Localization of Acoustical Sources

NKTP
Signal Processing Group
Technische Universität Darmstadt

2014

Abstract

In this practical, the localization of acoustical sources is discussed. The following objectives are emphasized:

1. Estimation of time-difference of arrival (TDOA) and angle of arrival (AOA) using correlation and generalized correlation functions.
2. The fusion of multiple AOA measurements from distributed microphone arrays is to achieve sound source localization.
3. The MATLAB© environment is used to implement and test the above objectives.

1 Introduction

Localization of acoustical sources, especially speaker localization, is a task which is often required for audio applications, such as hands-free telephony or telephone/video conferencing. With the knowledge of the angle-of-arrival (AOA) or the location of an acoustical source, a spatial filter can be used to suppress noise and interfering sources.

The AOA of an acoustical wave arriving at multiple microphones is directly related to the time-delay between the microphone inputs. In this practical, we show how to use cross-correlation techniques to estimate the time-difference of arrival (TDOA) between two signals received at a microphone pair, and how TDOA is related to the AOA. For speech signals, we

further show the drawbacks of using the standard cross-correlation against the more advanced generalized cross-correlation. Also, the fusion of multiple AOA measurements to localize the sources is discussed.

2 Theoretical Background

In this section, the signal model is introduced. Moreover, the estimation of TDOA and AOA using correlation between signals received from a microphone pair is presented. Finally, fusion of AOA measurements from multiple microphone pairs is summarized.

2.1 Signal Model

Figure 1 depicts an acoustical source signal $S(t)$ impinging onto a pair of microphones M_1 and M_2. The signal $S(t)$ reaches the first (reference) microphone M_1 at time t, and the second microphone M_2 at time $t + \Delta t$, i.e., delayed by Δt. It is assumed that the source is far away from the microphone pair. Thus, the waves at the microphone pair is considered planar, as shown in Figure 1. Under this assumption, the relation between the delay Δt and the AOA, denoted as θ, is

$$\Delta t = \frac{d}{v_s} \sin(\theta), \tag{1}$$

where d is the microphone spacing and $v_s = 343$ m/s is the speed of sound at room temperature.

We consider the microphone signals as a superposition of the analog speech signal S and noise, where one measurement is delayed and scaled with respect to the other signal. Thus, the output at the two microphones is modeled as

$$X_1(t) = S(t) + N_1(t) \tag{2}$$
$$X_2(t) = gS(t - \Delta t) + N_2(t), \tag{3}$$

where $X_1(t)$ and $X_2(t)$ are the outputs of M_1 and M_2, respectively, g is a scaling factor which represents the gain difference between the two microphones, and $N_1(t)$ and $N_2(t)$ are the noises at M_1 and M_2, respectively. The noise processes $N_1(t)$ and $N_2(t)$ are assumed to be uncorrelated. The signal diagram is shown in Figure 2.

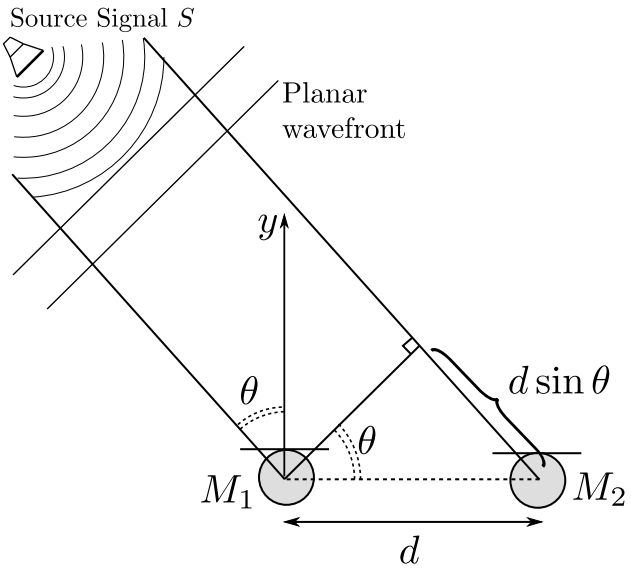


Figure 1: A plane wave is impinging, from azimuth angle-of-arrival θ, onto a pair of microphones M_1 and M_2, spaced by distance d.

