (spatial response is 1),
2. **Power minimization:** minimize the power of the signals coming from all other directions. For this step *DAS* assumes that the signal captured by the microphones is spatially-white, meaning that it is coming from the all the directions simultaneously with the same power; it is this assumption that, differently from other non-parametric methods (e.g *Capon* method), makes *DAS* data independent.

On the other hand *MUSIC* belongs to the parametric method class for DOA estimation and thus it is based on a parametrization of the covariance structure of the microphones signals array. *MUSIC*'s pseudo-spectrum function is derived by exploiting the fact that $A^H \cdot V = 0_{N\times(M-N)}$ where A is the propagation matrix and V is the matrix whose columns are the eigenvectors related to the noise components of the input signals. Solution to these equations are given by the propagation vectors $a(\theta_k)$ associated with the DOAs and by defining the pseudo-spectrum as

$$p(\theta) = \frac{1}{a^H(\theta)VV^Ha(\theta)} \tag{2}$$

we will see peaks at DOAs angles. Ideally the pseudo-spectrum should go to infinity but as in the real world we can only use a sample of the covariance matrix, actually we would see peaks with very high (but finite) amplitude.
Regarding the performances of this two methods, we can say that in general *MUSIC* gives more accurate results since it is designed to achieve noise robustness, but, being a parametric method, it is data dependent and requires strict assumptions on the array data model, as the prior knowledge of the number of sources N. *DAS* instead doesn't assume that you know in advance the number of sources: everything you need to know to build the spatial filter is the frequency of operation to design the propagation vector and the spacing between the candidate angles. In a practical situation this could be a discriminating factor for preferring one method to another. Having to carry out the decomposition into eigenvalues of a matrix, *MUSIC* clearly presents a higher computational complexity than *DAS*: this could be another a decisive factor for the choice between the two methods in situations where it is necessary to process data in real-time.

4.2 Microphone spacing

Knowing only the sampling frequency, if we assume that the sampling theorem has been respected, using the Nyquist frequency as the worst case is an understandable choice. But analyzing the spectrum of the array signals, we see that in reality the highest frequency emitted by the two sources is $f_{c_1} = 700$ Hz which is much lower than the Nyquist frequency ($F_n = 4$ KHz). Therefore, choosing f_{c_1} as the worst case, we are still sure to be able to distinguish the two sources in space but the distance between the microphones in the array can be extended to $d = \lambda_1/2 \simeq 24.5$ cm; as the distance between the microphones increases, the main lobe width decreases so we can improve the resolution of the system.

4.3 Synthetizing the microphone signals

The following code shows how we generated the required array vector. First we generate the source vector that is made by the spectra of the sinusoidal signals emitted by the sources: we assumed that the frequency is still $\leq F_n$ (here we choose $f_c = 500$ Hz) and that the sources still emitts signals with different amplitudes. Then we compute the propagation matrix using the DOAs θ_1 and θ_2 estimated with *MUSIC* and we the generate the noise matrix (each row is a random WGN process with the given variance the represents the noisy component of the sensor). Finally the array $y(w)$ is generated by applying the ULA model defintion.

```
% Parameters
M = 3;
f = 500;
w_c = 2*pi*f;
amp_1 = 0.5;
amp_2 = 0.3;
% Generate the sinusoidal source signals
duration = N/Fs;
n = 0:1/Fs:(duration-1/Fs);
s_1 = amp_1 * sin(w_c * n);
s_2 = amp_2 * sin(w_c * n);
S = fft([s_1; s_2], N, 2);
% Compute the propagation matrix
w_s = w_c*d*[sin(MUSIC_DOA(1)); sin(
    MUSIC_DOA(2))] / c;
propagation_matrix = exp(-1j*w_s).^(0:M-1).';
% Generate a noise signal
var = 0.6;
w = randn(3, N) * sqrt(var);
W = fft(w, N, 2);
% Generate the array signals
y_generated = propagation_matrix * S + W;
```