P&D Information Systems and Signal Processing

Week 2: February 17-21, 2020
MUSIC-based DOA estimation

Giuliano Bernardi, Randall Ali, Santiago Ruiz[1], Marc Moonen

Version 1.0

Prelimenary remarks:

- Week 2 will also have a recording session in the audio lab, where audio signals will be recorded and used later in the project. Details, including the schedule, will be communicated.

Part 1: Narrowband MUSIC (single-source case)

1. In the simulation environment (see Week 1), create a non-reverberant acoustic scenario with one target audio source and no noise sources, where the target audio source is in the far field of a 5-microphone array with an inter-microphone distance equal to 5cm.

   Important: For all experiments in Week 2, sample the RIRs at 44.1kHz to obtain simulated recordings with a sufficiently high time resolution. It may be appropriate to include a test at the beginning of your code to check if the sampling frequency of the RIRs is indeed 44.1kHz, to avoid any mistakes.

2. Create an m-file `MUSIC_narrowband.m` based on the MUSIC algorithm as explained in the Appendix. The m-file should perform the following tasks:

   - Generate the microphone signals defined by the scenario in the simulation environment (re-use the code in `create_micsigs.m` from the previous session). Use `speech1.wav` as the file name of the target audio source, and truncate the signal to 10s.
   - Compute a (subsampled) short-time Fourier transform (STFT) of each microphone signal (use a DFT with window length $L = 1024$, and use 50% overlap). Stacking these STFT representations for each microphone signal returns a 3D array of dimensions $[M \times n_F \times n_T]$ with M the number of microphones, n_F the number of frequency bins and n_T the number of time samples. See the Appendix for more details.
   - Compute the signal power in each frequency bin, and select the frequency bin $\omega_{\max}$ with the highest power (averaged over all the microphones and time frames).
   - Evaluate the pseudospectrum $p_\omega(\theta)$ for the selected frequency bin, i. e. $p_{\omega_{\max}}(\theta)$, for $\theta = [0 : 0.5 : 180]$. Avoid using a `for` loop to speed up the code and double-check the order of the eigenvalues as computed in MATLAB.
   - Plot the pseudospectrum.

[1] For questions and remarks: `santiago.ruiz@esat.kuleuven.be`.

- Find the DOA of the target audio source by identifying the largest peak in the pseudospectrum. Indicate the selected DOA on the previous plot (e.g., using `stem`).
- Store the DOA estimate in a variable `DOA_est` and store this variable in a mat-file with the same name

Remark: In the GUI, the numbering of the microphones is from bottom to top, i.e., the first RIR corresponds to the microphone with the smallest vertical coordinate. This means that a source signal with a DOA of 180°, i.e. the end-fire direction where the source is under under the microphone array, first impinges on the first microphone. Use this convention when writing your code.

3. Test your code on the generated acoustic scenario (using the 'Draw' button in the GUI).

   Only if you have a DOA estimation error of less than 5°, you can proceed with the next step.

4. The above description of the MUSIC algorithm assumes that there is no noise. What happens if uncorrelated noise of equal power is added to each microphone? Predict the effect theoretically, and confirm experimentally.

   Hint: What does the noise covariance matrix look like? What happens to the eigenvalues of $\mathbf{R}_{yy}(\omega)$ when this matrix is added? And more importantly for MUSIC, what happens to its corresponding eigenvectors?

Part 2: Narrowband MUSIC (multi-source case)

1. The MUSIC algorithm be generalized for the case of Q target sources, assuming that Q is known. The changes are very minor: there will be a change in the dimension of $\mathbf{E}(\omega)$, as well as a change in the number of peaks that have to be selected in the pseudospectrum.

   Modify your m-file `MUSIC_narrowband.m` accordingly. Note that Q can be inferred from one of the variables in `Computed_RIRs.mat` as generated by the simulation environment. Identify the Q largest peaks and indicate the corresponding estimated DOAs on the plot of the pseudospectrum. Store the DOA estimates in a vector variable `DOA_est` and store this variable in a mat-file with the same name.

   Hint: In the single-source-case, you can simply search for the maximum of the pseudospectrum to find the DOA. In the multi-source case, you will first have to identify the samples of the the pseudospectrum that effectively correspond to peaks (based on the neighboring samples), and then sort these to find the Q largest ones.

