IN5450/9450 Array Signal Processing mandatory assignment 2

Andreas Austeng and Sven Peter Näsholm

March–April, 2022

Distribution date	14 March, 2022
Deadline	4 April, 2022
Version	1.0, March 11, 2022
Student report submission	use the *Devilry* portal

Part A: High-Resolution Beamforming on farfield monochromatic signals

Background and instructions

This is based on the examples shown in Figures 4b and 5 in the paper:

H. Krim, M. Viberg, *Two decades of array signal processing research – The parametric approach*, IEEE Signal Processing Magazine, pp. 67–94, July 1996

You may download a PDF hardcopy of the paper from here: http://dx.doi.org/10.1109/79.526899
You will need to be on the University of Oslo network to get access, or you can use this browser plugin: https://www.ub.uio.no/english/about/news/all-libraries/2018/lean-library.html)

Students' feedback over the years indicate that they learn a lot from solving the tasks in this assignment. In fact, a variation of the current task set was introduced into a predecessor of the current Array signal processing course already more than 2 decades ago.

- Problem 1)–7) are mandatory for all students.
- Problems 8) is mandatory for PhD-level students (course code INF9450), but voluntary and still recommended for other students. .
- The subject is on estimation of the spatial spectrum (the direction of arrival) for an $M = 10$ element uniform linear array with half-wavelength spacing ($d = \lambda/2$). The input consists of two incoherent signals at 0 and -10 degrees in additive spatially white noise. The signal-to-noise ratio (SNR) for both sources is 0 dB, and $N = 100$ samples are available of the input.

  The models are described on page 73 of the paper.
- The MATLAB code for the generation of the input dataset format is found in the folder: https://www-int.uio.no/studier/emner/matnat/ifi/IN5450/v22/undervisningsmateriale/mandatory_two/ and the file `generate_data.m` therein.

Checklist for figure plotting and saving

Make sure that your plots fulfill the following:

- x axis labels are always provided, including units.
- y axis labels are always provided, including units.

- Font sizes are large enough to come out with at least the size of the main body text.
- Line widths are wide enough to come out well in the final layout.
- Line styles and coloring are selected so that the different curves are distinguishable also in black-and-white printouts.
- The figure captions tell what the figure shows and what are the curves.
- There is a legend provided.
- Avoid using the *Jet* colormap. It's preferred to instead if you use, e.g., Batlow, Turbo, or other more perceptually uniform maps. More reading here on this topic here: https://www.mn.uio.no/ceed/english/about/blog/2020/using-better-colours-in-science.html and here: https://www.fabiocrameri.ch/colourmaps/ or here: https://ai.googleblog.com/2019/08/turbo-improved-rainbow-colormap-for.html if you're interested.
- It is preferred that figures are exported from Matlab / Python in a vectorized format (PDF / SVG or possibly EPS). In Matlab, it is often experienced that the `export_fig` add-on is superior for export of high-quality graphics compared to the builtins. (In case the programme you use for generating your slides does not work with vector graphics, you can instead use a pixelated format like PNG.)

Concrete tasks and hints for assignment Part A

1) Run `generate_data.m` and let the variable x be your signal.

   Estimate the spatial correlation matrix and plot its absolute value.

   It can be advisable to us the `imagesc` function (in MATLAB) or `imshow` in Python matplotlib.pyplot.

   Hint: The variable N is the number of time samples. Then, you can estimate $\mathbf{R}$ using the outer product multiplication `R = x*x' / N`, where the `'` operator is the complex conjugate transpose.

2) Estimate the classical spatial spectrum using the conventional method (Figure 4, bottom panel) and Equation (27). Plot the response both in linear and dB scale. Discuss why the sources are not separated.

   Hints:

   - You can set up a normalized wave vector with half a wavelength spacing using: `d = 0.5; kd = 2*pi*d;`

     Then, you can loop over the DOA angles $\theta \in [-40°, 50°]$ using a step of, e.g., 0.25°.
   - For each angle, you calculate the phase factor stepping `phi = -kd*sin(DOA*pi/180);` and thereafter pack these into the steering vector using the following expression:
     `a = exp(j*phi).^[0:M-1]';` where M is the number of sensor elements.
   - If `n` is your DOA loop counter, then you can calculate the conventional delay-and-sum beamformer output using `P_DAS(n) = a'*R*a/M;`

     As noted below Equation (29), the denominator of Equation (27) is equal to the number of elements when we have a uniform linear array.

