Delft University of Technology
Faculty of Electrical Engineering, Mathematics, and Computer Science
Circuits and Systems Group

ET4 147: Signal Processing for Communications

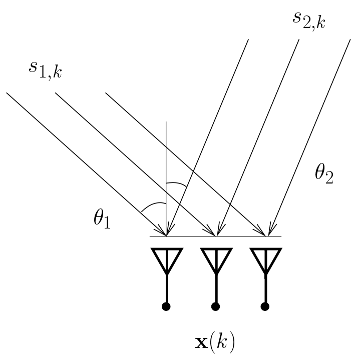
Spring 2020

EXAM ASSIGNMENT

The exam consists of two parts: (1) estimating directions and frequencies of sinusoidal signals entering a uniform linear array, and (2) equalizing a signal received over a coded channel. Make a short report containing the required Matlab files, plots, and answers, and bring it with you when you have your exam. The report can be made alone or in groups of two. The layout of the report is not important, only the content is. The report will be discussed during an oral online exam of about 45 minutes per student. During this exam I will ask you to explain your programs, results, and answers. The most important thing is that you understand WHY certain steps are taken. To further test your understanding, I may also ask you about the applicability of some of the studied algorithms to related problems. Sufficient time will be provided to express yourself, in case you have problems with the English language. If you have never taken an oral exam before and/or do not feel comfortable with taking one, please organize a trial exam with one of your fellow students.

In case something is unclear, (limited) assistance is given by Pim van der Meulen, e-mail: `P.Q.vanderMeulen@tudelft.nl`.

ESTIMATION OF DIRECTIONS AND FREQUENCIES



We consider the case of d complex sinusoidal signals impinging on a uniform linear array of M antennas. We want to estimate their directions and frequencies.

Signal model

1. Make a data generator function

```
function [X,A,S] = gendata(M,N,Delta,theta,f,SNR)
```

   to generate the received signal matrix $\mathbf{X} : M \times N$, where M represents the number of antennas, and N the number of samples. The antenna array is a uniform linear array with $\Delta = 1/2$ wavelengths antenna spacing. Further, $\boldsymbol{\theta} = [\theta_1, \cdots, \theta_d]^{\mathrm{T}}$ contains the directions of the sources in degrees, $-90 \leq \theta_i < 90$, and $\mathbf{f} = [f_1, \cdots, f_d]^{\mathrm{T}}$ the normalized frequencies of the sources, $0 \leq f_i < 1$. Finally, SNR denotes the signal-to-noise ratio (SNR) **per source** (this means it is the ratio of the signal power of one source over the noise power). We assume for simplicity that the noise is temporally and spatially white, complex Gaussian. The ith source is a complex sinusoid with normalized frequency f_i, given by:

$$s_{i,k} = \exp(j2\pi f_i k).$$

   The noise is scaled to obtain the desired SNR per source.

2. Make plots of the singular values of $\mathbf{X}$, for $d = 2$ sources, $M = 5$, $N = 20$, $\boldsymbol{\theta} = [-20, 30]^{\mathrm{T}}$, $\mathbf{f} = [0.1, 0.3]^{\mathrm{T}}$, and $SNR = 20$ dB. What happens to the singular values if the number of samples doubles, if the number of antennas doubles, if the angles between the sources becomes small, if the frequency difference becomes small? Why?

Estimation of directions

1. Implement the ESPRIT algorithm to estimate the angles of arrival from a data matrix $\mathbf{X}$, assuming the number of sources is known

```
function theta = esprit(X,d)
```

2. Check the correctness in case there is no noise (SNR very high). The angles of arrival should then be perfectly estimated (up to an order ambiguity).

Estimation of frequencies

1. For the model $\mathbf{X} = \mathbf{AS}$, is there any special structure present in the $\mathbf{S}$ matrix?

2. Modify the ESPRIT algorithm to estimate the frequencies directly from $\mathbf{X}$, using the structure that is present in $\mathbf{S}$:

```
function f = espritfreq(X,d)
```

3. Check the correctness in case there is no noise (SNR very high). The frequencies should then be perfectly estimated (up to an order ambiguity).