The analog signals at the output of the microphones are sampled at $t = nT$, where $n \in \mathbb{N}$ and T is the sampling period. Thus, using Equation (3), the sampled signals are written as

$$X_1(n) = S(n) + N_1(n) \tag{4}$$

$$X_2(n) = g \underbrace{S(nT - \Delta t)}_{S_{\Delta t}(n)} + N_2(n). \tag{5}$$

Note that Δt does not need to be an integer multiple of the sampling interval T. *Thus, $S_{\Delta t}(n)$ is not simply a delayed version of $S(n)$, and therefore it may be difficult to estimate Δt from the sampled data.* However, it can be shown using the reconstruction property of band-limited analog signals, that this problem can be resolved by *an adequate interpolation*.

Assuming fixed and finite source and noise waveforms, the Fourier transforms of (4) and (5) are:

$$\begin{aligned} X_1(e^{j\omega}) &= S(e^{j\omega}) + N_1(e^{j\omega}) \\ X_2(e^{j\omega}) &= gS(e^{j\omega})e^{-j\omega\Delta t/T} + N_2(e^{j\omega}) \end{aligned}$$

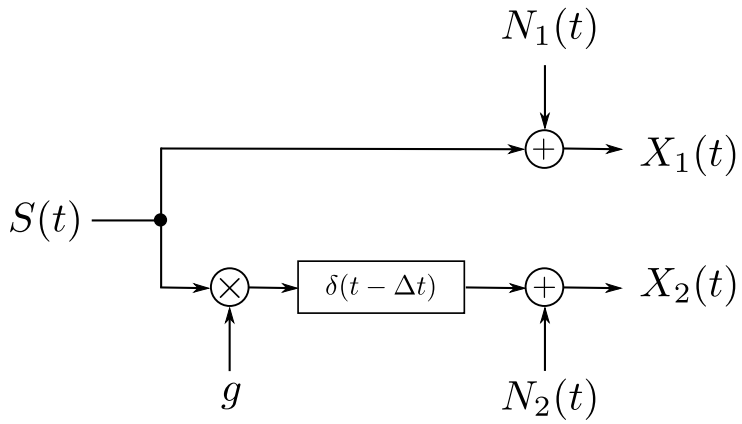


Figure 2: Signal flowgraph in the analog domain, where δ is the unit impulse.

2.2 TDOA Estimation

In this section, it is shown how to estimate the TDOA using the cross-correlation and the generalized cross correlation function.

2.2.1 The Cross-Correlation Function

By assuming that X_1 and X_2 are jointly stationary, and that N_1 and N_2 are zero-mean and uncorrelated, the cross power spectral density (PSD) of X_1 and X_2 is

$$P_{X_1X_2}(e^{j\omega}) = \sum_{\kappa=-\infty}^{\infty} r_{X_1X_2}(\kappa)e^{-j\omega\kappa} = gP_{SS}(e^{j\omega})e^{j\omega\Delta t/T}. \tag{6}$$

This means that the time-delay is contained in the phase of the cross-PSD. The cross second order moment function (cross-SOMF), also called cross-correlation function, between the sampled processes can then be approximated as

$$r_{X_1X_2}(\kappa) = \mathrm{E}\{X_1(n+\kappa)X_2(n)\} \approx g\, r_{SS}\left(\kappa + \frac{\Delta t}{T}\right).$$

A simple algorithm is to estimate the time-delay Δt by the maximum value of an estimate of $r_{X_1X_2}(\kappa)$:

$$\frac{\Delta t}{T} \approx \Delta n = -\arg\max_{\kappa} \hat{r}_{X_1X_2}(\kappa). \tag{7}$$

In Figure 3, the cross-correlation of the signals at the output of the microphone pair is shown in two cases. First, the signal $S(t)$ is a white noise.

Second, the signal $S(t)$ is a speech signal. Note that the cross-correlation shows a sharp peak only if $S(n)$ is white noise. For non-white signal waveforms such as speech, a modification is necessary to produce a sharp peak. This can be achieved, e.g., using the so called generalized cross-correlation (GCC) function.

