Adaptive Array Signal Processing

[5SSC0]

Assignment Part 1C: Adaptive array processing

REPORT

Group number: 3

Names including ID:
1: Floris Naber 1253514
2: Shao Hsuan Hung 1723219

Date: 18/03/2023

3.2.1 Assignment 1C: Scenario 1

a

We assume that the sources are narrow-band and so far away that we consider far field conditions. Given that we have a uniform linear are (ULA), then we can define the steering vector $\underline{\mathbf{a}}(\theta_{s_i})$ for signal $s_i[k]$ of a given source i in frequency domain by:

$$\begin{aligned}\underline{\mathbf{a}}(\theta_{s_i}) &= \begin{bmatrix} a_1(\theta_{s_i}) & \cdots & a_J(\theta_{s_i}) \end{bmatrix}^\top \\ &= \begin{bmatrix} e^{-j\omega\frac{\underline{\mathbf{v}}_{s_i}^\top \cdot \underline{\mathbf{P}}_1}{c}} & \cdots & e^{-j\omega\frac{\underline{\mathbf{v}}_{s_i}^\top \cdot \underline{\mathbf{P}}_J}{c}} \end{bmatrix}^\top,\end{aligned} \tag{1}$$

where $\underline{\mathbf{v}}_{s_i}^\top = \begin{bmatrix} \sin(\theta_{si}) & -\cos(\theta_{si}) \end{bmatrix}$ is the direction vector of signal s_i in terms of its angle of arrival θ_{s_i} and $\underline{\mathbf{P}}_j^\top = \begin{bmatrix} p_{j,x} & p_{j,y} \end{bmatrix}$ is the position vector of sensor j, described by its x and y coordinates. Finally ω is the angular frequency and c is the propagation speed of the signals. We then have that the signal model is described by:

$$\underline{\mathbf{x}}[k] = \sum_{i=1}^{P} \underline{\mathbf{a}}(\theta_{s_i}) \cdot s_i[k] + \underline{\mathbf{n}}[k], \tag{2}$$

where P is the amount of signal sources and $\underline{\mathbf{n}}[k]$ is a vector containing the noise. We can write this in matrix notation as follows:

$$\underline{\mathbf{x}}[k] = \mathbf{A} \cdot \underline{\mathbf{s}}[k] + \underline{\mathbf{n}}[k], \tag{3}$$

where $\mathbf{A}$ is a matrix containing the vectors $\underline{\mathbf{a}}(\theta_{s_i})$ for every source i and $\underline{\mathbf{s}}[k]$ is a vector containing the signal from every source i. Now we can describe the covariance matrix $\mathbf{R}_x$ by:

$$\begin{aligned}\mathbf{R}_x &= E\{\underline{\mathbf{x}}[k] \cdot \underline{\mathbf{x}}^h[k]\} \\ &= E\big\{(\mathbf{A} \cdot \underline{\mathbf{s}}[k] + \underline{\mathbf{n}}[k]) \cdot (\mathbf{A} \cdot \underline{\mathbf{s}}[k] + \underline{\mathbf{n}}[k])^h\big\} \\ &= E\big\{\mathbf{A} \cdot \underline{\mathbf{s}}[k] \cdot \underline{\mathbf{s}}^h[k] \cdot \mathbf{A}^h + \underline{\mathbf{n}}[k] \cdot \underline{\mathbf{s}}^h[k] \cdot \mathbf{A}^h + \mathbf{A} \cdot \underline{\mathbf{s}}[k] \cdot \underline{\mathbf{n}}^h[k] + \underline{\mathbf{n}}[k] \cdot \underline{\mathbf{n}}^h[k]\big\} \\ &= \mathbf{A} \cdot E\big\{\underline{\mathbf{s}}[k] \cdot \underline{\mathbf{s}}^h[k]\big\} \cdot \mathbf{A}^h + E\big\{\underline{\mathbf{n}}[k] \cdot \underline{\mathbf{s}}^h[k]\big\} \cdot \mathbf{A}^h \\ &\qquad + \mathbf{A} \cdot E\big\{\underline{\mathbf{s}}[k] \cdot \underline{\mathbf{n}}^h[k]\big\} + E\big\{\underline{\mathbf{n}}[k] \cdot \underline{\mathbf{n}}^h[k]\big\} \\ &= \mathbf{A} \cdot \mathbf{R}_s \cdot \mathbf{A}^h + \mathbf{R}_{ns} \cdot \mathbf{A}^h + \mathbf{A} \cdot \mathbf{R}_{sn} + \mathbf{R}_n,\end{aligned} \tag{4}$$

where $\mathbf{R}_s = E\{\underline{\mathbf{s}}[k] \cdot \underline{\mathbf{s}}^h[k]\}$ is the auto-correlation of $\underline{\mathbf{s}}[k]$, similarly $\mathbf{R}_n$ is the auto-correlation of $\underline{\mathbf{n}}[k]$ and $\mathbf{R}_{ns} = \mathbf{R}_{sn}^h = E\{\underline{\mathbf{n}}[k] \cdot \underline{\mathbf{s}}^h[k]\}$ is the cross-correlation between $\underline{\mathbf{n}}[k]$ and $\underline{\mathbf{s}}[k]$. Now we assume that the noise signal is spatially and temporally white Gaussian noise. Then we have that $\mathbf{R}_n$ is a diagonal matrix containing the elements σ_n^2 on the diagonal, thus $\mathbf{R}_n = \sigma_n^2 I$, where I is the identity matrix. The noise is also not correlated with any signal, thus we have that $\mathbf{R}_{ns} = \mathbf{R}_{sn}^h = 0$. We also assume that the signals from the

different sources are uncorrelated and that they all have the same power. Thus we have that $\sigma_{s_1}^2 = \sigma_{s_2}^2 = \cdots = \sigma_{s_P}^2 = \sigma_s^2$ and then $\mathbf{R}_s = \sigma_s^2 I$ is a diagonal matrix. These assumptions simplify equation 4 to:

$$\begin{aligned} \mathbf{R}_x &= \mathbf{A} \cdot \mathbf{R}_s \cdot \mathbf{A}^h + \mathbf{R}_n \\ &= \mathbf{A} \cdot \sigma_s^2 I \cdot \mathbf{A}^h + \sigma_n^2 I \\ &= \sigma_s^2 \mathbf{A} \cdot \mathbf{A}^h + \sigma_n^2 I. \end{aligned} \tag{5}$$

