Consider a sinusoidal arriving from angular direction $\theta$. Take first antenna as a reference.

\[ y_0(k) = C \epsilon^{j\omega_0 k} + n_0(k) \]
\[ y_1(k) = C \epsilon^{j(\omega_0 - 90) k} + n_1(k) = C \epsilon^{j\omega_0 k} \cdot \epsilon^{-\pi/4} \sin(\theta) + n_1(k) \]
\[ y_2(k) = C \epsilon^{j(\omega_0 - 180) k} + n_2(k) \]
\[ y_{H-1}(k) = C \epsilon^{j(\omega_0 - (H-1)90) k} + n_{H-1}(k) \]

\[ U(k) = C \epsilon^{j\omega_0 k} \begin{bmatrix} 1 + j \theta \\ e^{-j\theta} \\ \vdots \\ e^{-j(H-1)\theta} \end{bmatrix} + n = C \epsilon^{j\omega_0 k} S(\theta) + n \]

\[ S(\theta) = \begin{bmatrix} 1 \\ e^{-j\theta} \\ \vdots \\ e^{-j(H-1)\theta} \end{bmatrix} - \text{steering vector} \]

The output of the antenna combiner:

\[ y(k) = \omega^H U(k) = \omega^H \left( C \epsilon^{j\omega_0 k} S(\theta) + n \right) \]
\[ y(k) = \omega^H S(\theta) e^H u(k) + \omega^H n(k) \]

The power of the output signal

\[ P = |y|^2 = E[y(k)\cdot \overline{y(k)}] = E[\omega^H u(k) \cdot \overline{u(k)}] = \omega^H \mathbf{R}\omega \]

where \( \mathbf{R} \) is subband matrix for the waves obtained at the output of antenna elements. Note - processing is spatial (not temporal); the correlation is spatial correlation

In general, one has to minimize the power of the output signal subject to constraint

\[ \omega^H S(\Theta_0) = 1 \quad \text{gain in direction } \Theta_0 \text{ is kept as } 1 \]

\[ (\Theta_0 = \frac{2\pi k}{N} \sin(\theta)) \]

The minimization can be conducted through the method of Lagrange multipliers

Minimize \[ J(\omega) = \omega^H \mathbf{R} \omega \]

subject to \( \omega^H S(\Theta_0) = 1 \)

Form Lagrangean function

\[ L(\omega) = \omega^H \mathbf{R} \omega + \lambda (\omega^H S(\Theta) - 1) \]

\[ \lambda = \frac{1}{2} \]
\[ \frac{\partial J(\omega, \gamma)}{\partial \omega} = 2R_0 \omega + \gamma S(\gamma) = 0 \quad \omega = \omega_{opt} \]

or

\[ \omega_{opt} = - \frac{\gamma}{2} R^{-1} S(\gamma) \quad (\star) \]

\[ \frac{\partial J(\omega, \gamma)}{\partial \gamma} = \omega_{opt}^H S(\gamma) - 1 = 0 \]

Taking Hermitian transpose of (\star) and postmultiplying with \( S(\gamma) \), one obtains

\[ \omega_{opt}^H S(\gamma) = - \frac{\gamma}{2} \omega_{opt}^H R^{-1} S(\gamma) \]

\[ \lambda = - \frac{2 \omega_{opt}^H S(\gamma)}{S^H(\gamma_0) R^{-1} S(\gamma_0)} = \frac{-2}{S^H(\gamma_0) R^{-1} S(\gamma_0)} \quad (\star \star) \]

Substituting (\star \star) into (\star)

\[ \omega_{opt} = - \frac{R^{-1} S(\gamma_0)}{S^H(\gamma_0) R^{-1} S(\gamma_0)} - \text{optimum weights} \quad (\star \star) \]

This gives explicit solution for optimum weights.

\[ \mathbf{w} \] is a vector of the subspace through which we would like to design an LMS-based iterative procedure to accomplish the least-squares minimization.

What we are trying to do

\[ \min_{\omega} J(\omega) = \omega^H R_0 \omega \quad \text{subject to } \omega^H S(\gamma_0) = 1 \]

Unconstrained minimization

\[ SD \quad \omega(k+1) = \omega(k) - \mu \frac{\partial J(\omega)}{\partial \omega} = \omega(k) - 2 \mu R^{-1} \omega(k) \]

\[ LMS \quad \omega(k+1) = \omega(k) - 2 \mu R^{-1} \omega(k) = \omega(k) - 2 \mu \omega(k)^H \omega(k), \omega(k) \]
However, both learning rules would drive $w$ to $0$. To prevent this from happening, we introduce explicit normalization after each step.

$$w(k+1) = \frac{w(k)}{w(k)^T \delta(k)}$$

**Summary (LMS algorithm)**

1. Initialize $w$ to all zeros (or some random numbers).
2. Compute $\delta(k)$.
3. Update $w(k)$:
   $$w(k+1) = w(k) - \mu(k) \delta(k) y^H(k) \text{ and } w(k) = \left[ I - \mu(k) y(k) y^H(k) \right] w(k)$$
4. If $w(k+1)$ is not converged, go back to step 2.

**Example:** Consider an array with 4 elements ($i.e. N=4$)

Assume $\psi_0 = 0$, $\theta_0 = \frac{2\pi}{\lambda} L \sin(\psi_0) = 0$

$$\psi = \frac{\lambda}{2}, \quad n_i(\psi) = 0 \quad \delta_n = 0.1, \quad c = 1$$

Analyze the case when $\psi = \frac{\pi}{4}$, $\theta_i = \frac{2\pi L \sin(\psi)}{\lambda} = \frac{\pi \sqrt{2}}{2}$.
The optimum weights are given as.

$$\omega_{opt} = \frac{R^{-1} \Sigma(\theta_0)}{\Sigma'(\theta_0) \cdot R^{-1} \cdot \Sigma(\theta_0)}$$

$$\theta_0 = 0, \Sigma(\theta_0) = \left[ \begin{array}{cccc} 1 & e^{-j\Phi_0} & e^{-j2\Phi_0} & e^{-j3\Phi_0} \end{array} \right]$$

$$= \left[ \begin{array}{cccc} 1 & 1 & 1 & 1 \end{array} \right]^T$$

$$R = E_x y(x) y^H(x)^T = E_x [c e^{j\omega_0 k} \cdot S(\theta_i) + \eta \cdot [c e^{j\omega k} \cdot S'(\theta_i) + \eta H]^T]^T = E_x [c e^{j\omega_0 k} \cdot S(\theta_i) + \eta \cdot [c e^{j\omega k} \cdot S'(\theta_i) + \eta H]^T]^T$$

$$= E_x [c^2 S(\theta_i) S'(\theta_i) + c e^{j\omega_0 k} \cdot \eta S'(\theta_i) + c e^{j\omega k} S(\theta_i) \eta H + \eta H H]^T$$

$$= c^2 S(\theta_i) S'(\theta_i) + \eta^2 I = C^2 \Sigma + \eta^2 I = \Sigma_i + \eta \Sigma_i$$

$$S(\theta_i) = \left[ \begin{array}{cccc} 1 & e^{-j\Phi_i} & e^{-j2\Phi_i} & e^{-j3\Phi_i} \end{array} \right]^T$$

$$= \left[ \begin{array}{cccc} 1 & e^{-j\Phi_i/2} & e^{-j\Phi_i} & e^{-j3\Phi_i/2} \end{array} \right]^T$$

Spend the rest of the class analyzing MATLAB code for above example.

Homework: type up the MATLAB code
- generate plots for $\Phi = 120^\circ (2\pi/3)$
- experiment with different $M$. 