Comparison

1. Make a plot of the estimation performance of the two algorithms (mean values and standard deviations of the estimated angles and frequencies) as a function of the SNR ($0, 4, 8, \cdots, 20$ dB), for $d = 2$ sources, $M = 3$, $N = 20$, $\boldsymbol{\theta} = [-20, 30]$, $\mathbf{f} = [0.1, 0.12]$. To compute the mean value and the standard deviation, take the statistics over 1000 test runs with independent complex Gaussian noise.

2. Compute two zero-forcing beamformers, one based on the direction estimates from 'esprit', and one based on the frequency estimates from 'espritfreq'. Hence, you do NOT have to use any intermediate result from the ESPRIT algorithm. Test the correctness of the two beamformers in case there is no noise. In that case $\mathbf{W}^{\mathrm{H}}\mathbf{X}$ should perfectly recover the transmitted sources (up to an order ambiguity).

3. Compare the two above zero-forcing beamformers by plotting their spatial responses $y(\theta) = |\mathbf{w}^{\mathrm{H}}\mathbf{a}(\theta)|$ for $-90° \leq \theta \leq 90°$. Use the same parameters as before, and take $SNR = 10$ dB. Which one gives the best ~~suppression of the interference?~~ Why?

CMA algorithm

1. Implement the CMA algorithm in the function

```
function [w,y] = cma(X,mu,w_init)
```

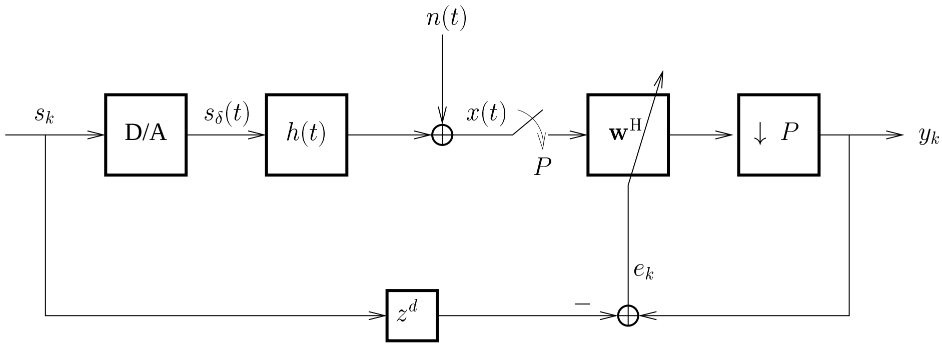
   where $\mathbf{X}$ is the data matrix generated by 'gendata', μ is the step size, and $\mathbf{w}_{init}$ is the initial beamformer. Further, $\mathbf{w}$ is the computed final beamformer and $\mathbf{y}$ is the sequence of estimated signal samples during the iterations.

2. Verify the correctness of the algorithm. Assume there is no noise, and take $N = 5000$, $\mu = 0.001$, and $\mathbf{w}_{init} = [1 \;\; \cdots \;\; 1]^T$. Plot $|\mathbf{y}|$ to see if the algorithm converges.

3. Plot the spatial response of the final beamformer. What is the difference with the previous beamformers?

4. Make plots of the spatial response of the CMA beamformer for several random initial weight vectors. Is always the same signal recovered?

CHANNEL EQUALIZATION



Here we consider the scenario where a single source is transmitting data towards a single receive antenna over a specific channel. The received signal $x(t)$ can be modeled as

$$x(t) = \sum_k h(t-k)s_k + n(t).$$

We assume the channel $h(t)$, whch includes the transmit and receive filter effects, is some kind of a coded channel and looks like

$$h(t) = \begin{cases} 1 & \text{for } t \in [0, 0.25) \\ -1 & \text{for } t \in [0.25, 0.5) \\ 1 & \text{for } t \in [0.5, 0.75) \\ -1 & \text{for } t \in [0.75, 1) \end{cases}$$

Furthermore, the data symbols s_k are assumed to be QPSK data symbols, i.e., $s_k \in \{\pm 1/\sqrt{2} \pm j/\sqrt{2}\}$. The additive noise $n(t)$ is finally modeled as complex zero-mean Gaussian with standard deviation σ. The goal now is to reconstruct the transmitted data symbols by linear equalization.