To conclude, we can write equation 5 of the auto-correlation matrix $\mathbf{R}_x$ in terms of σ_s^2, σ_n^2 and $\underline{\mathbf{a}}(\theta_{s_i})$ as follows:

$$\mathbf{R}_x = \sigma_s^2 \sum_{i=1}^{P} \underline{\mathbf{a}}(\theta_{s_i}) \cdot \underline{\mathbf{a}}^h(\theta_{s_i}) + \sigma_n^2 I. \tag{6}$$

b

From the observations $\underline{\mathbf{x}}$, we can estimate $\mathbf{R}_x$ by taking a time-average:

$$\begin{aligned} \mathbf{R}_x &= E\{\underline{\mathbf{x}}[k] \cdot \underline{\mathbf{x}}^h[k]\} \\ &\approx \frac{1}{T} \sum_{k=1}^{T} \underline{\mathbf{x}}[k] \cdot \underline{\mathbf{x}}^h[k] = \hat{\mathbf{R}}_x, \end{aligned} \tag{7}$$

where T is the sampling period over which the average is computed. In the case of our observations, we assume that the environment and source signal statistics do not change over time. Then the best estimate $\hat{\mathbf{R}}_x$ of the auto-correlation matrix $\mathbf{R}_x$ is computed by taking the average over all 5000 observations. Now we can calculate the steered response $P(\theta)$, which is defined as:

$$\begin{aligned} P(\theta) &= \frac{P_y(\theta)}{||\underline{\mathbf{w}}||^2} \\ \implies P_y(\theta) &= E\{|y|^2\} = E\{|y^2|\} = |E\{y^2\}| \\ &= |E\{(\underline{\mathbf{w}}^h \cdot \underline{\mathbf{x}}[k])^2\}| \\ &= |E\{\underline{\mathbf{w}}^h \cdot \underline{\mathbf{x}}[k] \cdot \underline{\mathbf{x}}^h[k] \cdot \underline{\mathbf{w}}\}| \\ &= |\underline{\mathbf{w}}^h \cdot E\{\underline{\mathbf{x}}[k] \cdot \underline{\mathbf{x}}^h[k]\} \cdot \underline{\mathbf{w}}| \\ &= |\underline{\mathbf{w}}^h \cdot \mathbf{R}_x \cdot \underline{\mathbf{w}}| \\ &\approx |\underline{\mathbf{w}}^h \cdot \hat{\mathbf{R}}_x \cdot \underline{\mathbf{w}}| \\ \implies P(\theta) &\approx \frac{|\underline{\mathbf{w}}^h \cdot \hat{\mathbf{R}}_x \cdot \underline{\mathbf{w}}|}{||\underline{\mathbf{w}}||^2}, \end{aligned} \tag{8}$$

where we define $\underline{\mathbf{w}} \equiv \underline{\mathbf{a}}(\theta)$, as we want to find the angles for which the steered response is maximum, to find the source locations. The elements of $\underline{\mathbf{a}}(\theta)$ have a magnitude of 1. Thus

we have that the magnitude squared of those elements is also 1, and because the length of the vector $\underline{\mathbf{a}}(\theta)$ is J, we have that $||\underline{\mathbf{w}}||^2 = ||\underline{\mathbf{a}}(\theta)||^2 = \underline{\mathbf{a}}^h(\theta) \cdot \underline{\mathbf{a}}(\theta) = J$. Finally, we can determine that:

$$P(\theta) \approx \frac{|\underline{\mathbf{a}}^h(\theta) \cdot \hat{\mathbf{R}}_x \cdot \underline{\mathbf{a}}(\theta)|}{J}. \tag{9}$$

The output steered response power $P(\theta)$ using the estimated matrix $\mathbf{R}_x$ for a 6 sensor ULA with elements spaced 3.4 cm apart is shown in figure 1. We can see that there are 4 peaks, but given that the ULA cannot distinguish between a signal coming from the front or back side of the array, we have that there are 2 sources. We know that on the domain $-90° \leq \theta \leq 90°$ the 2 sources satisfy the following constraint: $|\theta_{s_1}| < |\theta_{s_2}|$, where s_1 is the desired source and s_2 is the interference source. The maxima in the figure on the given domain are the

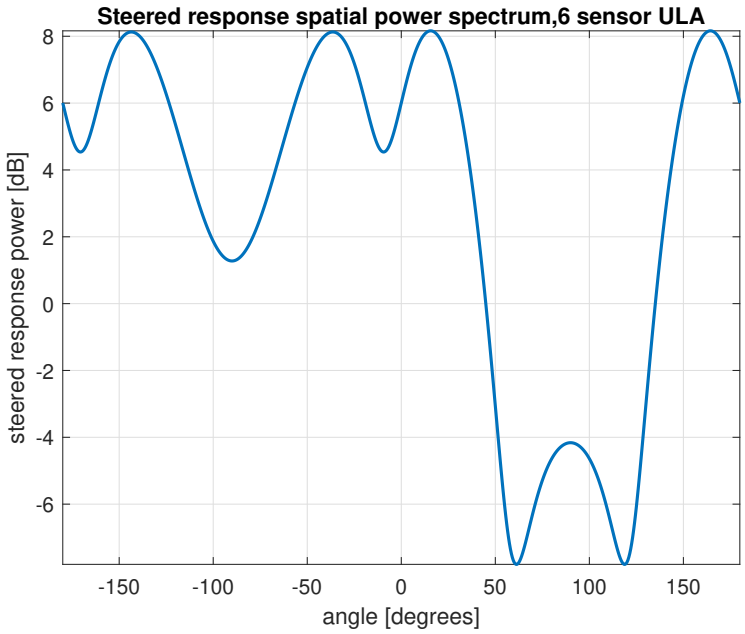

Figure 1: Steered response power of the 6 sensor ULA.

