LAS Report

TECHNISCHE UNIVERSITÄT DARMSTADT

Problem 1 Preparation

Problem 1.1 Provide a formula for the maximum possible time-delay between two microphones spaced by d. What is $\Delta n_{\max}$ for $d = 0.175\,\mathrm{m}$ and $f_s = 8\,\mathrm{kHz}$?

$\Delta n_{\max} = \mathrm{round}\left(f_s \cdot \frac{d}{v}\right) = 4$

Problem 1.2 Which problem occurs generally when the time-delay is calculated in the time domain for sampled signals? How can it be fixed?

The time delay is not necessarily an integer multiple of the sampling interval. Interpolation of the estimated cross-correlation function may lead to better results when searching for the maximum.

Problem 1.3 For now assume $\Delta t/T$ to be an integer. Derive the result of (6).

The prime $'$ denotes the complex conjugate.

Goal:

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

Knowns:

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

The left equation is the identity under Fourier transform between CPSD and cross-SOMF

$$P_{X_1X_2}(e^{j\omega}) = \sum_{\kappa=-\infty}^{\infty} r_{X_1X_2}(\kappa)e^{-j\omega\kappa}$$

The outer equation is to show

$$\begin{aligned}
P_{X_1X_2}(e^{j\omega}) &= gP_{SS}(e^{j\omega})e^{j\omega\Delta t/T} \\
P_{X_1X_2}(e^{j\omega}) &= \mathrm{E}\{X_1'(e^{j\omega})X_2(e^{j\omega})\} \\
&= \mathrm{E}\{(S'(e^{j\omega}) + N_1'(e^{j\omega}))(gS(e^{j\omega})e^{-j\omega\Delta t/T} + N_2(e^{j\omega}))\} \\
&= \mathrm{E}\{gS'(e^{j\omega})S(e^{j\omega})e^{-j\omega\Delta t/T} + N_1'(e^{j\omega})gS(e^{j\omega})e^{-j\omega\Delta t/T} + N_1'(e^{j\omega})N_2(e^{j\omega}) + S'(e^{j\omega})N_2(e^{j\omega})\} \\
&= g\mathrm{E}\{S'(e^{j\omega})S(e^{j\omega})\}e^{-j\omega\Delta t/T} \\
&= gP_{SS}(e^{j\omega})e^{-j\omega\Delta t/T}
\end{aligned}$$

Problem 1.4 How can the cross-SOMF (cross-correlation function) be estimated using the cross-periodogram?

The cross-periodogram is an estimator for the cross-PSD, and therefore the IDFT of the cross-periodogram is an estimator for the cross-SOMF due to the identity

$$P_{X_1X_2}(e^{j\omega}) = \sum_{\kappa=-\infty}^{\infty} r_{X_1X_2}(\kappa)e^{-j\omega\kappa}$$

Problem 1.5 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).

The prime $'$ denotes the complex conjugate, the ^{H} the complex conjugate transpose. The ^{H} reduces to ^{T} for real valued $\hat{\boldsymbol{A}}$.

$$\begin{aligned}
d_k^* &= \left| \frac{\tan\hat{\theta}_k}{\sqrt{\tan^2\hat{\theta}_k + 1}}\hat{x}^* - \frac{1}{\sqrt{\tan^2\hat{\theta}_k + 1}}\hat{y}^* + \frac{1}{\sqrt{\tan^2\hat{\theta}_k + 1}}\hat{b}_k \right| \\
\Rightarrow \boldsymbol{d}^* &= \left|\hat{\boldsymbol{A}}\hat{\boldsymbol{p}}^* + \hat{\boldsymbol{b}}\right| \\
\boldsymbol{d}^{*2} &= (\hat{\boldsymbol{A}}\hat{\boldsymbol{p}}^* + \hat{\boldsymbol{b}})^H(\hat{\boldsymbol{A}}\hat{\boldsymbol{p}}^* + \hat{\boldsymbol{b}}) \\
&= \hat{\boldsymbol{p}}^{*H}\hat{\boldsymbol{A}}^H\hat{\boldsymbol{A}}\hat{\boldsymbol{p}}^* + \hat{\boldsymbol{b}}^H\hat{\boldsymbol{b}} + \hat{\boldsymbol{p}}^{*H}\hat{\boldsymbol{A}}^H\hat{\boldsymbol{b}} + \hat{\boldsymbol{b}}^H\hat{\boldsymbol{A}}\hat{\boldsymbol{p}}^*
\end{aligned}$$