2. Add another target audio source to the scenario defined in Part 1. Make sure that the two target audio sources have a significantly different DOA (e.g., > 45°). Use `speech2.wav` as the filename of the second speech source. Test your code on this two-source scenario.

Only if you have a DOA estimation error of less than 5° for both sources, you can proceed with the next step.

Part 3: Wideband MUSIC

1. Create a non-reverberant acoustic scenario with two target audio sources and no noise sources where the target audio sources are in the far field of a 5-microphone array with an inter-microphone distance equal to 10cm. Put the first microphone (i.e., the lowest) at position (4, 4), and the two target speech sources at position (2, 2) and (1.5, 2).

2. Use `MUSIC_narrowband.m` to estimate the DOA of both target sources. What do you observe? What does the pseudospectrum look like?

3. In your current implementation of the MUSIC algorithm, you have only included information from one frequency bin. A natural extension is to combine the information from multiple frequency bins to improve the resolution of the MUSIC algorithm. This can be done in many different ways, e.g., by averaging over the pseudospectra or over the DOA estimates, with different ways of averaging (arithmetic, geometric, harmonic, ...). This choice has a large impact on the performance of the algorithm. Here, we suggest to use a geometric averaging of the pseudospectra, which has been shown to provide the most robust results, although you are free to experiment with other choices.

   Create a new m-file `MUSIC_wideband.m`, which is similar to the `MUSIC_narrowband.m`, but with the following modifications:

   - Use the command `hann` to compute a Hann-windowed STFT (again with $L = 1024$ and 50% overlap)
   - The Q peaks are selected in the geometrically averaged pseudospectrum

     $$\overline{p}(\theta) = \left(\prod_{k=2}^{L/2} p_{\omega_k}(\theta) \right)^{\frac{1}{\frac{L}{2}-1}} \tag{1}$$

     with $\omega_k = 2\pi f_k$, where f_k corresponds to the discrete frequency in the k-th frequency bin of the STFT. Due to the symmetry of the DFT, the second half of the frequency bins (for $k = L/2+2, \ldots, L$) are omitted in (1). Why are the bins $k = 1$ and $k = L/2 + 1$(which do not have a mirror frequency) omitted?
   - Plot two figures: one containing the individual pseudospectra $p_{\omega_k}(\theta)$, $k = 2, \ldots, L/2$ and one containing $\overline{p}(\theta)$. In the latter, indicate the selected Q largest peaks.

   Hint: Computing the geometric average directly based on (1) often results in a numerical over- or underflow (why?). Transform this formula to the logarithmic domain for a more robust computation, and transform the final result back to the linear domain.

4. Test your code first on the single-source scenario of Part 1 (to make sure that it works properly).

Only if you have a DOA estimation error of less than 5°, you can proceed with the next step.

5. Test your code on the two-source scenario. Do you observe an improved resolution compared to the narrowband MUSIC algorithm?

6. Spectral leakage from high-power frequency bins to low-power bins often introduces artifacts in the averaged pseudospectrum. Which of the steps in the MUSIC algorithm reduces this effect, and why? For the scenario defined earlier, check what happens to $\overline{p}(\theta)$ if you do not apply this step.

Part 4: Effect of reverberation

1. Create a scenario with one target audio source at position (1,1), which is recorded by a 5-microphone array with an inter-microphone distance $d = 10$cm, with the first microphone at position (4,4). Set the T60 to 0.5s and set the room size to 5m. Apply the wideband MUSIC algorithm and inspect the averaged pseudospectrum. Check if reverberation has a significant effect on the performance of the MUSIC algorithm.

2. Re-run the same experiment, but with the target audio source at position (2,2). Re-run the same scenario once more with the target audio source at position (3,3). Explain the obtained results.

Part 5: Extension of the MUSIC algorithm to the head-mounted microphones case

1. Consider the head-mounted microphone setup (see Week 1 Fig. 3). Generate a set of microphone signals with `part2_track1_dry.wav` coming from a chosen direction (loudspeaker) on the left-hand side and `part2_track2_dry.wav` coming from a chosen direction (loudspeaker) on the right-hand side.