approximate locations of the sources: $\theta_{s_1} \approx 15.6°$ and $\theta_{s_2} \approx -36.4°$.

c

When using only four microphones that are closest to the origin, it means there will be less sidelobes, but the resolution is worse. The steered response power of the system, using only 4 sensors, can be seen in figure 2.

It can be seen that the resolution is in fact so low, that the peaks of the two sources are not very distinguishable, and the locations of the maxima have shifted, which can clearly be seen in figure 3. The highest peak is now very close to an angle of 0° instead of 15.5°. The peak of the other source is now gone and the existence of the second source can only be noticed by a slight change in slope around $-20°$.

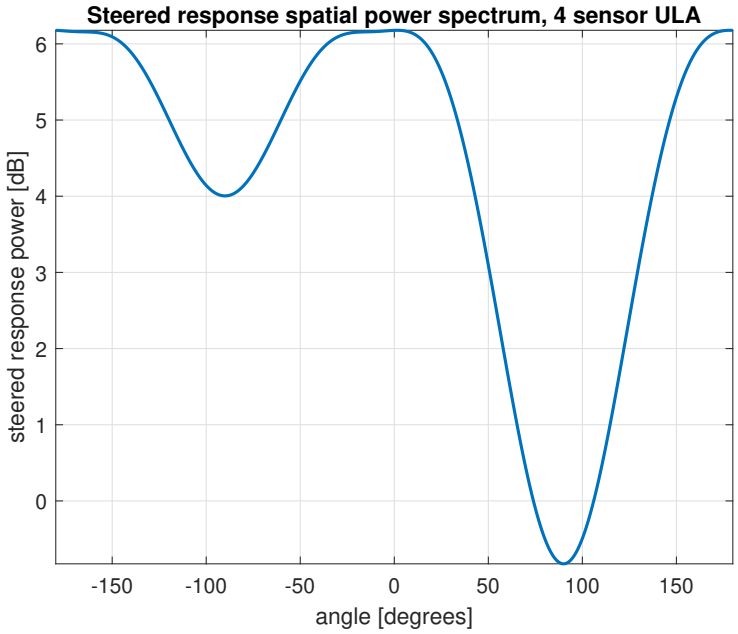

Figure 2: Steered response power of the 4 sensor ULA.

d

In the case of 2 active uncorrelated sources, whose average power may differ, and noise-free observations, we can rewrite equation 5 as:

$$\begin{aligned}
\mathbf{R}_x &= \mathbf{A} \cdot \mathbf{R}_s \cdot \mathbf{A}^h + \mathbf{R}_n \\
\mathbf{R}_x &= \begin{bmatrix} \underline{\mathbf{a}}(\theta_{s_1}) & \underline{\mathbf{a}}(\theta_{s_2}) \end{bmatrix} \cdot \begin{bmatrix} \sigma_{s_1}^2 & 0 \\ 0 & \sigma_{s_2}^2 \end{bmatrix} \cdot \begin{bmatrix} \underline{\mathbf{a}}^h(\theta_{s_1}) \\ \underline{\mathbf{a}}^h(\theta_{s_2}) \end{bmatrix} + 0 \\
\mathbf{R}_x &= \sigma_{s_1}^2 \cdot \underline{\mathbf{a}}(\theta_{s_1}) \cdot \underline{\mathbf{a}}^h(\theta_{s_1}) + \sigma_{s_2}^2 \cdot \underline{\mathbf{a}}(\theta_{s_2}) \cdot \underline{\mathbf{a}}^h(\theta_{s_2}).
\end{aligned} \tag{10}$$

We can plug this equation into equation 8:

$$\begin{aligned}
P(\theta) &= \frac{|\underline{\mathbf{w}}^h \cdot \mathbf{R}_x \cdot \underline{\mathbf{w}}|}{||\underline{\mathbf{w}}||^2} \\
&= \frac{\left|\underline{\mathbf{w}}^h \cdot \left(\sigma_{s_1}^2 \cdot \underline{\mathbf{a}}(\theta_{s_1}) \cdot \underline{\mathbf{a}}^h(\theta_{s_1}) + \sigma_{s_2}^2 \cdot \underline{\mathbf{a}}(\theta_{s_2}) \cdot \underline{\mathbf{a}}^h(\theta_{s_2})\right) \cdot \underline{\mathbf{w}}\right|}{||\underline{\mathbf{w}}||^2}
\end{aligned} \tag{11}$$

.