3) Estimate the spatial spectrum for the same signal using the minimum variance beamformer (Capon's beamformer) (Figure 5). Plot the response both in linear and dB scale. Discuss the differences from the conventional beamformer.

   Hint: The Capon beamformer output power is given from Equation (32). Note that you need to invert the spatial covariance matrix. You can hence use an expression on this form: `P_capon(n) = 1 / (a'*Rinv*a)`, where you have pre-calculated `Rinv = inv(R)`.

4) Plot the distribution of the eigenvalues of the correlation matrix and explain it on the basis of the signal and noise model.

   Hint: First, you shall make an Eigendecomposition of $\mathbf{R}$: `[V,D] = eig(R);` Then you shall make sure to sort the normalized eigenvectors according to the associated eigenvalue in ascending order.
   `[dd,I] = sort(diag(D)); dd = flipud(dd); V = V(:,flipud(I));`

   In your plot of the sorted eigenvalues, verify that there are two which are greater than the others.

5) Estimate the spectrum using the MUSIC algorithm (Figure 5) assuming that the number of signals is known. Plot the response both in linear and dB scale. Discuss the differences from the previous estimates.

6) Estimate the spatial spectrum by the eigenvector method (see the lecture notes for definition). Plot the response both in linear and dB scale. Discuss the differences from the MUSIC beamformer.

7) *Incorrect estimate of the number of sources*
Estimate the spatial spectrum with the MUSIC method (and eigenvector method) when the number of signals is incorrectly estimated. Let the estimate of the number of signals be $0, 1$, and 3. Discuss the differences between the estimates for the various cases. (Which spatial spectrum estimator is the eigenvector method equivalent to when 0 signals are assumed to be present?)

8) *Coherent sources*
Modify the signal generator so that it generates coherent signals instead. Find the properties of the previous beamformers for coherent signals (You may have to change the angles of incidence also). Implement the various forms of averaging of the correlation estimate and see if this gives the methods a better ability to handle coherence.

Part B:
Working on signals recorded from a commercially available microphone array

Many thanks to Jon Petter Aasen at Squarehead Technology for suggesting and designing this task!

- First, download the dataset following the link and password provided here: https://www-int.uio.no/studier/emner/matnat/ifi/IN5450/v22/undervisningsmateriale/mandatory_two/link-and-password-for-dataset-to-part-b.html (requires login)

- Move the downloaded zip file `in5450_oblig2_extra_task_data.zip` to a dedicated folder you have created.

- Unzip `in5450_oblig2_extra_task_data.zip`

- Unzip also the file `data.int16.zip` that appeared after the previous unzip.

- Then, download the Jupyter notebook `in5450_oblig2_extra_task.ipynb` which is available here: https://www-int.uio.no/studier/emner/matnat/ifi/IN5450/v22/undervisningsmateriale/mandatory_two/

- You hence need an install of Python and Jupyter Notebook. Feel free to use your favorite Python install and package mechanism.

  If you prefer Conda (see here for a basic install https://docs.conda.io/en/latest/miniconda.html), you can do the following to set up a basic environment and activate it

  ```
  > conda create -n in5450 python=3.8
  > conda activate in5450
  > conda install -c conda-forge notebook
  > conda install -c conda-forge numpy plotly pandas
  ```

  Now navigate to the folder where you have saved the notebook file `in5450_oblig2_extra_task.ipynb` and start Jupyter Notebook:

  ```
  > jupyter notebook
  ```

  Then you can load and run `in5450_oblig2_extra_task.ipynb` from withing Jupyter. You might need to adjust the variable `INPUT_PATH` to point to the folder where you have stored the dataset.

Your assignment submission and presentation for Part A

Please provide an electronic presentation containing a short description of the problem, necessary derivations, results, and discussions.
In addition to the presentation, please also provide your MATLAB source code function as well as a separate script which does all the calls needed to generate all your figures. This script shall be named `run_script.m` (or `run_script.py` if you code in Python) and please make sure a command-line call to these scripts will run all your calculations and plots.

- Your presentation should not exceed 20 slides. The time for you to present it must not be more than 15 minutes.
- Submit your files to Devilry in a zip archive named `mandatory_2_LASTNAME.zip`, where you replace LASTNAME with your last name.

Your assignment presentation for Part B

For Part B, you must be able to run the Jupyter notebook in an interactive session live on your own Laptop (or remote login). In our common session, you must be able to talk the audience through the processing and descriptions in the notebook, explain what are the different processing steps, and discuss this in class.