2.2.2 The Generalized Cross-Correlation Function

As indicated by the Fourier transform pair in Equation (6), the cross-SOMF is related to the cross-PSD by inverse Fourier transformation. The GCC is defined similarly, except that the cross-PSD is multiplied by a suitable weighting function $\psi(e^{j\omega})$ before inverse Fourier transformation, i.e.

$$g_{X_1X_2}(\kappa) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \psi(e^{j\omega}) P_{X_1X_2}(e^{j\omega}) e^{j\omega\kappa} d\omega.$$

The weighting is intended to equalize the magnitude of the cross-PSD while leaving the phase untouched. There are two common approaches for the weighting, which are the smoothed coherence transform (SCOT)

$$\psi(e^{j\omega}) = \frac{1}{\sqrt{P_{X_1X_1}(e^{j\omega}) P_{X_2X_2}(e^{j\omega})}}$$

and the phase transform (PHAT)

$$\psi(e^{j\omega}) = \frac{1}{|P_{X_1X_2}(e^{j\omega})|}.$$

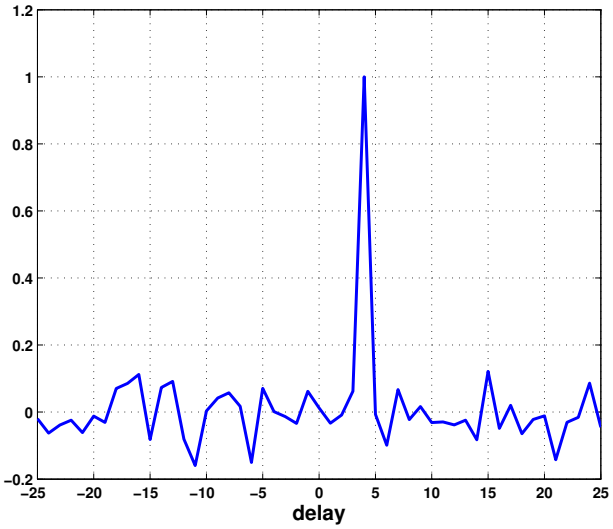
The advantage of the latter weighting is that the cross-PSD has to be computed anyway and thus no additional calculations are required.

Figure 4 shows the GCC of the signals at the output of the microphone pair, where $S(t)$ is the same speech signal used in Figure 3b. Remark that GCC shows a sharp peak.

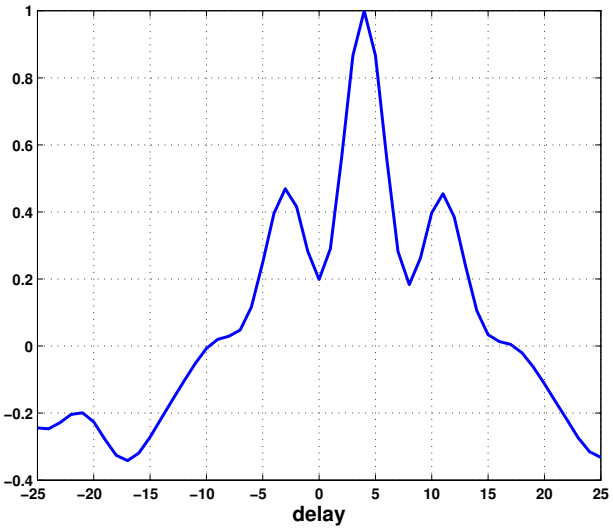
The estimation of the AOA from the TDOA is followed directly using Equation (1).

2.2.3 Practical Considerations

From signal sequences $X_1(n)$ and $X_2(n)$ for $n = 0, \ldots, N-1$ the cross-PSD has to be estimated in order to estimate the delay. This can be done using



(a) Normalized cross-correlation of white noise as a function of delay.



(b) Normalized cross-correlation for a speech signal as a function of delay.

Figure 3: Normalized cross-correlation for different signals.

the cross-periodogram, which is defined as

$$I_{X_1X_2}(e^{j\omega}) = \frac{1}{N}\left(\sum_{n=0}^{N-1} X_1(n)e^{-j\omega n}\right)\left(\sum_{n=0}^{N-1} X_2(n)e^{-j\omega n}\right)^* .$$