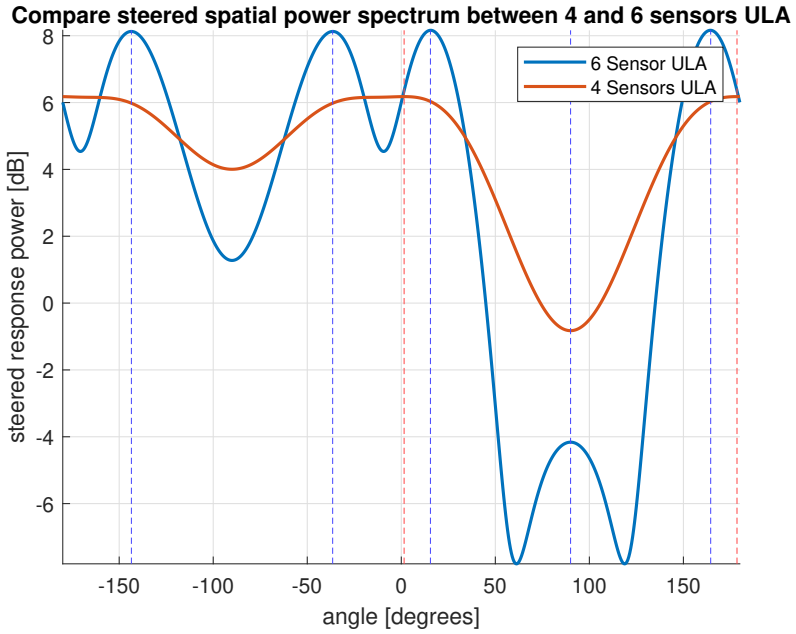

Figure 3: Steered response power of the 4 and 6 sensor ULA.

Now we can calculate the steered response in the direction of s_1, $P(\theta_{s_1})$, by defining the weights $\underline{\mathbf{w}}$ as before: $\underline{\mathbf{w}} \equiv \underline{\mathbf{a}}(\theta_{s1})$ and simplifying the above equation:

$$\begin{aligned} P(\theta_{s1}) &= \frac{\left|\underline{\mathbf{a}}^h(\theta_{s_1}) \cdot \left(\sigma_{s_1}^2 \cdot \underline{\mathbf{a}}(\theta_{s_1}) \cdot \underline{\mathbf{a}}^h(\theta_{s_1}) + \sigma_{s_2}^2 \cdot \underline{\mathbf{a}}(\theta_{s_2}) \cdot \underline{\mathbf{a}}^h(\theta_{s_2})\right) \cdot \underline{\mathbf{a}}(\theta_{s_1})\right|}{||\underline{\mathbf{a}}(\theta_{s_1})||^2} \\ &= \frac{\left|\sigma_{s_1}^2 \cdot \underline{\mathbf{a}}^h(\theta_{s_1}) \cdot \underline{\mathbf{a}}(\theta_{s_1}) \cdot \underline{\mathbf{a}}^h(\theta_{s_1}) \cdot \underline{\mathbf{a}}(\theta_{s_1}) + \sigma_{s_2}^2 \cdot \underline{\mathbf{a}}^h(\theta_{s_1}) \cdot \underline{\mathbf{a}}(\theta_{s_2}) \cdot \underline{\mathbf{a}}^h(\theta_{s_2}) \cdot \underline{\mathbf{a}}(\theta_{s_1})\right|}{\underline{\mathbf{a}}^h(\theta_{s_1}) \cdot \underline{\mathbf{a}}(\theta_{s_1})} \\ &= \frac{\left|J^2\sigma_{s_1}^2 + \sigma_{s_2}^2 \cdot \underline{\mathbf{a}}^h(\theta_{s_1}) \cdot \underline{\mathbf{a}}(\theta_{s_2}) \cdot \underline{\mathbf{a}}^h(\theta_{s_2}) \cdot \underline{\mathbf{a}}(\theta_{s_1})\right|}{J} \\ &= J\sigma_{s_1}^2 + \frac{1}{J}\sigma_{s_2}^2 \cdot \underline{\mathbf{a}}^h(\theta_{s_1}) \cdot \underline{\mathbf{a}}(\theta_{s_2}) \cdot \underline{\mathbf{a}}^h(\theta_{s_2}) \cdot \underline{\mathbf{a}}(\theta_{s_1}). \end{aligned} \tag{12}$$

We can remove the absolute value operation since the value is always real and positive, because we know that J, $\sigma_{s_1}^2$ and $\sigma_{s_2}^2$ are positive and real and we can quickly show that the remaining term is positive and real ($\in \mathbb{R}_+$) as well:

$$\begin{aligned} &\underline{\mathbf{a}}^h(\theta_{s_1}) \cdot \underline{\mathbf{a}}(\theta_{s_2}) \cdot \underline{\mathbf{a}}^h(\theta_{s_2}) \cdot \underline{\mathbf{a}}(\theta_{s_1}) \\ =&\left(\underline{\mathbf{a}}^h(\theta_{s_2}) \cdot \underline{\mathbf{a}}(\theta_{s_1})\right)^h \cdot \left(\underline{\mathbf{a}}^h(\theta_{s_2}) \cdot \underline{\mathbf{a}}(\theta_{s_1})\right) \\ =&||\underline{\mathbf{a}}^h(\theta_{s_2}) \cdot \underline{\mathbf{a}}(\theta_{s_1})||^2 \in \mathbb{R}_+. \end{aligned} \tag{13}$$

From equation 12 it can be seen that the steered response power in the direction of θ_{s_1} depends on 2 terms, a term proportional to J and a noise term proportional to $1/J$, which also depends on the signal power and location of s_2. this suggest that the more sensors are

used, the more the noise-term is suppressed and the other term contributes, which is desired. The noise term depends on the squared magnitude of the dot product of the two steering vectors of s_1 and s_2. This is a continuous function in θ_{s_1}, not a sharp peak like the spectral-MUSIC algorithm, so the resolution is much lower. This suggests that the estimation of the DOA using this equation might not be very accurate if a low amount of sensors is used. Also the closer the two sources are together, the bigger the dot product between the two steering vectors will be, and then the noise term will contribute more. Thus this equation is also only accurate when the two sources are relatively far away from each other

