The Cooper Union Department of Electrical Engineering
Prof. Fred L. Fontaine
ECE416 Adaptive Algorithms
Beamforming Problems
February 16, 2023

1. **Sensor Array Signal Model**

   Assume all random signals are 0-mean. This means, for example, correlation matrix equals covariance matrix.

   First some preliminary comments regarding Gaussian noise. If $v = v_I + jv_Q$ is scalar 0-mean Gaussian, its variance $E\left(|v|^2\right) = E\left(|v_I|^2\right) + E\left(|v_Q|^2\right)$ (even if v_I, v_Q are correlated!). Thus, for example, the following will generate an $M \times N$ matrix, where every entry is UNIT variance complex Gaussian, and all entries are independent:

   $$v = 1/sqrt(2) * (\mathrm{rand}\, n\,(M, N) + j * \mathrm{rand}\, n\,(M, N))$$

   Suppose we want to create an $M \times 1$ random Gaussian vector with covariance matrix C, and mean μ. If v is a unit variance white (complex) Gaussian vector, then:

   $$x = C^{1/2} v + \mu$$

   will yield $x \sim N(\mu, C)$ where $C^{1/2}$ could either be the Cholesky factor or Hermitian square root of C. [If C is diagonal, they would be the same; otherwise, for this assignment, use the Cholesky factor]

   Here we will assume M sensors, L sensors, and N snapshots. Following the notation of the notes, $\vec{s}(\Theta)$ denotes a unit length steering vector for AOA $\Theta = (\theta, \phi)$, S is the $M \times L$ matrix whose columns are the source steering vectors, A is the $L \times N$ matrix of the coefficients $\{\alpha_\ell[n]\}$ riding on the respective steering vectors (ℓ the source index and n the snapshot), and V is the $M \times N$ matrix of noise samples. The output of the sensor array at snapshot n is:

   $$\mathbf{u}[n] = \sum_{\ell=1}^{L} \alpha_\ell[n]\, \mathbf{s}(\Theta_\ell) + \mathbf{v}[n]$$

   Collecting the output vectors $\mathbf{u}[n]$, $1 \le n \le N$ into an $M \times N$ matrix X, we get:

   $$X = SA + V$$

   Assume all the sensor locations are on a rectangular grid with spacing d, i.e., every location:

   $$\vec{r} = \begin{bmatrix} m_x d \\ m_y d \\ m_z d \end{bmatrix}$$

   where m_x, m_y, m_z are integers.

(a) Write code to generate S, A, V and X given the following parameters:

- A list of M 3-D integer vectors (e.g., a $3 \times M$ matrix) that specify the locations of the sensors.
- d/λ
- A list of L of AOA's (θ, ϕ) for the sources (e.g., a $2 \times M$ matrix).
- The relative powers on a decibel scale, and the noise power. For example, taking the first sensor as the reference, with three sources, $[0, -3, -5]$ and noise power -10 means the second source is $3dB$ below the first, the third source is $5dB$ below the first, and the noise is $10dB$ below the first source. **Note:** Since we are taking our steering vectors to be unit vectors with $1/\sqrt{M}$ normalization, scaling is a bit tricky. If the noise power is say P_{noise} dB, the variance of each of the components of an $M \times 1$ noise vector should be $\frac{1}{M} 10^{P_{\text{noise}}/10}$. If you have say $\alpha \mathbf{s}(\Theta)$ with $\mathbf{s}$ a unit steering vector, if α is said to have power P_α in dB, then α should have variance $10^{P_\alpha/10}$.
- N, the number of snapshots

Remark: Encapsulate your code. For example, write a function to compute the 3-D unit vector $\hat{a}(\theta, \phi)$ from θ, ϕ, and function to generate a steering vector given the source locations, d/λ, and θ, ϕ. You will need this steering vector generator, for example, when you search for peaks in the MUSIC and MVDR spectrum.

(b) Compute the theoretical correlation matrix R of each snapshot $\mathbf{u}[n]$, and the estimate $\hat{R} = \frac{1}{N} X X^H$.