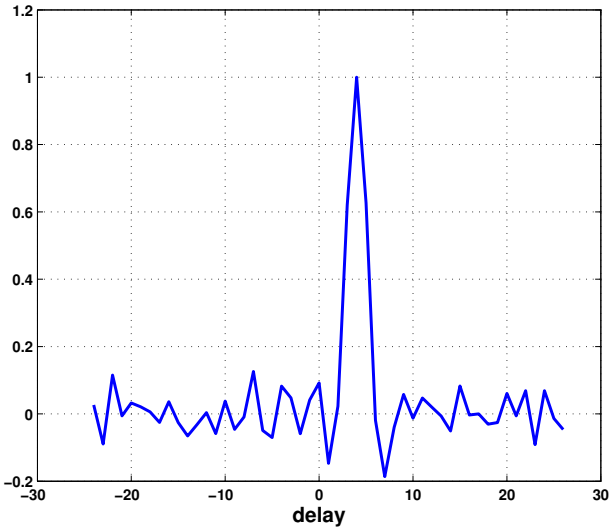


Figure 4: Normalized generalized cross-correlation for a speech signal as a function of delay.

If the source location is allowed to vary during the experiment, it is appropriate to use only short sequences (N is small), and estimate the GCC and the corresponding AOA for each sequence. Additionally, since the source movement is likely to be slow, the estimation can be enhanced by averaging consecutive cross-periodograms in an adaptive fashion, with forgetting factor α, i.e. for the Mth sequence we calculate

$$I_{X_1X_2}(e^{j\omega}, M) = \alpha I_{X_1X_2}(e^{j\omega}, M-1) + (1-\alpha) I_{X_1X_2}(e^{j\omega}, M). \tag{8}$$

When using the PHAT weighting function, the estimate of the GCC is then calculated for each sequence M by

$$\hat{g}_{X_1X_2}(\kappa, M) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{I_{X_1X_2}(e^{j\omega}, M)}{|I_{X_1X_2}(e^{j\omega}, M)|} e^{j\omega\kappa} d\omega. \tag{9}$$

As in Equation (7), the normalized time-delay $\Delta t/T$ for the Mth sequence is then estimated by the lag variable κ which maximizes $\hat{g}_{X_1X_2}(\kappa, M)$. *An interpolation of the GCC estimate before the maximum search can improve the accuracy of the estimate.*

2.3 The Fusion of AOA Measurements

Using multiple microphone pairs at known locations, multiple estimates of the AOA can be calculated. The location of the source is determined by fusing these AOA estimates. Figure 5 shows a scenario where $K = 4$ pairs of microphones are used. The estimation of AOA is carried out in a decentralized manner at each pair. Thus, 4 estimates of the AOA is available and can be used to localize the sound source.

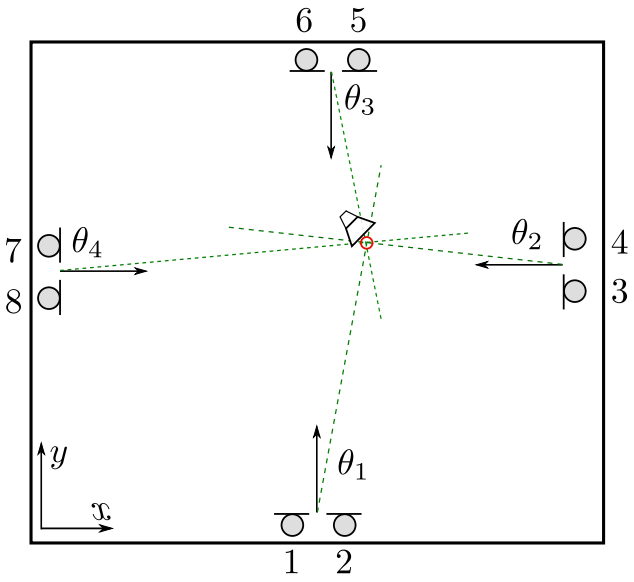


Figure 5: Sound source localization using 4 microphone pairs.