3.3.1 Assignment 1C: Scenario 2

a

We can use subspace techniques to find the estimated noise variance. The sensor correlation matrix can be decomposed into eigenvectors and eigenvalues as follows:

$$\mathbf{R}_x = \mathbf{U}_x \cdot \mathbf{\Lambda}_x \cdot \mathbf{U}_x^h = \mathbf{U}_s \cdot \mathbf{\Lambda}_{s,n} \cdot \mathbf{U}_s^h + \mathbf{U}_n \cdot \mathbf{\Lambda}_n \cdot \mathbf{U}_n^h, \tag{14}$$

where the source and noise components can be separated, as we assume they are uncorrelated. First we compute the estimated autocorrelation matrix $\hat{\mathbf{R}}_x$ using the observations with equation 7, such that we can then estimate the decomposition of $\mathbf{R}_x$ as well. Now we can do an eigenvalue decomposition to find the estimated noise and signal variances. We have 6 eigenvalues, and we know that there are 2 sources, thus the smallest 4 eigenvalues correspond to the noise power. Using eigenvalues decomposition, we find that:

$$\begin{aligned} \mathbf{R}_x &= \mathbf{U}_x \cdot \mathbf{\Lambda}_x \cdot \mathbf{U}_x^h \\ &= \mathbf{U}_x \cdot \mathrm{diag}(\sigma_{s_1}^2 + \sigma_n^2,\ \sigma_{s_2}^2 + \sigma_n^2,\ \sigma_n^2,\ \sigma_n^2,\ \sigma_n^2,\ \sigma_n^2) \cdot \mathbf{U}_x^h \\ \implies \hat{\mathbf{R}}_x &= \hat{\mathbf{U}}_x \cdot \hat{\mathbf{\Lambda}}_x \cdot \hat{\mathbf{U}}_x^h \\ &\approx \hat{\mathbf{U}}_x \cdot \mathrm{diag}(7.99,\ 4.35,\ 0.15,\ 0.11,\ 0.10,\ 0.10) \cdot \hat{\mathbf{U}}_x^h. \end{aligned} \tag{15}$$

To make an estimation of the noise variance, we simply take the average of the 4 computed noise components, based on the observations. Thus we have that:

$$\begin{aligned} \sigma_n^2 &\approx \frac{1}{4}(0.15 + 0.11 + 0.10 + 0.10) \\ &\approx 0.115. \end{aligned} \tag{16}$$

b

For the spectral-MUSIC algorithm, we continue from equation 15. After finding the eigenvalues, we can find the eigenvectors. As we know that there are 2 sources, we can split the

eigenvectors and values into 2 signal components and assume the rest is noise:

$$\begin{aligned}
\hat{\mathbf{R}}_x &= \hat{\mathbf{U}}_x \cdot \hat{\mathbf{\Lambda}}_x \cdot \hat{\mathbf{U}}_x^h \\
&= \begin{bmatrix} u_{s_1+n} & u_{s_2+n} & u_n & \dots & u_n \end{bmatrix} \cdot \hat{\mathbf{\Lambda}}_x \cdot \begin{bmatrix} u_{s_1+n} & u_{s_2+n} & u_n & \dots & u_n \end{bmatrix}^h \\
&= \hat{\mathbf{U}}_s \cdot \hat{\mathbf{\Lambda}}_{s,n} \cdot \hat{\mathbf{U}}_s^h + \hat{\mathbf{U}}_n \cdot \hat{\mathbf{\Lambda}}_n \cdot \hat{\mathbf{U}}_n^h \\
&= \begin{bmatrix} u_{s_1+n} & u_{s_2+n} & \underline{\mathbf{0}} & \dots & \underline{\mathbf{0}} \end{bmatrix} \cdot \mathrm{diag}(\sigma_{s_1}^2 + \sigma_n^2,\ \sigma_{s_2}^2 + \sigma_n^2,\ 0,\ \dots,\ 0) \cdot \begin{bmatrix} u_{s_1+n}^h \\ u_{s_2+n}^h \\ \underline{\mathbf{0}}^h \\ \dots \\ \underline{\mathbf{0}}^h \end{bmatrix} \\
&\quad + \begin{bmatrix} \underline{\mathbf{0}} & \underline{\mathbf{0}} & u_n & \dots & u_n \end{bmatrix} \cdot \mathrm{diag}(0,\ 0,\ \sigma_n^2,\ \dots,\ \sigma_n^2) \cdot \begin{bmatrix} \underline{\mathbf{0}} & \underline{\mathbf{0}} & u_n & \dots & u_n \end{bmatrix}^h,
\end{aligned} \tag{17}$$

where u_m is the eigenvector corresponding to the eigenvalue σ_m^2. The steered response power is the calculated using the following equation:

$$P_{SM}(\theta) = \frac{J}{\underline{\mathbf{a}}^h(\theta) \cdot (\hat{\mathbf{U}}_n \cdot \hat{\mathbf{U}}_n^h) \cdot \underline{\mathbf{a}}(\theta)}. \tag{18}$$

The implementation is done as follows:

```
Rx = 1/length(x)*(x*x'); %x is the observation matrix
[Ux,Lambda_x]=eig(Rx);
Un = [Ux(:,1:4), zeros(6,2);];
Pn = Un*Un';
P_music = zeros(length(x),1);
for i = 1:length(x)
    P_music(i)=J/abs(a(:,i)'*Pn*a(:,i));
end
```

The spatial spectrum of the 6 sensors ULA, computed using the spectral-MUSIC algorithm, is shown in the figure 4. We can see that there are 4 peaks, but given that the ULA cannot distinguish between signals coming from the front or back of the array, we have that there are 2 sources. The sharpest peaks in the spectrum calculated according to the spectral-MUSIC algorithm, determine the directions of arrival. According to this method the locations of the sources are determined to be: $\theta_{s_1} \approx 13.6°$ and $\theta_{s_2} \approx -34.0°$.

These values are slightly different from the calculated angles using maximized steered response power, which were $\theta_{s_1} \approx 15.6°$ and $\theta_{s_2} \approx -36.4°$. That is a 2° and 2.4° difference for s_1 and s_2 respectively. We now compare the spectral-MUSIC algorithm, with the method from 1C, scenario 1 (see figure 5), then we can see that the sidelobe near 90° is compressed a lot (it is almost flat) and that the peaks at the source locations are much sharper. This means the resolution is very high compared to the first method as noise near the sources is suppressed a lot more compared to the first method.