Differentiating $\boldsymbol{d}^{*2}$ w.r.t. $\hat{\boldsymbol{p}}^*$:

$$\begin{aligned}
\frac{\partial \boldsymbol{d}^{*2}}{\partial \hat{\boldsymbol{p}}^*} &= \frac{\partial \hat{\boldsymbol{p}}^{*H}\hat{\boldsymbol{A}}^H\hat{\boldsymbol{A}}\hat{\boldsymbol{p}}^* + \hat{\boldsymbol{b}}^H\hat{\boldsymbol{b}} + \hat{\boldsymbol{p}}^{*H}\hat{\boldsymbol{A}}^H\hat{\boldsymbol{b}} + \hat{\boldsymbol{b}}^H\hat{\boldsymbol{A}}\hat{\boldsymbol{p}}^*}{\partial \hat{\boldsymbol{p}}^*} \\
&= 2\hat{\boldsymbol{A}}^H\hat{\boldsymbol{A}}\hat{\boldsymbol{p}}^* + 2\hat{\boldsymbol{A}}^H\hat{\boldsymbol{b}}
\end{aligned}$$

Setting the derivative to zero for finding the extrema yields:

$$\hat{\boldsymbol{A}}^H\hat{\boldsymbol{A}}\hat{\boldsymbol{p}}^* = -\hat{\boldsymbol{A}}^H\hat{\boldsymbol{b}}$$

Multiplying with from the left results in the LSE solution for non-singular $\hat{\boldsymbol{A}}^H\hat{\boldsymbol{A}}$:

$$\hat{\boldsymbol{p}}^* = -(\hat{\boldsymbol{A}}^H\hat{\boldsymbol{A}})^{-1}\hat{\boldsymbol{A}}^H\hat{\boldsymbol{b}}$$

The solution is a minimum, if the Hesse matrix is positive definite. As $\hat{\boldsymbol{A}}^H\hat{\boldsymbol{A}}$ is positive semi-definite by structure and non-singular, the Hesse matrix is positive definite:

$$\frac{\partial^2 \boldsymbol{d}^{*2}}{\partial \hat{\boldsymbol{p}}^{*2}} = 2\hat{\boldsymbol{A}}^H\hat{\boldsymbol{A}}$$

Problem 2 Experiment

Problem 2.1 Setup and Data Acquisition

Problem 2.1.1 Time-Delay Estimation Using the Cross-Correlation Function

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.

```
% white Gaussian signal
y = randn(1024,1);
% OR real data
y = audioread('beepbeep.wav');
% sampling frequency in hertz
f=48000;
% microphone distance in meter
d = 0.175;
% speed of sound in meter/second
v = 343;
% maximum TDOA in samples
dn_max = round(f*d/v);
% simulated recording
x=nktp_sim(f,y,[2, 3],ones(1,8),-22*ones(1,8),0);
% signal of the first array
X1 = x(:,1);
X2 = x(:,2);
% size of the data
N = size(x,1);
% calculate angle of arrival
theta = AOA(X1, X2, dn_max, N);

%% plotting
% calculate cross-correlation
rxy = xycorr(X1, X2, N);
% shift zero lag to the middle
rxy = ifftshift(rxy);
% take middle part out, with index correction
rxy = rxy(ceil(N/2)-dn_max+1:ceil(N/2)+dn_max+1);
% time axis in relevantrange
t = -dn_max:dn_max;
% plot center part of the cross correlation
plot(t, rxy)
% xlabel('$n$','Interpreter','LaTex')
% ylabel('$\hat{r}_{xy}$','Interpreter','LaTex')
% matlab2tikz('beepbeep.tikz');
```

Calculate $\Delta n_{\max}$.