2. Apply the wideband MUSIC algorithm to the two microphones pertaining to the right ear, $\mathbf{y_{R1}}, \mathbf{y_{R2}}$.

3. Apply the wideband MUSIC algorithm to the two microphones pertaining to the left ear, $\mathbf{y_{L1}}, \mathbf{y_{L2}}$

4. Apply the wideband MUSIC algorithm to the two frontal microphones, $\mathbf{y_{L1}}, \mathbf{y_{R1}}$

5. Check how the results differ based on the chosen pair of microphones. Explain.

6. Modify the wideband MUSIC algorithm such that it can use all the four microphones, $\mathbf{y_{L1}}$, $\mathbf{y_{L2}}$, $\mathbf{y_{R1}}$, and $\mathbf{y_{R2}}$. For this, first adjust the model defined by Eqs (2) and (3) and then use this to extend the wideband MUSIC algorithm.

7. Compare the performance of the algorithm when 2-microphone and 4-microphone arrays are used.

Appendix

Introduction to MUSIC

The MUltiple SIgnal Classification (MUSIC) algorithm is a DOA estimation algorithm that can handle any number of input microphone signals and allows to estimate the DOAs of multiple simultaneous target sources. Only a brief explanation is provided, the reader is referred to the www for further details.

Consider a microphone array with M microphones that captures a time domain signal $\mathbf{y}(t) = [y_1(t) \ \dots \ y_M(t)]^T$ as shown in Fig. 1. Applying a Short Time Fourier Transform (STFT) to $y(t)$, yields the M-channel microphone signal $\mathbf{y}(k, \omega) = [y_1(k, \omega) \ \dots \ y_M(k, \omega)]^T$, in the STFT domain, where k is the frame index and $\omega = 2\pi f$ with f being the frequency. The function of k is dropped in the remaining notation for conciseness. Assuming a single target source signal $s(\omega)$, the vector signal $\mathbf{y}(\omega)$, at time frame k, can be described as:

$$\mathbf{y}(\omega) = \mathbf{a}(\omega)s(\omega) + \mathbf{n}(\omega) \tag{2}$$

where $\mathbf{a}(\omega)$ is the so-called steering vector containing the M acoustic transfer functions from the target source to the M microphones, and $\mathbf{n}(\omega)$ is noise. In a non-reverberant scenario, and assuming a small inter-microphone distance d such that the signal power can be assumed to be equal over the different microphones, the steering vector can be approximated as (why?)

$$\mathbf{a}(\omega) = a_1(\omega) \begin{bmatrix} 1 \\ e^{-j\omega\tau_{12}} \\ \vdots \\ e^{-j\omega\tau_{1M}} \end{bmatrix} \tag{3}$$

where $a_1(\omega)$ is the acoustic transfer function from the target source to the first microphone, and where τ_{1m} is the TDOA (in seconds) between microphone 1 and microphone m (defined to be positive if microphone 1 is closest to the target source). Note that the TDOA τ_{1m} directly depends on the DOA of the target source. We define the so-called array manifold vector

$$\mathbf{g}(\omega, \theta) = \begin{bmatrix} 1 \\ e^{-j\omega\tau_{12}(\theta)} \\ \vdots \\ e^{-j\omega\tau_{1M}(\theta)} \end{bmatrix} \tag{4}$$

which is the parameterized steering vector as a function of the DOA θ, and normalized such that the first entry is equal to 1.

The $M \times M$ spatial correlation matrix of the microphone signals at frequency ω is defined as $\mathbf{R}_{yy}(\omega) = E\{\mathbf{y}(\omega)\mathbf{y}(\omega)^H\}$. If we assume a noiseless scenario ($\mathbf{n}(\omega) = 0$), then (2) implies that $\mathbf{R}_{yy}(\omega)$ is a rank-1 matrix, proportional to $\mathbf{a}(\omega) \cdot \mathbf{a}(\omega)^H$. The steering vector $\mathbf{a}(\omega)$ will then be orthogonal to the subspace spanned by the $M - 1$ eigenvectors of $\mathbf{R}_{yy}(\omega)$ corresponding to the $M - 1$ smallest (in this case zero-valued) eigenvalues. Let $\mathbf{E}(\omega)$ denote the $M \times (M - 1)$ matrix containing these eigenvectors in its columns, then we define the MUSIC pseudospectrum at frequency ω as the function $p_\omega(\theta) : [0, 180] \to \mathbb{R}^+$, where

$$p_\omega(\theta) = \frac{1}{\mathbf{g}(\omega, \theta)^H \mathbf{E}(\omega) \mathbf{E}(\omega)^H \mathbf{g}(\omega, \theta)} \ . \tag{5}$$