2. **SVD and MUSIC / MVDR Spectra**

Take $d/\lambda = 0.5$, and consider an array in the form of a cross of two linear arrays aligned along the x- and y-axes, respectively. That is, take locations at $(md, 0, 0)$ and $(0, md, 0)$ where $-10 \leq m \leq 10$ (Don't count the origin twice!) Assume three sources, with source #1 the strongest, the other two sources $4dB$ and $8dB$ below the primary, and the noise $12dB$ below the primary. Assume the sources are uncorrelated with each other. Assume AOAs as follows:

$$\begin{aligned} \theta_1 &= 15^\circ,\ \phi_1 = 30^\circ \\ \theta_2 &= 20^\circ, \phi_2 = 40^\circ \\ \theta_3 &= 30^\circ, \phi_3 = -40^\circ \end{aligned}$$

Take $N = 100$ snapshots.

(a) Compute an SVD of X, draw a stem plot of the singular values, and confirm there are 3 dominant singular values. Compute σ_4/σ_3 (this will give you a sense of how large the "dropoff" to the next singular value is).

(b) Working off the theoretical correlation matrix R, compute the eigenvalues (sorted in descending order), draw a stem plot, and confirm there are 3 dominant values. Compare λ_4/λ_3.

(c) Using the SVD, compute the projection matrix P_N onto the noise subspace. Compute $|P_N s(\Theta_\ell)|$, $1 \leq \ell \leq 3$, for the source singular vectors. These should be 0 in theory.

(d) We want to examine the MUSIC spectrum based on the SVD of the data, and the MVDR spectrum based on the estimated correlation matrix $\hat{R}$. This is comparable because both are based on the random data, not theoretical values. Generate a grid of (θ, ϕ) with $0 \leq \theta \leq 90^\circ$ and $-180^\circ \leq \phi \leq 180^\circ$ (i.e., the upper hemisphere) with a resolution of 1°. [You may want to start with something coarse, like 10°, and reduce it if it doesn't seem to be straining your computer too much; if you can go down to 1° that's great, but don't try to go finer than that]. Obtain separate plots of each, using whichever form you think is more informative (a contour plot, color mapped image, surface or mesh plots, etc.). Place markers at the 3 sources. **Note:** I think a color mapped image, e.g., red at peaks and blue at nulls, works best, but use something else if you prefer.

(e) Take the "slice" where $\theta = 20^\circ$ and obtain the 1-D plot for $-180^\circ \leq \phi \leq 180^\circ$. You'll note one of your sources lands exactly on this slice.

(f) Repeat for a slice at $\phi = 30^\circ$ and $0^\circ \leq \theta \leq 90^\circ$.

(g) Since the MUSIC spectrum and MVDR spectrum are computed for steering vectors, not "general" vectors, finding their theoretical minimum can be complicated. But we can at least get a lower bound: assuming we evaluate these spectra for UNIT LENGTH vectors, obtain a lower bound for each. That is, if you are allowed to plug in ANY unit vector (not necessarily a steering vector), what would be the MINIMUM for each? **Hint:** If C is Hermitian pd and x a unit vector what can you say about the range of values of $x^H C x$?

(h) Plug in the three source steering vectors into the MVDR and MUSIC spectra, and compare to the theoretical lower bounds found above. You expect a peak at the source steering vectors: I want you see how tall the peak is relative to the theoretical minimum values.

(i) Actually, since you are computing the MVDR and MUSIC spectra on a grid of (θ, ϕ) points, find the overall minimum values you computed, and compare to the theoretical lower bounds you found. Are they reasonably close?

3. $\boxed{\textbf{Optimal Beamforming: MVDR and GSC}}$

Given the parameters of **Part 2,** compute the MVDR and GSC for each of the sources (i.e., each source is, in turn, taken to be the target with a distortionless responses). For each case, compute the array response, and obtain the 2-D plots and 1-D slices as from the previous problem. Use reasonable densities for the (θ, ϕ) angles. Also, for the MVDR, compute the array response at each of the sources, and create a table with the depth of the nulls (e.g., when steered to source #1, how much are sources #2, and sources #3 suppressed, in dB).