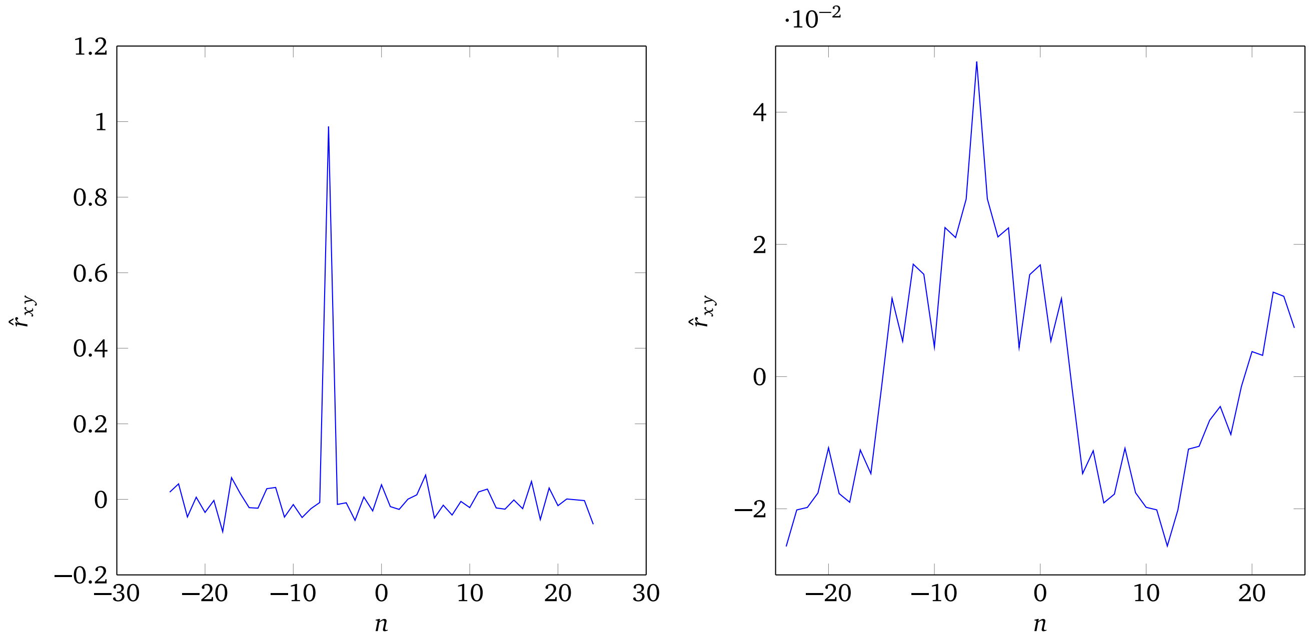
$\Delta n_{\max} = 24$

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

```
function rxy = xycorr(x, y, N)
% calculate FFTs
X = fft(x(1:N));
Y = fft(y(1:N));
% calculate periodogram
I = X.*conj(Y)/N;
% calculate IFFT and cut off imaginary part
rxy = real(ifft(I));
```

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

The peak of the cross-correlation of white noise is narrower and the second-highest value is much closer to the peak value in the real-signal case.



