ELG 5376 Digital Signal Processing

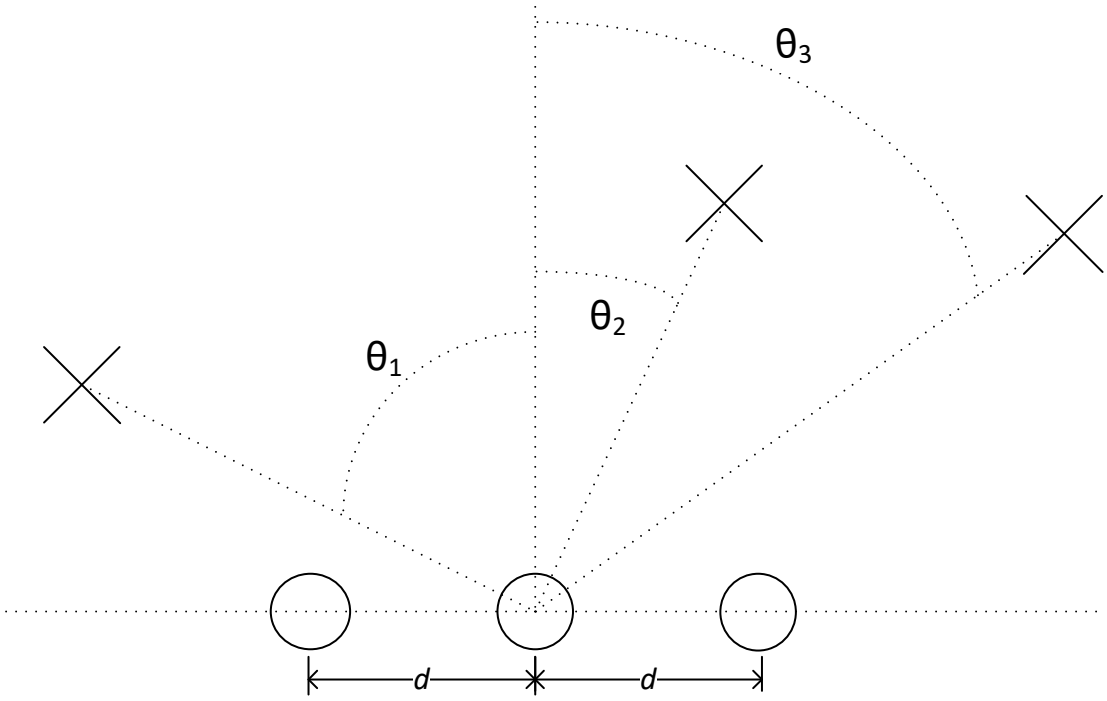
Term Project, Winter 2018

Problem setting and simulated system

This project's objective is the extraction of a target speech signal from 3-channel signal mixtures simulating a linear microphone array in an acoustic environment. The acoustic environment includes:

- 3 speakers, who can all be considered as directional far field point sources (plane wave propagation):
  - an English-speaking female, who is the target to be extracted;
  - a German-speaking female, who is considered as an interference;
  - an English-speaking male, who is also considered as an interference;
- non-directional white Gaussian noise in each channel or sensor (spatially uncorrelated between the 3 sensors, i.e., no coherence between them).

The following figure illustrates the setup, where the "Xs" represent the 3 directional sources with unknown angles $\theta_1, \theta_2, \theta_3$ and the "Os" represent the 3 microphone sensors:



To generate the signals, a speed of sound of 343 m/s and a distance between microphones d = 25 cm was used. The signals from the 3 microphones were generated over 12 seconds, at a sampling rate of 16 kHz. The first 2 seconds of the signals consist only of noise (no speech source is active then). This was done to help you measure the noise statistics (e.g. noise level at each frequency), which can be required by some of the speech enhancement methods to be discussed later.