Let the center of the kth pair be denoted as $P_k = [x_k, y_k]^T$. Since this location is known, using the AOA estimate at the kth pair $\hat{\theta}_k$, the location of the source is restrained to the line

$$\begin{aligned} y = &\quad \tan\hat{\theta}_k \ (x - x_k) + y_k \\ = &\quad \tan\hat{\theta}_k \ x + \underbrace{y_k - \tan\hat{\theta}_k \ x_k}_{\hat{b}_k}. \end{aligned} \tag{10}$$

Note that the AOAs estimated at different pairs should be transformed to a global coordinate system before they can be used to localize the sound source. Since only the location of the source is not known in Equation (10), the AOA estimates of two different pairs[1] is sufficient to localize the source using

[1]Except when the source is located on the line connecting the two pairs.

line intersection. However, using AOA estimates of more than two pairs can increase the accuracy of the estimated location. Note that in the presence of the noise, you can not use simple line intersection since the intersection of any two lines could differ from the others, this is shown in Figure 6. Thus, another criterion should be used to extract the location from the K AOA estimates.

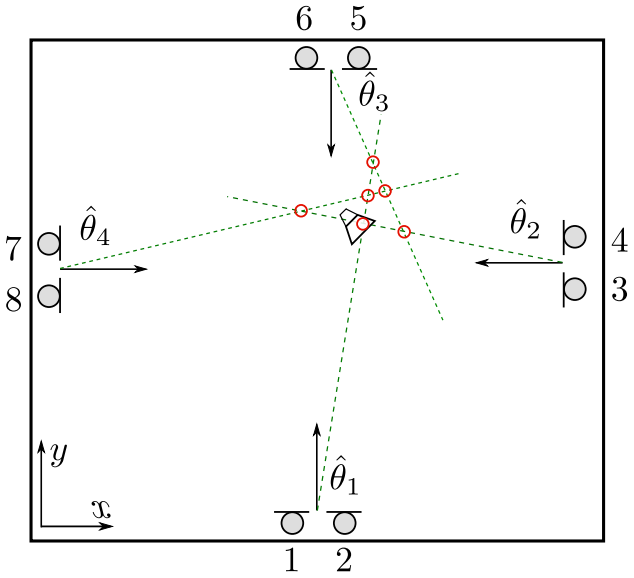


Figure 6: Intersection of lines estimated at different pairs.

A simple approach is to find the location which has the minimum Euclidean distance from the K lines estimated at the K different pairs. Let $\hat{\boldsymbol{p}^*} = [\hat{x^*}, \hat{y^*}]^T$ denote the estimate of the source location. Then, the distance from this estimate to the line defined in Equation (10) is

$$d_k^* = \left| \frac{\tan\hat{\theta}_k}{\sqrt{\tan^2\hat{\theta}_k + 1}}\hat{x^*} - \frac{1}{\sqrt{\tan^2\hat{\theta}_k + 1}}\hat{y^*} + \frac{1}{\sqrt{\tan^2\hat{\theta}_k + 1}}\hat{b}_k \right|. \tag{11}$$

The least squares estimate of the location which minimizes the squares of $\boldsymbol{d}^* = [d_1^*, \ldots, d_K^*]^T$ can be found to be

$$\hat{\boldsymbol{p}^*} = -\left(\hat{\boldsymbol{A}}^T\hat{\boldsymbol{A}}\right)^{-1}\hat{\boldsymbol{A}}^T\hat{\boldsymbol{b}}, \tag{12}$$

where

$$\hat{\boldsymbol{A}} = \begin{pmatrix} \frac{\tan\hat{\theta}_1}{\sqrt{\tan^2\hat{\theta}_1+1}} & \frac{-1}{\sqrt{\tan^2\hat{\theta}_1+1}} \\ \vdots & \vdots \\ \frac{\tan\hat{\theta}_K}{\sqrt{\tan^2\hat{\theta}_K+1}} & \frac{-1}{\sqrt{\tan^2\hat{\theta}_K+1}} \end{pmatrix} \tag{13}$$

and

$$\hat{\boldsymbol{b}} = \begin{pmatrix} \frac{\hat{b}_1}{\sqrt{\tan^2\hat{\theta}_1+1}} \\ \vdots \\ \frac{\hat{b}_K}{\sqrt{\tan^2\hat{\theta}_K+1}} \end{pmatrix}. \tag{14}$$

3 Preparation

1. Provide a formula for the maximum possible time-delay between two microphones spaced by d. What is $\Delta n_{\max}$ for $d = 0.175$ m and $f_s = 8$ kHz?

2. Which problem occurs generally when the time-delay is calculated in the time domain for sampled signals? How can it be fixed?