(a) CC of white Gaussian

(b) CC of real signal

Figure 1: Cross-correlation

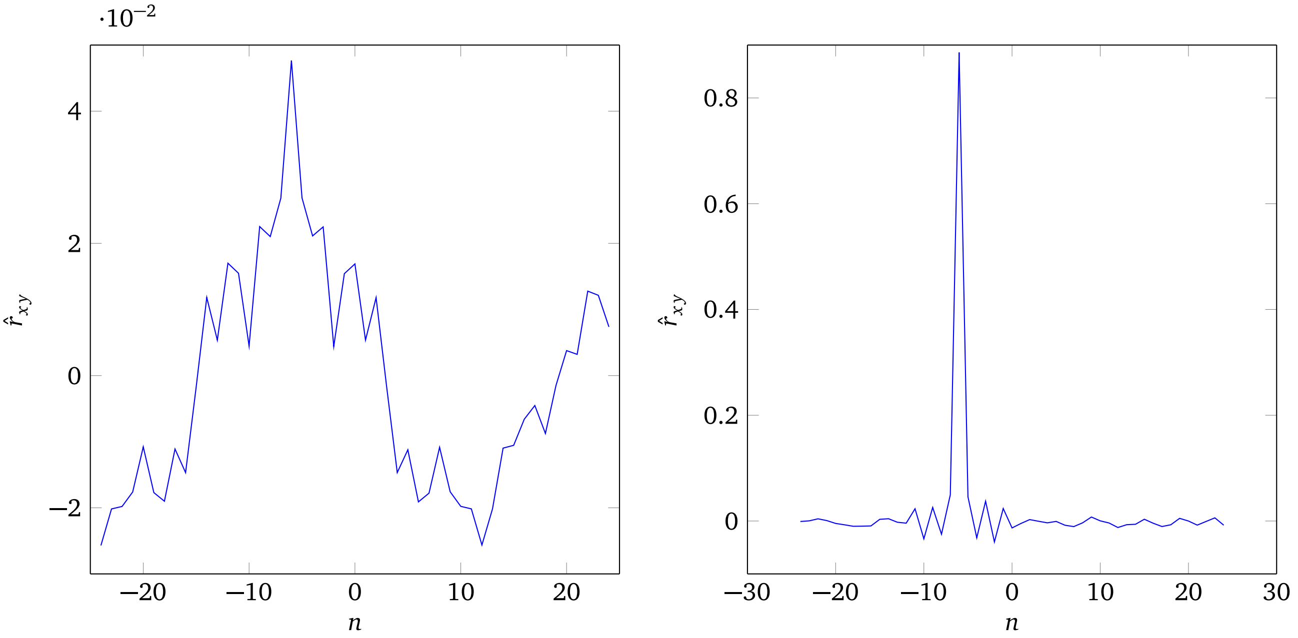
Problem 2.1.2 Improvements by Using the GCC

Write a function genxcorr which calculates the GCC of two input vectors.

```
function [rxy, I] = genxcorr(x, y, N)

eps = 1e-7;
% calculate FFTs
X = fft(x(1:N));
Y = fft(y(1:N));
% calculate periodogram
I = X.*conj(Y);
% normalize periodogram
I = I./(abs(I)+eps);
% calculate IFFT and cut off imaginary part
rxy = real(ifft(I));
```

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

The peak shows at the same position, but is much narrower for the GCC compared to the normal cross-correlation. The output of the GCC is similar to the cross-correlation of white noise signals.



(a) CC of real signal

(b) GCC of real signal

Figure 2: Generalized cross-correlation

Problem 2.1.3 Source Localization

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 $\hat{\theta}_1$.

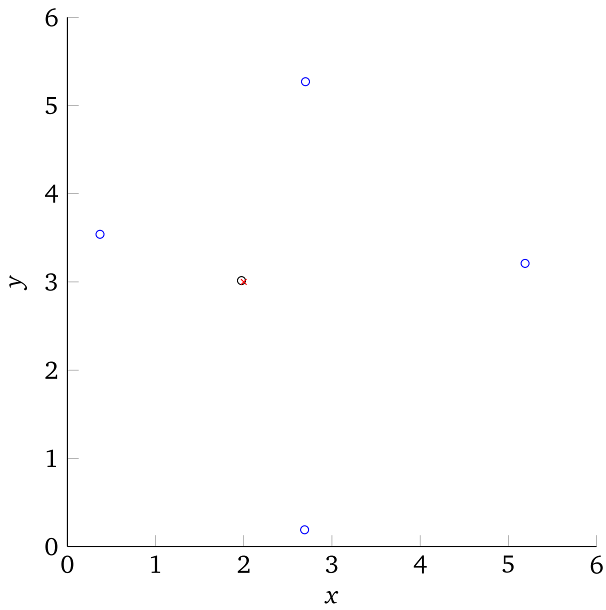
3. Repeat Step 2 for the remaining pairs and compute $\hat{\theta}_2$, $\hat{\theta}_3$ and $\hat{\theta}_4$.