In several aspects, this simulated setup is simplified compared to real-life situations:

- all the sources are fixed spatially (no movement of the sources, no environmental changes such as objects moving), and the speech sources are active during a continuous 10 seconds period;

- free field (anechoic) conditions are simulated, as opposed to reverberant conditions with possibly long echoes, room acoustics, etc. This means that the impulse responses / frequency responses between the sources and the sensors are simple, i.e., they correspond to pure delays;
- there is no physical object between microphones (this would also affect the transfer functions and cause effects such as diffraction);
- plane wave propagation is assumed, meaning that we consider that the sources are in the far field, with no near field propagation effects, and it means that each source reaches all the different sensors from the same angle;
- the time-frequency statistics of the white Gaussian noise are stationary (no variation over time), and 2 seconds of noise-only signals are available at the beginning of the files;
- offline processing is possible (no real time processing with low latency and short frames is required here, so you can read and analyze the whole 12 seconds of signals before generating the output enhanced speech file);
- it is often not possible in physical systems to have such large distances (25 cm) between microphones (e.g., in laptops, cell phones), and large distances help in the detection of the angles $\theta_1, \theta_2, \theta_3$ and for the target source extraction.

Signal processing methods to use:

In order extract the target speech component (English-speaking female speech), you are required to proceed in 3 steps:

1. Estimation of the directions of arrivals /angles of arrivals (DOA/AOA) of the sources, i.e., the angles $\theta_1, \theta_2, \theta_3$. Equivalently, the "time differences of arrival" (TDOA) of the sources can be estimated. This information is useful for the step below.
2. To remove the directional interferences (German-speaking female speech and English-speaking male speech), it is required to perform beamforming filtering, i.e., multichannel spatial filtering producing a single-channel output where the directional interferences have been removed.
3. The single-channel output of the beamforming stage should have both the target English-speaking female signal and some strong background noise component (possibly even stronger than in the original signal). Since the background noise is stationary and since its statistics in the time-frequency domain differ from the statistics of the target speech signal, it is possible to use single channel time-frequency filtering methods to attenuate the noise (as opposed to spatial domain filtering like the beamforming of the previous step). This step of filtering is often called "post-filtering", and it can be achieved using single channel speech enhancement de-noising methods.

A search using tools such as Google, Google Scholar and IEEE Xplore should be sufficient to provide you with plenty of possible references. You are encouraged to find your own original references. However, having performed a quick informed Google search of appropriate keywords, I also provide some possible references below:

1) For detecting the angles of arrivals or time-differences-of-arrival of the sources (TDOA detection):

http://en.wikipedia.org/wiki/Acoustic_source_localization

A recent paper comparing state of the art TDOA estimation methods (including the simple GCC-PHAT method) is:

http://www.sciencedirect.com/science/article/pii/S0165168411003525

and some python code for the GCC-PHAT method can easily be found from Google (or equivalently Matlab code can be found, which can then be translated in python).

2) For the beamforming part:

One possibility is to use a Minimum Variance Distortionless Response beamformer (MVDR), to handle the directional interferences. This paper investigating the MVDR beamformer performance can be used as a reference (some key equations are (4), (18), (29)-(30)):

http://externe.emt.inrs.ca/users/benesty/papers/aslp_jan2014.pdf

(note: you should use only one multichannel beamforming filter (or beamformer) to remove the two interferers, i.e., not use a beamformer for each interferer to cancel).

Alternatively, a Google search on acoustic beamforming for python can produce the following result (and other results):

acoular 16.5 Library for acoustic beamforming https://pypi.python.org/pypi/acoular

(note: I have not personally used that library). Therefore, many solutions are possible.

3) For the post-processing part (single channel speech enhancement de-noising):

Methods such as spectral subtraction, Wiener filtering, log-MMSE, and short-time spectral amplitude (STSA) estimation can be applied. Some references can be found here:

http://www2.ece.rochester.edu/~zduan/teaching/ece477/lectures/Speech%20Enhancement.pdf
http://www.eng.biu.ac.il/~gannot/Tutorials/tutorial_icassp_2012_part1_intro

as well as in several other sources. Demo codes for speech enhancement can be found either directly for python or from the Matlab central https://www.mathworks.com/matlabcentral/ (to be converted in python).

