Sensor Array Signal Processing

Prof. Rodrigo C. de Lamare
CETUC, PUC-Rio, Brazil
and
Department of Electronic Engineering, University of York, UK
delamare@cetuc.puc-rio.br

Syllabus

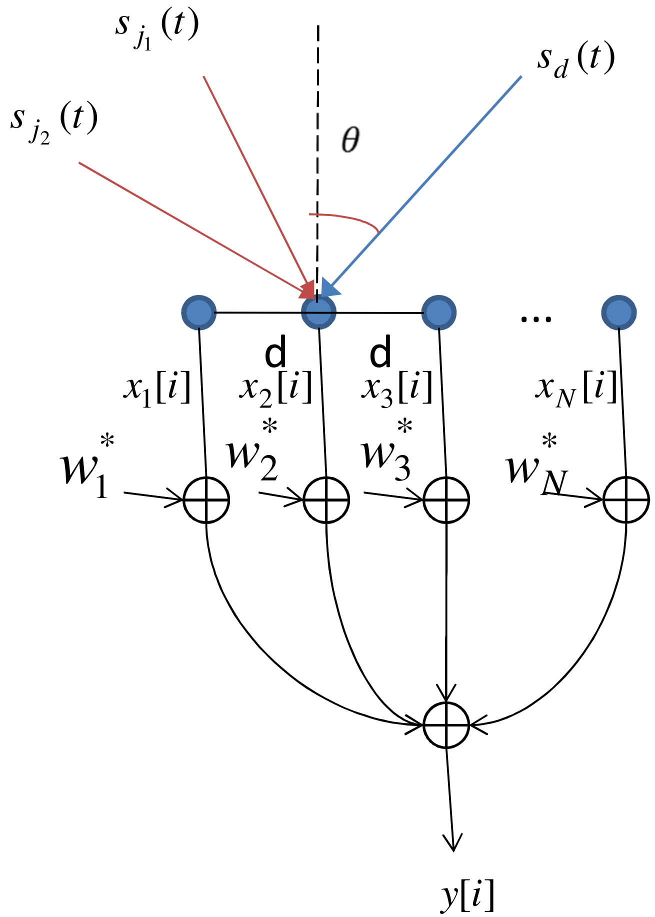
I. Introduction
- Fundamentals
- Sensor arrays
- Discrete-time models

II. Beamforming
- Main principles
- Optimum beamforming
- Robust beamforming
- Adaptive algorithms

III. Direction finding
- Maximum likelihood estimation
- Cramèr-Rao lower bound
- Capon´s technique
- MUSIC
- ESPRIT

II. Beamforming

- Aims:
  - To extract information from signals received by an array of sensors.
  - To enhance the quality of the signals of interest and mitigate interference.



- Information of interest:
  - Signal content. Ex: communications.
  - Signal existence. Ex: radar and sonar.
- Beamforming principles:
  - Linear combination of signals received by an array of sensors.
  - Signals are weighted such that the signals of interest are enhanced and the interference is mitigated.

A. Fundamentals

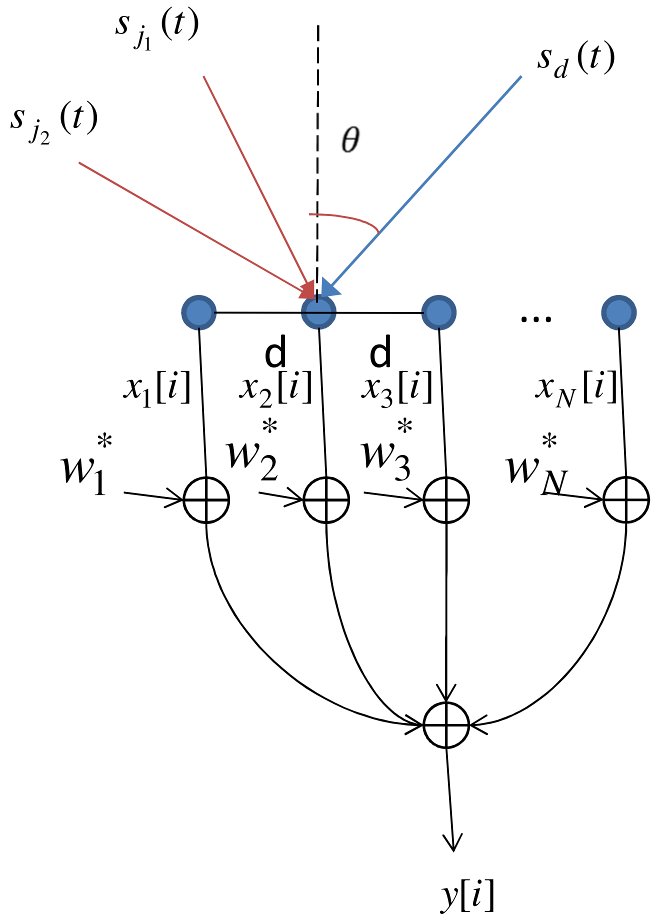
- Beamforming attenuates undesired signals and enhances the signals of interest via weighting.
- In particular, the beamformer applies the following weights:

$$w_n, \quad n = 1,2,\cdots,N$$

- The output of a beamformer with N weights produces a linear combination of the signals from N sensors given by

$$y[i] = \sum_{n=1}^{N} w_n^* x_n[i]$$
$$= \mathbf{w}^H \mathbf{x}[i]$$

where $\boldsymbol{w} = [w_1 w_2 \dots w_N]^T$ is a vector of weights that must be computed by a signal processing algorithm



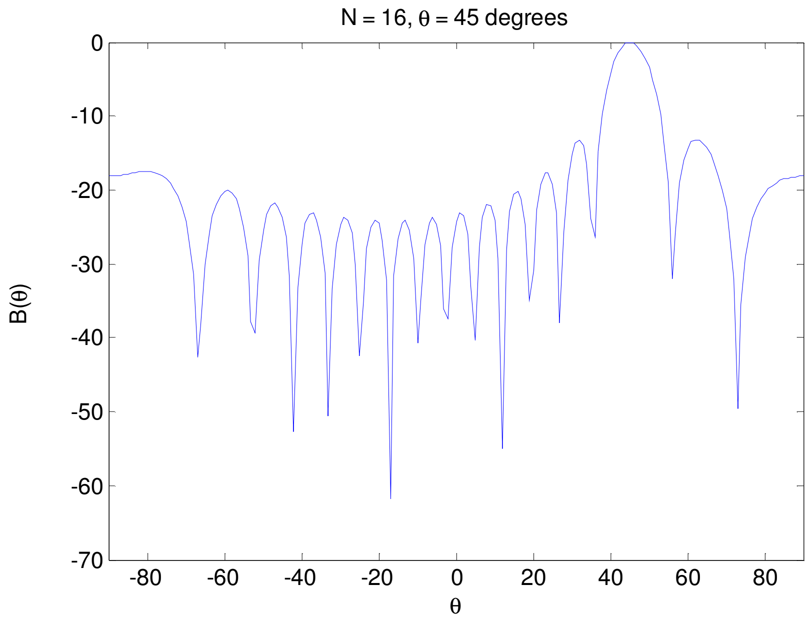
Beampattern:

- The beampattern can be computed by applying the beamforming weights to a set of steering vectors comprising the angles of interest.

- The beampattern function is described by

$$B(\theta) = \boldsymbol{w}^H \boldsymbol{a}(\theta),$$

where $\theta_{\min} \leq \theta \leq \theta_{\max}$ and the step size in degrees can be chosen.