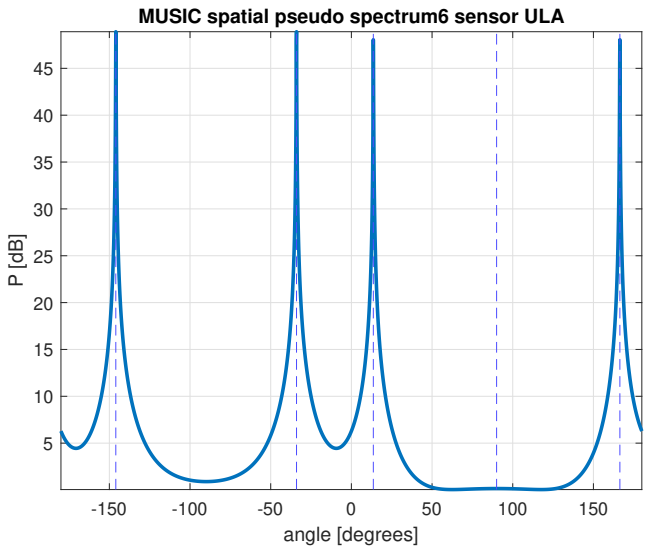

Figure 4: Spatial spectrum of 6 sensors ULA using spectral-MUSIC algorithm.

c

The spectral-MUSIC algorithm will try to find a fixed amount of sources. The amount of sources determines how the eigenvalue decomposition of equation 17 splits up the noise and source components. If the wrong amount of sources is specified, the matrix $\hat{\mathbf{U}}_n$ will either not contain all noise components, or also include signal components. If not all noise components are contained, then the matrix will have too many zero vectors. For every zero vector it will try to find a source location, as in equation 18, there is an inverse relation between $\hat{\mathbf{U}}_n$ and P_{SM}. Division by a value close to 0 will result in high values for P_{SM}. Thus if you estimate too many sources, then the algorithm will 'try' to make too many peaks. If you estimate too little amount of sources, then the algorithm will try to suppress peaks, because of the division over a large number (large vector in $\hat{\mathbf{U}}_n$).

This result can be seen in figure 6. If 3 sources are estimated, the locations of the two real sources are determined to be: $\theta_{s1} \approx 13.8°$ and $\theta_{s2} \approx -34.2°$. The difference in angle between these and the angles when the correct number of sources are estimated is: $|\Delta\theta_{s1}| \approx 0.2°$ and $|\Delta\theta_{s2}| \approx 0.2°$. These estimations are still pretty accurate compared to the maximized steered response power estimate, for example. However, it can also clearly be seen that there is a big bump near $-20°$, which is a result of the expected 3rd source. The algorithm tries to 'find' an extra source location, that does not exist.

If 1 source is estimated, the location of the source is determined to be: $\theta_{s1} \approx 17.9°$ and $\theta_{s2} \approx -39.3°$. The difference in angle between these and the angles when the correct number of sources are estimated is:$|\Delta\theta_{s1}| \approx 3.3°$ and $|\Delta\theta_{s2}| \approx 5.3°$. It can bee seen that the algorithm suppresses the peaks as a result of expecting only 1 source. Not only is the estimation accuracy decreased (bad angle estimation), also the peaks are not very pronounced, decreasing the spatial resolution.

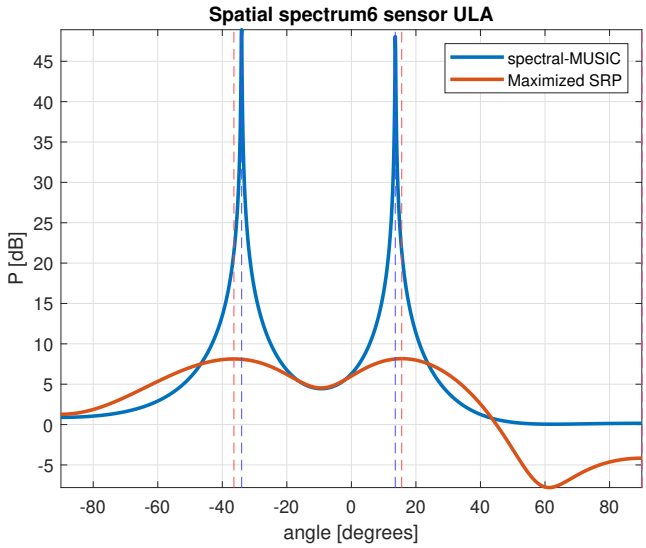

Figure 5: Spatial spectrum of 6 sensors ULA using spectral-MUSIC algorithm and maximized steered response power on the domain $[-90°, 90°]$.

3.4.1 Assignment 1C: Scenario 3

To implement the beamformer, we adopt the same mathematical model as in the paper by Synnevåg et al. We divide the signal array into overlapping sub-arrays, and then average the covariance matrices for all sub-arrays.