Methods not to be used

Note: there are other methods such as methods based on blind source separation or independent component analysis (BSS/ICA) which can also be used to extract/separate the different sources, and which do not require any estimation of the sources DOA, etc. Those methods can work very well for the simple conditions considered here. However, due to the availability of BSS/ICA toolboxes and to avoid that the project becomes just the use of a single "black box" tool, you are required to follow the above 3-steps approach involving DOA estimation, beamforming and post-processing de-noising.

Projects evaluation

Projects are to be done in teams of two students.

In your report, it is required to present the mathematical description of the proposed solution. An experimental component involving simulation of your proposed methods is required. You may make use, in part, of public domain software packages, libraries, codes, etc., as long as you clearly state this, and as long as you clearly identify what is your contribution in the development of the simulated system and the proposed solution.

The final report should be in the form of a journal paper, with a maximum of twelve pages. Use the format of the IEEE Institute, for which Word or LaTex templates can be found at:

(Word) https://ieeeauthor.wpengine.com/wp-content/uploads/Transactions-template-and-instructions-on-how-to-create-your-article.doc
(Latex) https://ieeeauthor.wpengine.com/wp-content/uploads/WIN-or-MAC-LaTeX2e-Transactions-Style-File.zip

You must write the reports yourself; plagiarism can lead to a failing mark, see the leaflet "Beware of Plagiarism" https://www.uottawa.ca/about/sites/www.uottawa.ca.about/files/plagiarism.pdf for guidelines. This leaflet uses several examples to explain clearly what is acceptable and what is not acceptable when citing, paraphrasing or using previous work.

The following criteria will be considered when grading the reports:

- the technical soundness of the material presented;
- the clarity and quality of the document, including the presentation of results;
- how much "added-value" or new contribution was made (for example the degree of new code written, compared to the degree of previously existing code/libraries);
- in addition to the report, you are also required to submit a 12 seconds 16 kHz sound file with your extracted target signal (output signal), which will be considered in the evaluation, as well as your source code which may be executed by the Corrector.

Deadline:

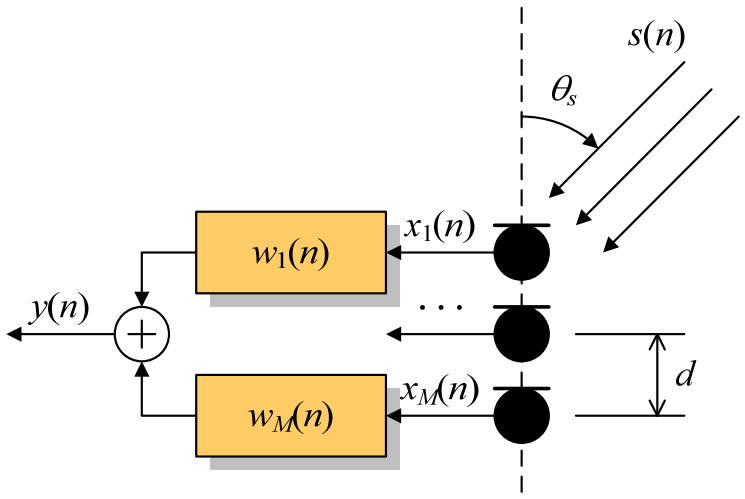
Last day of classes: April 11th (cannot be delayed because a Corrector will be marking the reports and her contract ends in April). A penalty of 10% per day can be applied for late submissions.

Reports and sound files are to be submitted using the Virtual Campus platform. To keep things simple, please provide only one submission per team (submission to be done with the account of only one team member).

A few additional hints:

- Free field DOA / TDOA, steering vector, and beamformer formulation

Consider a linear array of M sensors/microphones with equal spacing d, with a far-field (plane wave) source signal impinging on the array at angle θ from the axis of the array.