3. For now assume $\Delta t/T$ to be an integer. Derive the result of (6).

4. How can the cross-SOMF (cross-correlation function) be estimated using the cross-periodogram?

5. Use Equation (11) to write $\boldsymbol{d}^*$ using $\hat{\boldsymbol{A}}, \hat{\boldsymbol{b}}$ and $\hat{\boldsymbol{p}}^*$, then derive the least squares estimate which is given in Equation (12).

4 Experiment

In the following experiments, you will first learn how to acquire measurements from distributed microphone pairs. Then, these measurements are used to implement and test the concepts from the previous sections.

4.1 Setup and Data Acquisition

Figure 7 depicts the setup of the experiment, where $K = 4$ pairs of microphones are used to record sound signals. The centers of the four pairs are

$\boldsymbol{p}_1 = (2.69, 0.19), \boldsymbol{p}_2 = (5.19, 3.21), \boldsymbol{p}_3 = (2.70, 5.27)$ and $\boldsymbol{p}_4 = (0.37, 3.54)$, measured in meters. The separation distance between the pairs is $d = 0.175$ meters.

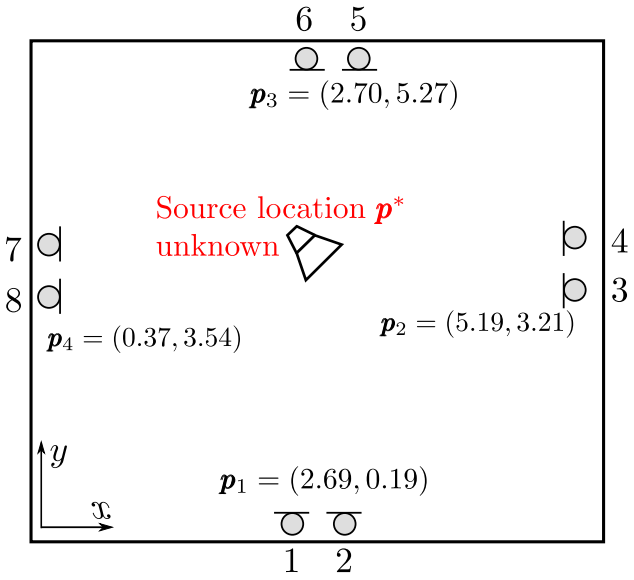


Figure 7: Experiment setup.

Signals from the 8 microphones can be recorded using the MATLAB© command:

```
x=nktp_rec(SAMPLES, FREQ)
```

where the command requires two parameters, i.e., the number of samples to acquire and the sampling frequency in Hertz. The output parameter `x` is a SAMPLES $\times 8$ matrix whose columns is the sampled output of the 8 microphones.

Another MATLAB© routine is available to simulate the output of the microphones. This routine gives more control over the parameters of the experiment and allows you to test your functions before using the real data. Simulation data can be generating using the routine

```
x=nktp_sim(FREQ, SPEECH, LOC, G, SNR, NO_NOISE )
```

where the following parameters should be provided:

1. The sampling frequency in Hertz.

2. The signal of the source. Note that wave files can be loaded in MATLAB© using the command `wavread` which takes the name of the wave file as a parameter. You can also use white Gaussian noise as a test signal using the command `randn`.

3. The true source location $\boldsymbol{p}^*$.

4. The gain of the 8 microphones.

5. The signal to noise ratios at the 8 microphones in dB.

6. 0 or 1 indicating the presence or the absence of additive noise, respectively.

For example, the following code:

```
x=nktp_sim(48000,randn(1024,1),[2,3],ones(1,8),-5*ones(1,8),0);
```

simulates the output of the 8 microphones, where the sound source emits a white Gaussian signal of length 1024 and is located at $\boldsymbol{p}^* = (2, 3)$. The sampling frequency is 48,000 Hertz and all the microphones have unitary gain, and the SNR at all the microphones is -5dB.

4.2 Time-Delay Estimation Using the Cross-Correlation Function

We first want to determine the AOA of the first microphone pair. Proceed as follows:

1. Use `nktp_sim` to simulate a stationary source, and extract the signal of the first and second microphones `X1=x(1,:)` and `X2=x(2,:)`. Use the signal length of 1024 and sampling frequency of 48,000, and a white Gaussian signal as source signal.