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.

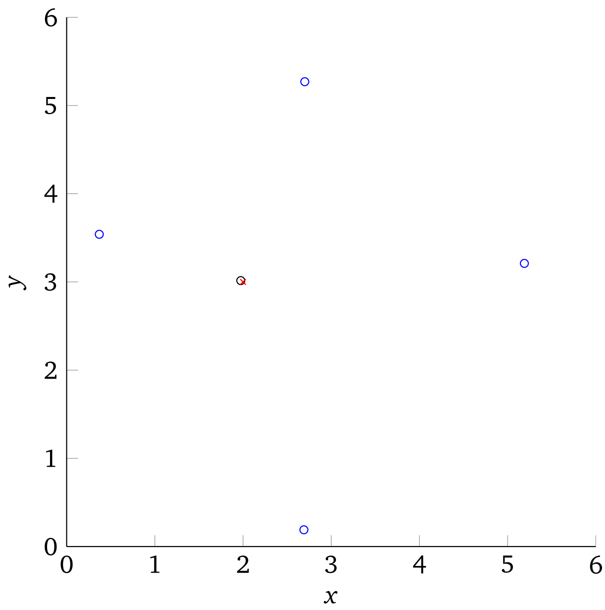
```
% white Gaussian signal
y = randn(1024,1);
% OR real data
y = audioread('beepbeep.wav');
% sampling frequency in hertz
f=48000;
```

```
% microphone distance in meter
d = 0.175;
% speed of sound in meter/second
v = 343;
% maximum TDOA in samples
dn_max = round(f*d/v);
% simulated recording
x=nktp_sim(f,y,[2, 3],ones(1,8),5*ones(1,8),0);
% size of the data
N = size(x,1);
% array positions in meter
arrays = [2.69, 0.19; 5.19, 3.21; 2.70, 5.27; 0.37, 3.54];

p_hat = estimate_position(x, dn_max, N, arrays);

figure; hold on
scatter(p_hat(1), p_hat(2), 'ok');
scatter(2, 3, 'xr');
scatter(arrays(:,1), arrays(:,2), 'ob');

% xlabel('$x$','Interpreter','LaTex')
% ylabel('$y$','Interpreter','LaTex')
% matlab2tikz('position_sim.tikz');
```

Plot the true and the estimated source location and the position of the four pairs.
Repeat the above steps using the real data.



(a) White Gaussian signal



(b) Real signal

Figure 3: Position estimate of simulated audio sources

Problem 2.1.4 Extensions for a Moving Source

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.

The recorded signals were too noisy for reliable estimates.

```
% capture time in seconds
T = 10;
% sampling frequency in hertz
```

```
f=48000;
% record data
x=nktp_rec(T*f, f);
% size of the data
N = f*T;
% microphone distance in meter
d = 0.175;
% speed of sound in meter/second
v = 343;
% maximum TDOA in samples
dn_max = round(f*d/v);
% array positions in meter
arrays = [2.69, 0.19; 5.19, 3.21; 2.70, 5.27; 0.37, 3.54];
% number of estimates to calculate, should be an integer fraction of N
M = 100;
% memory allocation
p_hat = zeros(2,M);
for t = 1:M
    x_t = x((t-1)*N/M+1:t*N/M,:);
    p_hat(:,t)  = estimate_position(x_t, dn_max, N, arrays);
end

%scatter plot to visualize movement
scatter(p_hat(1,:), p_hat(2,:))
hold on
scatter(arrays(1,:), arrays(2,:), 'rx')
```

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. $\alpha = 0.5$.

```
function [rxy, I] = genxcorr(x, y, N, I_old, alpha)

eps = 1e-7;
% calculate FFTs
X = fft(x(1:N));
Y = fft(y(1:N));
% calculate periodogram
I = X.*conj(Y);
% normalize periodogram
I = I./(abs(I)+eps);
% smooth
I = alpha*I_old+(1-alpha)*I;
% calculate IFFT and cut off imaginary part
rxy = real(ifft(I));
```