$$\underline{\mathbf{w}}(t) = \frac{\hat{\mathbf{R}}_x^{-1} \cdot \underline{\mathbf{a}}}{a^h \cdot \hat{\mathbf{R}}_x^{-1} \cdot \underline{\mathbf{a}}}, \tag{19}$$

where $\underline{\mathbf{a}}$ is a vector of ones, and $\underline{\hat{R}}_x$ is described by:

$$\hat{\mathbf{R}}_x(t) = \frac{1}{N-L+1} \sum_{l=0}^{N-L} x_l(t) \cdot x_l^h(t), \tag{20}$$

where $x_l(t) = \begin{bmatrix} x_l(t) & x_{l+1}(t) & \cdots & x_{l+L-1}(t) \end{bmatrix}^T$, a subarray of the measured input $x(t)$. L is the subarray length and N is the array length. To guarantee a stable solution of the inverse of $\hat{\mathbf{R}}_x$, we add a diagonal loading factor to the matrix according to:

$$\hat{\mathbf{R}}_x(t) = \hat{\mathbf{R}}_x(t) + \left(\frac{1}{100L}\text{tr}\{\hat{\mathbf{R}}_x(t)\}\right) I, \tag{21}$$

where $tr\{\cdot\}$ is the trace operator. The code that implements the adaptive capon estimator for the weights $\underline{\mathbf{w}}$ is shown below.

```
if adaptive % set adaptive = 1
    Rx_est = zeros(L);
    % Calculate the Rx_est
```

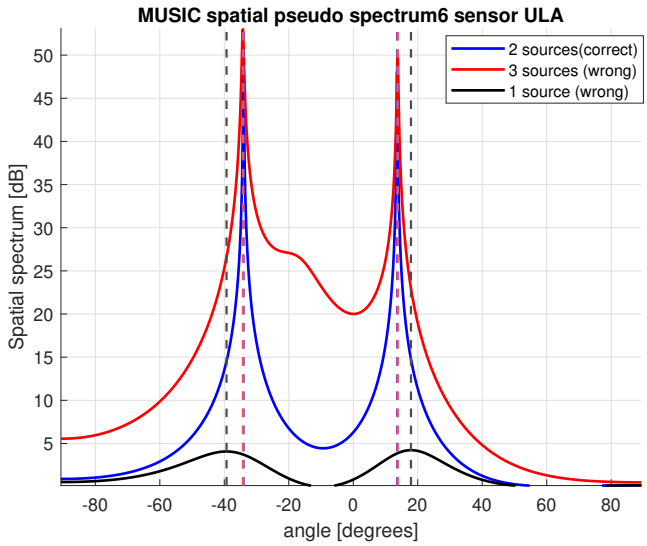

Figure 6: Comparison of spatial spectrum between different estimated number of sources for 6 sensor ULA using spectral-MUSIC algorithm on the domain $[-90°, 90°]$.

```
for j = 1:(N-L+1)
    Rx_est = Rx_est + y(j:j+L-1)'*y(j:j+L-1);
end
% Use the method suggested in the paper to estimate covariance:
% Add delta= 1/(100L).*tr(Rx_est), to ensure well conditied cov matrix
epsilon = (1/(100*L))*trace(Rx_est./(N-L+1));
Rx_est = (Rx_est./(N-L+1))+epsilon.*eye(L);
Rx_inv = Rx_est^(-1);
% Compute the capon filter
w = Rx_inv*a*(a'*Rx_inv*a)^(-1);
% Estimated amplitude
z = 0;
for j = 1:(N-L+1)
    z = z + w'*y(j:j+L-1)';
end
beamformed_channel_data(i, pw) = z./(N-L+1);
```

The results obtained using the **point_scat.mat** ultrasound dataset are shown in figure 7, and figure 8. In all images the right image is the adaptive beamformer and in the left image is the non-adaptive beamformer. The figure 7 shows the beamformed image at an angle 0°, using non-adaptive and adaptive beamforming. Figure 8 shows the beamformed image after summing contribution of multiple angles ($-16°$ to 16°, 75 datasets) using non-adaptive and adaptive beamforming. The figure 9 shows beamformed images from different angles. The several bright spots in the beamformed images are the targets, the rest are seen as noise.

The difference between using one of multiple angles

For one of multiple angles, there are always "X"shape noise around the bright spot in the

non-adaptive beamformer, while for the adaptive beamformer, the "X" shape noise is being filtered out nicely. However, the gray-band noise in the spectrum (around $z = 0 \sim 30mm$) appears in both non-adaptive and adaptive beamformers, but it slightly less for the adaptive beamformer (see figure 7 and 9). The gray-band noise disappeares only when summing the contributions of many different angles (see figure 8). It appears that the non-adaptive beamformer performs very good as well, when there are many contributions from different angles.

Comment 2. Discuss the difference in terms of complexity and performance between adaptive vs non-adaptive)
In terms of complexity, the non-adaptive beamformer outperferms the adaptive beamformer drastically, as it is approximately 40 times faster. THis makes sense since the non adaptive beamformer does only a vector multiplication inside 2 nested for loops to finish the computation. One loop to loop over all angles and one to loopover the number of apodizations. The adaptove filters also walks through these loops, but inside also loops over the signal length N to recursively calculate the the auto-correlation matrix and the output twice, and also has to compute a large matrix inversion.

The performance difference between non-adaptive and adaptive beamformer can clearly be seen in figure 7 and figure 8. At angle = 0° , the non-adaptive beamformer result contains a lot of noise in the spectrum (around $z = 0 \sim 30mm$). Also, the bright spots are hard to localize, due to the "X" shape noise, which lower the resolution. When using the adaptive beamformer, globally, the noise decreases a lot all over the spatial spectrum. Around the bright spots (targets), the "X" shaped noise disappeares when using the adaptive beamformer, making the samples more distinguishable and the resolution higher. On the other hand, after summing the contributions of different angles (as shown in the figure 8), the gray noise brand at $z = 0 \sim 30(mm)$ are compressed a lot for both the adaptive and non-adaptive method. For the non-adaptive beamformer, the "X" shape noise around bright spots are also compressed a lot, but the the width of the bright spots looks a bit ambiguous compare to the adaptive beamformer, whose spots are a bit dimmer in some areas, but they seem more concentrated. It is hard to determine which is more accurate, since we do not have ground truth data. In conclusion, when doing the measurement on single angles, the adaptive beamformer performs much better than the non-adaptive beamformer in terms of performance. When doing the measurements on multiple angles, the performance of adaptive beamformer is similar to the non-adaptive case, but it takes much more time to do the processing. This suggests that if large amounts of data from different angles is available, a non-adaptive beamformer is sufficient to obtain good results.