2. Calculate $\Delta n_{\max}$.

3. Write a MATLAB© function `xycorr` which calculates the cross-correlation function between two signals X and Y for $n \in \{-\Delta n_{\max}, -\Delta n_{\max} + 1, \ldots \Delta n_{\max}\}$, using the cross-periodogram. Plot the output as a function of n.

4. Use the MATLAB© function `max` to find the maximum and determine Δn. Also estimate θ using the command `asin`.

5. Repeat the above steps using real data. Compare the cross-correlation function of the real speech signal and white noise.

4.3 Improvements by Using the GCC

Write a function `genxcorr` which calculates the GCC of two input vectors. The function `gencorr` should return the GCC for $n \in \{-\Delta n_{\max}, -\Delta n_{\max} + 1, \ldots \Delta n_{\max}\}$. Furthermore, we modify (9) to

$$\hat{g}_{X_1X_2}(\kappa) = \text{IFFT}\left\{\frac{I_{X_1X_2}(e^{j\omega_k})}{|I_{X_1X_2}(e^{j\omega_k})| + \epsilon}\right\}$$

with a small ϵ to avoid zero division, e.g. 10^{-7}. After calculating the FFT for each vector, form the cross-periodogram and weighting function, then calculate the GCC as given above. Consider only the real part after the IFFT, since the imaginary part is non-zero only due to numerical errors. For extracting the relevant part, you can use the command `ifftshift`, which transforms the "zero lag" to MATLAB© index $N/2 + 1$.

Compare `xycorr` and `genxcorr` using real speech data, i.e. plot the output of the two functions and comment on the results.

4.4 Source Localization

Now, you will use the AOA estimates from all microphone pairs to localize the sound source, as shown in the room setup of Figure 7. Proceed as follows:

1. Use `nktp_sim` to simulate a stationary sound source as before.

2. Extract the output signal of the first pair and compute the delay and then the AOA estimate $\hat{\theta}_1$.

3. Repeat Step 2 for the remaining pairs and compute $\hat{\theta}_2$, $\hat{\theta}_3$ and $\hat{\theta}_4$.

4. Use Equation (12) to estimate the location of the source. Note that the AOA estimates should first be converted to the global coordinate system.

5. Plot the true and the estimated source location and the position of the four pairs.

6. Repeat the above steps using the real data.

Note that, using real data, the estimate can be much worse than the simulated data, since the simulation does not consider reflections from the walls and the floor and interferences from other sound sources.

4.5 Extensions for a Moving Source

In this section, you will use your code from the previous section for a moving source. Use the real data directly and processed as follows:

1. Use the function `nktp_rec` to acquire the output of the microphones for a short time, i.e., small frame. Thus, the source can be considered stationary during the acquisition time.

2. Use the functions that you wrote in the previous section to estimate the location of the source.

3. Plot the estimated source location and the position of the four pairs.

4. Repeat Steps 1, 2 and 3, and test the localization for at least 10 seconds.

Figure 8 shows the result for simulated data. In Figure 8, the true track of the sound source is also plotted as a black solid line.

Using real data, the estimated location will contain more errors, since the simulation does not account for reflections from the walls and the floor and interferences from other sound sources.

The slow variation of the spatial parameters can be exploited by averaging consecutive cross-periodograms, as in Equation (8). Extend your function `genxcorr` with two input parameters: the last cross-periodogram and the forgetting factor α. Use this function and repeat the above tests. Compare your results when using, e.g. $\alpha = 0.5$.

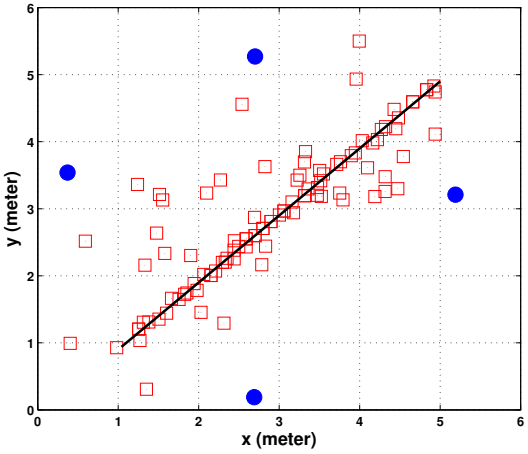


Figure 8: Localization of a moving source.