The microphone signals $x_i(n)$ are simply delayed versions of the source signal $s(n)$. For each microphone this can be written as:

$$x_i(n) = s(n-(i-1)\tfrac{d}{c}\cos\theta) \qquad 1 \le i \le M$$

While the source angle θ is the direction of arrival (DOA), the quantity $(i-1)\tfrac{d}{c}\cos\theta$ in the above equation is the time difference of arrival (TDOA), i.e., it is the time difference between when the signal $s(n)$ reaches the different microphones. In the frequency domain this becomes:

$$X_i(e^{j\omega}) = S(e^{j\omega})e^{-j\omega(i-1)\frac{d}{c}\cos\theta} \qquad 1 \le i \le M$$

For each possible source angle θ, this can lead to a vector form with respect to the *M* microphones:

$$\underline{X}(e^{j\omega}) = S(e^{j\omega})\underline{D}(e^{j\omega}) \qquad \underline{D}(e^{j\omega}) = [1 \quad e^{-j\omega\frac{d}{c}\cos\theta} \quad \cdots \quad e^{-j\omega(M-1)\frac{d}{c}\cos\theta}]^T$$

$\underline{D}(e^{j\omega})$ is called a steering vector for the source from angle θ, and we can see that it also includes the information of the TDOA $(i-1)\tfrac{d}{c}\cos\theta$.

The beamformer output is simply a linear combination of the microphone signals, i.e., a sum of convolutions between the beamformer filters and the signal from each sensor:

$$y(n) = \sum_{i=1}^{M} w_i(n) * x_i(n),$$

or, using the frequency domain and the vector form:

$$Y(e^{j\omega}) = \underline{W}^H(e^{j\omega})\underline{X}(e^{j\omega}).$$

$$\underline{X}(e^{j\omega}) = [X_0(e^{j\omega}) \quad X_1(e^{j\omega}) \quad \cdots \quad X_{M-1}(e^{j\omega})]^T \qquad \underline{W}(e^{j\omega}) = [W_0(e^{j\omega}) \quad W_1(e^{j\omega}) \quad \cdots \quad W_{M-1}(e^{j\omega})]^T$$

- Should the beamforming filtering be performed in the frequency domain or the time domain?

The design of the beamformer is often done in the frequency domain (but this depends on the beamforming method chosen), while the actual filtering with the beamforming filters can be done either in the frequency domain or the time domain.

If the frequency domain is chosen:

If the FFT size of your filter and your signals to be filtered are different, it will make the frequency domain processing complicated. Should you use the STFT (overlapping signal frames processed in frequency domain) or a single FFT on the whole signals? Both can work. But for the case that we have here, where everything is fixed spatially (no movement), we can use constant beamformer coefficients so we don't need to process the signals on a frame by frame basis with the STFT.

If you choose time domain filtering:

Be aware that many beamformer designs assume that the filtering is to be performed as:
y(f) = W'(f) X(f)
(where X(f), W(f) are *M* by 1 vectors, *M* is the number of sensors, and the ' operator is the Hermitian operator, which is both a transpose operator and a complex conjugate operator. This can also be written as:
y(f) = sum_for_all_i (conj(wi(f)) xi(f))

where each "i" represents a sensor.
The conjugate operator in the above expression is important. Since the beamformer coefficients in the time domain (impulse responses) are real-valued, then conj(wi(f)) = wi(-f). And having wi(-f) means that the corresponding time domain impulse responses are wi(-n). So the corresponding time domain responses need to be flipped (wi(-n)) if we are to use those wi coefficients in a filter function. The time domain coefficients that you get from the inverse FFT of the frequency domain coefficients thus require some processing: 1) a function like numpy.fft.ifftshift() to give a causal shape to the coefficients, and 2) a time flip operation.

- Matrix inversion in the beamforming design

In many beamformer design there is a matrix inversion involved. The matrix is often ill-conditioned or even singular, so in practice instead of inverting directly the matrix you should add some regularization through "diagonal loading", i.e., adding a small value to each element of the matrix diagonal so that its conditioning is improved and it becomes invertible. It turns out that the level of these small values added to the diagonal can be critical in the performance of the beamformer and it needs to be adjusted carefully.

An alternative is to use a “pseudo-inverse” function instead of a matrix inversion function, but it is not always a good idea, as it (approximately) corresponds to using a very small level of regularization, and it may not always lead to the best performance.