Signal model

1. For a given data symbol sequence $\mathbf{s} = [s_0\ s_1\ \cdots\ s_{N-1}]^{\mathrm{T}}$, construct the sequence $\mathbf{x} = [x(0)\ x(1/P)\ \cdots\ x(N-1/P)]^{\mathrm{T}}$, which is obtained by sampling the received signal $x(t)$ at rate $1/P$, i.e., the sampling rate is P times larger than the data rate:

```
function x = gendata_conv(s,P,N,sigma)
```

2. Take $P = 4$ (i.e., we sample at chip rate), $N = 500$, and $\sigma = 0$ and construct the data matrix

$$\mathcal{X} = \begin{bmatrix} x(0) & x(1) & \cdots & x(N-2) \\ x(\frac{1}{P}) & x(1+\frac{1}{P}) & \cdots & x(N-2+\frac{1}{P}) \\ \vdots & \vdots & & \vdots \\ x(1+\frac{P-1}{P}) & x(2+\frac{P-1}{P}) & \cdots & x(N-1+\frac{P-1}{P}) \end{bmatrix} : 2P \times (N-1) \qquad (1)$$

Note that the rows of this matrix $\mathcal{X}$ span two data symbol periods. What is the rank of $\mathcal{X}$ and why?

3. What if we double P? How does this change the rank of $\mathcal{X}$ and why?

Zero-forcing and Wiener equalizer

1. Take again $P = 4$ and $N = 500$, but add some noise with standard deviation $\sigma = 0.5$. Compute the zero-forcing and Wiener receiver for the data matrix $\mathcal{X}$ constructed in (1), assuming the channel $h(t)$ and the standard deviation of the noise σ are known. What is a good delay for these receivers, i.e., on which shift of the data symbol sequence should these receivers focus? For this delay, plot the estimated symbols of the two equalizers in the complex plane such that you observe four clusters. Is there a large difference in performance between the two receivers? Explain.

2. Take again $N = 500$ and $\sigma = 0.5$, but double P. Plot again the estimated symbols of the zero-forcing and Wiener equalizer in the complex plain. Consider the same delay as the one you selected above. Compare the plots and explain the differences.

Channel estimation

1. Suppose now that $h(t)$ is unknown and has to be estimated from the received sequence $\mathbf{x}$ and the data symbol sequence $\mathbf{s}$. Also assume that the channel length in symbol periods, L, is not known. For this scenario, implement the following channel estimator:

```
function h = channel_estimator(x,s,L)
```

2. Take $P = 4$, $N = 500$, and $\sigma = 0.5$, and estimate the channel for $L = 1$, $L = 2$, and $L = 3$. Compare the real and imaginary part of the estimated channel with the true channel in a plot. What can you observe? Is there any performance difference between $L = 1$, $L = 2$, and $L = 3$? What is the true L?

Blind spatial processing

1. Take again $P = 4$, $N = 500$, and $\sigma = 0.5$, and compute the SVD of $\mathcal{X}$ in (1). Determine the null-subspaces of the columns and the rows of $\mathcal{X}$, assuming that the true channel length in symbol periods, L, is known.

2. Estimate the channel $\mathbf{h}$ from the column null-subspace $\mathbf{U}_n$ of $\mathcal{X}$. Derive the Matlab function

```
function h = blind_channel(X)
```

3. Compare the real and imaginary part of the estimated channel with the true channel in a plot. Do they match? Why or why not?

4. Estimate the symbols s_k from the row null-subspace $\mathbf{V}_n$ of $\mathcal{X}$. Derive the Matlab function

```
function s = blind_symbol(X)
```

   Compare the estimated symbols with the true symbols in the complex plane. Do they match? Why or why not?