4. $\boxed{\textbf{Adaptive Beamforming}}$

Implement adaptive MVDR and GSC for each of the sources, using both LMS and RLS, taking just the standard case (the source AOAs as originally given, signal and noise powers as given, no correlation). It is your task to pick reasonable μ for LMS (I want you to use the same μ for each of the three sources and for both MVDR and GSC cases: pick a μ so all your algorithms are stable, but try not to make it too small). For RLS, use $\lambda = 0.9$; you can start with $\delta = 1$ but vary it to perhaps improve results. Generate learning curves. Also take one run for the case of steering to source #1, both MVDR and GSC, both LMS and RLS. Let's say in the case you take reasonable convergence seems to occur after N_0 iterations. Compute the beamformer vector after $N_0/2$ and N_0 iterations, and look at the array responses in each case. Also, take the last 10 iterations (i.e., $N_0 - 9 \leq n \leq N_0$), find the beamformer for each instant, and average them to get a new vector; first, why does this vector necessarily still satisfy the constraints; second, look at its array response as well. Make some comments, e.g., does the average vector work better than one chosen from one instant?

5. **Variations**

   (a) **Negative SNR:** Repeat **Problems 2,3,4** where the noise power is $10dB$ **above** the primary source. Comment on the impact this has on the results. If something is performing badly, e.g., you are having problems getting reasonable results with the adaptive algorithms, say so. *The result may very well be things are falling apart; it is your task to recognize if that happens, not necessarily to get it to work.* [The only tweaking you may need to do is perhaps adjusting the LMS step-size μ, or the RLS initialization parameter δ, keep the RLS parameter $\lambda = 0.9$]

   (b) **Fewer snapshots:** You have $M = 21$ sensors in your array, and used $N = 100$ snapshots. Repeat **Problems 2,3,4** with $N = 25$, and again with $N = 10$. [This means keep the original noise power, don't also do this with the higher noise.] Comment on what impact, if any, these reductions have on your results. In practice, people usually prefer to use $N \gtrsim 2M$. With $N = 25$, $N \gtrsim M$, and with $N = 10$ we have $N < M$. In particular, does anything "fail" with $N < M$? One could argue that as long as both $M > L$ and $N > L$, we should still be able to identify the sources. That doesn't mean all of the approaches we are using here will behave reasonably. *Again, if you can't get something to work well, say so.*

   (c) **Correlated sources:** So far, the sources are assumed uncorrelated, and your code is based on starting with their given powers. Now assume they can be correlated. Modify your code to accept correlation coefficients ρ_{ij} among the sources; you could write is so the input is an $L \times L$ matrix of correlation coefficients. For example, for $L = 3$:

$$\boldsymbol{\rho} = \begin{bmatrix} 1 & \rho_{12} & \rho_{13} \\ \rho_{21} & 1 & \rho_{23} \\ \rho_{31} & \rho_{32} & 1 \end{bmatrix}$$

   so that (with σ_ℓ^2 the variance of $\alpha_\ell[n]$) $cov(\alpha_i[n]\,\alpha_j[n]) = \sigma_i \sigma_j \rho_{ij}$. Note that $\rho_{ij} = \rho_{ji}^*$, and each $|\rho_{ij}| \leq 1$, with $\rho_{ii} = 1$. The covariance matrix of the $\boldsymbol{\alpha}[n]$ vector then has the form:

$$C = diag\{\sigma_1, \cdots, \sigma_L\}\ \boldsymbol{\rho}\ diag\{\sigma_1, \cdots \sigma_L\}$$

   As a rough rule of thumb, if $|\rho_{ij}| > 0.1$, the correlation between the quantities is considered significant. After you modify your code, now assume the first two sources are correlated with $\rho_{12} = 0.3$, but each is uncorrelated with the third source. This is a fairly strong degree of correlation. In this case:

$$\boldsymbol{\rho} = \begin{bmatrix} 1 & 0.3 & 0 \\ 0.3 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Repeat **Problems 2,3,4** [with the original parameters, not with the variations described in **Problems 5ab**] and comment on any changes you may notice. [Again, if some of the methods are falling apart, say so]

I want **comments,** not just a dump of 1000 figures. **COMMENT** on your observations! Also, it is fine to have loops, as opposed to trying to pack everything into cells or multidimensional arrays. Loops can provide (1)clarity, and (2)reduce memory requirements.
Every figure should be labeled (axes labeled and a title).