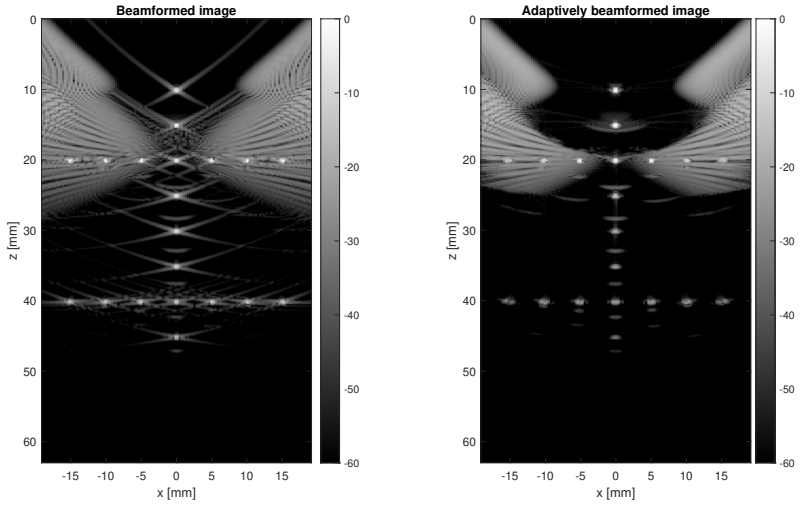

Figure 7: Compare the beamformed image between non adaptive (left) and adaptive (right) delay-and-sum beamforming at angle = 0°.

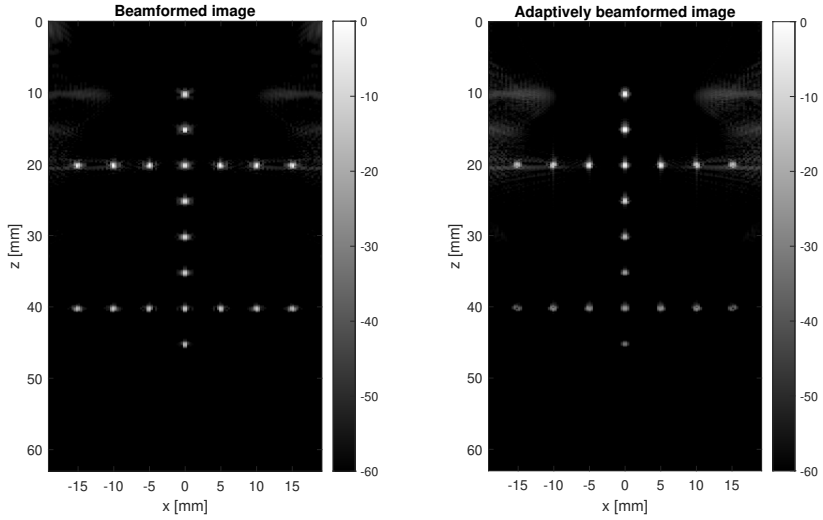

Figure 8: Compare the beamformed image between non adaptive (left) and adaptive (right) delay-and-sum beamforming after summing the contributions of different angle ($-16°$ to $16°$).

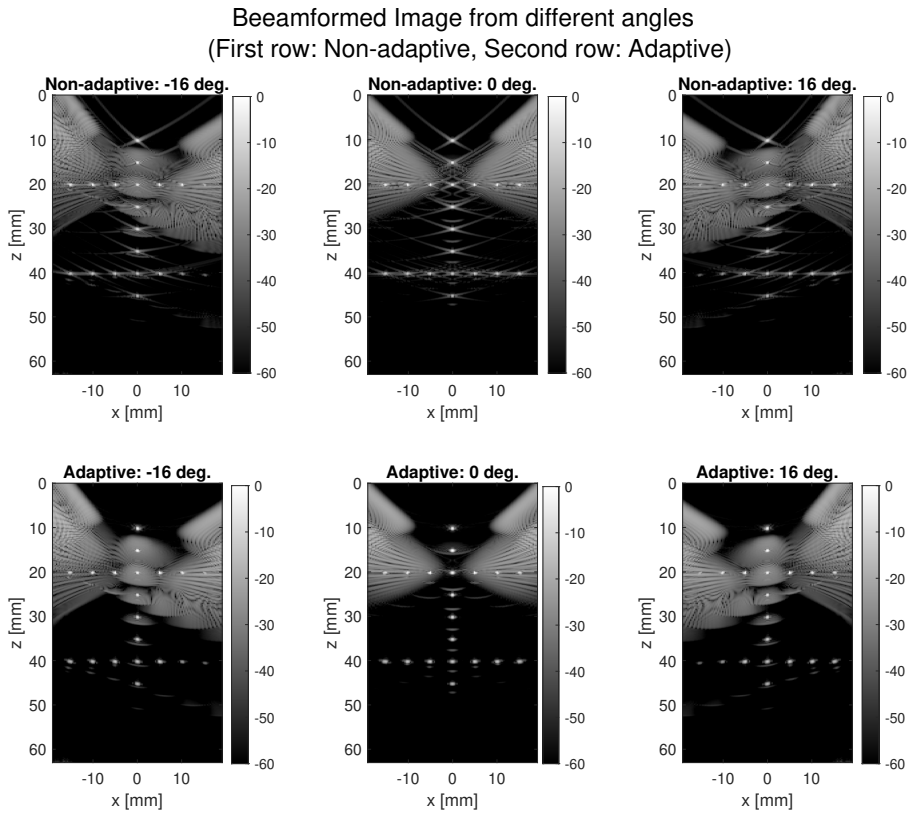

Figure 9: Compare the beamformed images from different angles ($-16°$, $-0°$, $16°$).