Spring 2007, Aaron Lanterman

ECE 6279: Spatial Array Processing Homework 4

Due: Mon 3/26/07 (campus) or Mon 4/2/07 (video)

You are welcome to discuss approaches to the problems and solutions to difficulties you encounter with one another and with others outside the class. You can and should learn from each other as much as, and even more than, you learn from me. However, your solutions should be your own work and should be written up by yourself; feel free to discuss things, but don't be looking at someone else's paper when you are writing your solution. It's too easy to freeload that way and not learn anything. See the class website for more guidelines.

For on-campus students, homework is due at the *start* of class; for video students, please have your homework submitted by noon on the day it's due (that gives time for the distance learning staff to get it logged and on its way to me that day.)

1 Required Problems

1. Consider a linear five-element array with elements at $x = -3, -1, 0$ and 2. (Note there isn't an element at $x = 1$ and $x = -2$.)

   (a) Find the co-array, along with the coarray values. Give your answer by plotting the co-array values along the vertical axis, with the postition in the coarray indicated by the horizontal axis. (To do problems like this, I find it convenient to draw out the array with little marks, draw out its mirror image, and then manually convolve the two, building up marks in the answer as I go along. Your mileage may vary.)

   (b) Is the array redundant, i.e., are there any non-zero lags with a co-array value greater than 1?

2. Consider a linear 7-element array with elements at $x = -4, -3, -1, 0, 1, 3, 4$. (Note there aren't elements at $x = \pm 2$.) Find three compatible subarrays of three elements each that you could use to do subaperture averaging.

3. In this problem and the next problem, we will further explore the 9-element array from Homework #3; hence, you will be able to reuse a lot of your code from that homework. (For both this problem and the next problem, be sure to turn in a listing of your code. Obviously, you don't need to turn in a separate listing for every parameter variation given in each problem; just one will suffice.)

   In this problem, you will create some example graphs comparing the steered responses of the conventional, MVDR, EV, and MUSIC algorithms as a function of θ. Plot the results of all four algorithms on a single graph. For each scenario given below, present two graphs, one using the "ideal" covariance and one using an "empirical" covariance from a limited number of snapshots. Use the same number of snapshots for all the scenarios. Choose enough snapshots that you get decent results, but don't choose so many that the ideal and empricial results look the same. You will need to experiment to find a good number of snapshots.

Consider two sources with power 2 and a noise power of 10. Set the true ϕ^0 angles for both targets to 60°, and plot your steered response as a function of θ for a fixed look angle $\phi = 60$. You will need to find good example θ^0 angles to use in your demonstrations via experimentation. For the EV and MUSIC algorithms, assume $N_s = 2$ (the correct number of targets); we'll explore the effect of assuming the wrong number of targets in the next problem.

- Scenario 1: Present an example in which the sources can be resolved by the MVDR beamformer but not by the conventional beamformer.
- Scenario 2: Now, by slowly separating the sources and looking at the results, present an example in which the sources are resolved by both the conventional and MVDR beamformer.
- Scenario 3: Now, by starting with the θ^0 parameters you used in Scenario 1 and slowly moving the sources closer to each other, present an example in which the sources are not resolved by either the conventional or MVDR methods.

Above, when I say "slowly move," I intend you to manually experiment with different values; there's no need to write any sort of complicated code to automatically find interesting values. The point of this problem is the intuition you will gain by playing around with different parameters.

4. Here we'll keep playing with that same 9-element array.

   We'll now dispense with the business of using the "ideal" covariance matrix. However, to make sure there's not too much variability from solution to solution, let's use 100 snapshots to form our covariance estimate. (There will still be some variability, so you may want to run the experiment a few times to watch overall trends and make sure you didn't get just one "lucky" case. You need only turn in one set of plots, though.)

   Let's have two sources, one at $\phi_1^0 = 45°$ and $\theta_1^0 = 30°$, and another $\phi_2^0 = 45°$ and $\theta_2^0 = 60°$. Let's let each source have power 1/4 (so you'll want to multiply your steering vectors by 1/2 when making the "seen by array" variables), and let's make the noise have power 2. Notice we're really pushing our algorithms hard now; the signals are pretty weak compared to the noise.

   To best show details, please plot everything on a decibel scale (i.e., it's time to pull out MATLAB's `log10` function).

   (a) Plot the power of the steered response of the eigenvalue method beamformer and the MUSIC beamformer as a function of θ, for $0 \leq \theta < 360°$, for a fixed $\phi = 45°$. Do this assuming zero signals, one signal, two signals, and finally three signals.

To see the effects I want you to see, it will really help to put all the eigenvector method cases on one single plot, and all the MUSIC cases on another plot.

(Note: In this particular experiment, I wasn't able to see any big differences between the EV and MUSIC techniques. If you manage to find some, let me know!)

(b) What method does EV technique correspond to for the $N_s = 0$ case?

(c) Does the MUSIC technique give you anything useful for the $N_s = 0$ case?

(d) For each technique, how big does N_s need to be before you can clearly see both targets?

(e) What difference do you see between the $N_s = 2$ and $N_s = 3$ cases?

(f) Finally, plot the steered response of the Pisarenko harmonic decomposition beamformer as a function of θ, for $0 \leq \theta < 360°$, for a fixed $\phi = 45°$. Comment on the usefulness of the results.

5. Consider this constrained optimization problem: minimize $\mathbf{w}^H\mathbf{R}\mathbf{w}$ over $\mathbf{w}$, subject to the constraint that $\mathbf{e}^H(\vec{k})\mathbf{R}_{MUSIC} = 1$, where $\mathbf{R}_{MUSIC}$ is the correlation matrix that is used by the MUSIC method when N_s signals are assumed, and $\mathbf{R}$ is the usual spatial correlation matrix.

   (a) Find the solution $\mathbf{w}_\diamond$ to this problem. Simplify your answer as much as possible. (Hint: after rearranging sums, it's good to remember that the eigenvectors are orthonormal).

   (b) Using the result from part (a), what well-known beamformer is the solution to this optimization problem?

6. Knowing that many of the shortcomings of Pisarenko's method result from using only a single eigenvector, an ECE6279 student decides to modify the algorithm somewhat. For M sensors and N_s signals ($N_s < M$) in white noise, he will use the average of the $M - N_s$ smallest eigenvalues

$$\mathbf{v}_{AVG} = \frac{1}{M - N_s} \sum_{i=N_s+1}^{M} \mathbf{v}_i$$

in place of the smallest eigenvector that is usually employed in Pisarenko's method.

   (a) Is this new vector, $\mathbf{v}_{AVG}$, still orthogonal to the signals? Why or why not?

   (b) Explain why this this student's proposed method won't perform any differently than the regular Pisarenko method *in white noise*.