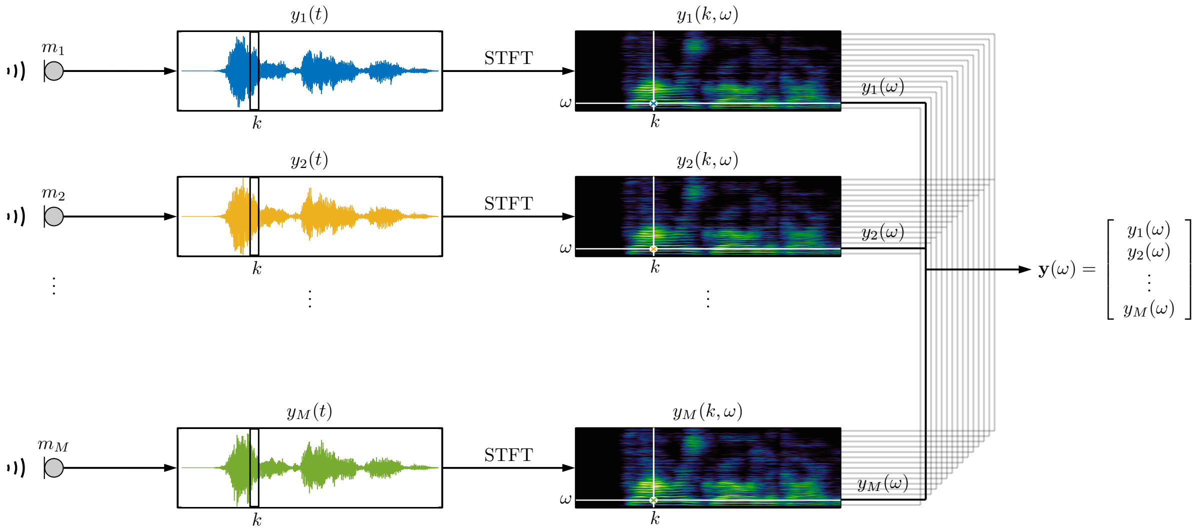

Figure 1: Processing scheme of the M microphone signals from the time domain to the STFT domain. At time frame k and frequency ω, the received signals are represented by the vector $\mathbf{y}(\omega_\mathrm{f})$ as shown on the right-hand side of the figure.

Verify the scalar nature of (5) by quickly performing a dimensionality analysis of $p_\omega(\theta)$. The pseudospectrum will show a peak at the θ corresponding to the true DOA (why?). The final step of the MUSIC algorithm is then to find this peak, by evaluating $p_\omega(\theta)$ for all values of $\theta \in [0, 180°]$.

Hint: Stacking the different array manifold vectors parameterized in $\boldsymbol{\theta} = [\theta_0 \dots \theta_{n_\theta - 1}]$ in matrix form

$$\mathbf{G}(\omega) = \begin{bmatrix} 1 & 1 & \dots & 1 \\ e^{-j\omega\tau_{12}(\theta_0)} & e^{-j\omega\tau_{12}(\theta_1)} & \dots & e^{-j\omega\tau_{12}(\theta_{n_\theta-1})} \\ \vdots & & \ddots & \\ e^{-j\omega\tau_{1M}(\theta_0)} & e^{-j\omega\tau_{1M}(\theta_1)} & \dots & e^{-j\omega\tau_{12}(\theta_{n_\theta-1})} \end{bmatrix}, \tag{6}$$

the $[n_\theta \times 1]$ vector containing the pseudospectrum at frequency ω spanning the whole angular range is given by:

$$\mathbf{p}_\omega = \frac{1}{\operatorname{diag}\left\{\mathbf{G}(\omega)^H \mathbf{E}(\omega) \mathbf{E}(\omega)^H \mathbf{G}(\omega)\right\}}. \tag{7}$$