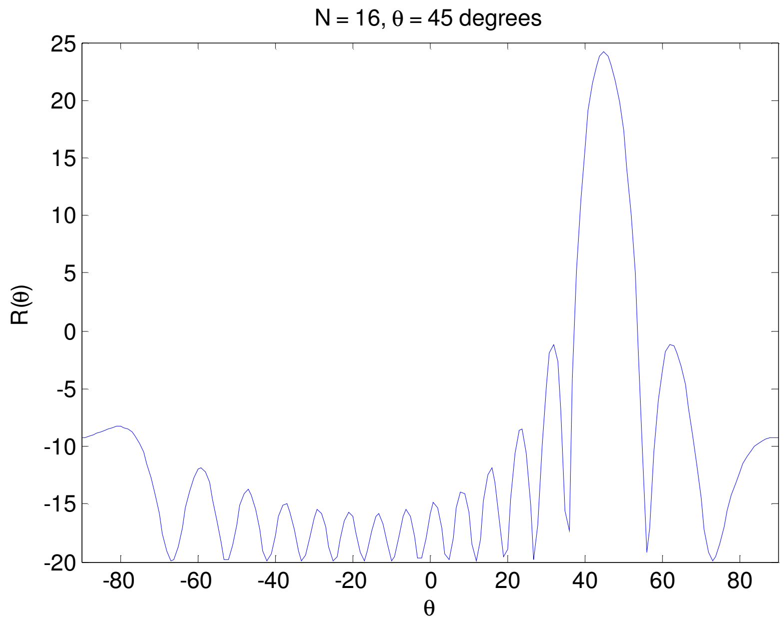


Spatial power spectrum:

- The spatial power spectrum can be obtained by

$$R(\theta) = E[|\boldsymbol{w}^H(\theta)\boldsymbol{x}[i]|^2],$$

where the angle dependent beamformer $w^H(\theta)$ determines the spatial spectrum.



Beamforming gain:

- To assess the beamforming gain, we evaluate the power of the signal and the noise at the input and output of the beamformer.

- Let us consider the signal model of a sensor array with N sensors in the absence of interference:

$$\boldsymbol{x}[i] = s[i]\boldsymbol{a}(\theta) + \boldsymbol{n}[i], \in \mathbb{C}^{N}$$

- The output of the beamformer is given by

$$\begin{aligned} y[i] &= \boldsymbol{w}^H \boldsymbol{x}[i] \\ &= s[i]\boldsymbol{w}^H \boldsymbol{a}(\theta) + \boldsymbol{w}^H \boldsymbol{n}[i] \end{aligned}$$

Beamforming gain:

- The power at the output of the beamformer is given by

$$P_y = E[|y[i]|^2] = \boldsymbol{w}^H \boldsymbol{R} \boldsymbol{w}$$

where $\boldsymbol{R} = E[\boldsymbol{x}[i]\boldsymbol{x}^{\boldsymbol{H}}[i]]$ is the covariance matrix of $\boldsymbol{x}[i]$.

- The signal-to-noise ratio (SNR) at the output of each sensor element and prior to the beamformer is given by

$$SNR_e \triangleq \frac{E[|s[i]|^2]}{E[[|n[i]|^2]} = \frac{\sigma_{\boldsymbol{s}}^{\boldsymbol{2}}}{\sigma_{\boldsymbol{n}}^{\boldsymbol{2}}}$$

- The above result is the SNR of a single sensor, which does not exploit spatial diversity or different noise realizations.

Beamforming gain:

- The signal power at the output of the beamformer is given by

$$P_s = E[|\mathbf{w}^H \mathbf{a}(\theta) s[i]\ |^2] = E[\mathbf{w}^H \mathbf{a}(\theta) \mathbf{a}^{\mathbf{H}}(\theta) \mathbf{w} |s[i]|^2]$$
$$= \sigma_{\mathbf{s}}^{\mathbf{2}} \mathbf{w}^H \mathbf{a}(\theta) \mathbf{a}^H(\theta) \mathbf{w}$$

- The noise power at the output of the beamformer is given by

$$P_n = E[|\mathbf{w}^H \mathbf{n}[i]\ |^2] = E[\mathbf{w}^H \mathbf{n}[i] \mathbf{n}^{\mathbf{H}}[i]\ \mathbf{w}]$$
$$= \sigma_{\mathbf{n}}^{\mathbf{2}} \mathbf{w}^H \mathbf{w}$$

- The SNR at the output of the beamformer is given by

$$SNR_a \triangleq \frac{E[|\mathbf{w}^H \mathbf{a}(\theta) s[i]\ |^2]}{E[|\mathbf{w}^H \mathbf{n}[i]\ |^2]]} = \frac{P_s}{P_n}$$
$$= \frac{\sigma_s^2 \mathbf{w}^H \mathbf{a}(\theta) \mathbf{a}^H(\theta) \mathbf{w}}{\sigma_n^2 \mathbf{w}^H \mathbf{w}} = SNR_a \frac{\mathbf{w}^H \mathbf{a}(\theta) \mathbf{a}^H(\theta) \mathbf{w}}{\mathbf{w}^H \mathbf{w}}$$

Beamforming gain:

Beamforming (or sensor array) gain:

- The beamforming gain is described by

$$G \triangleq \frac{SNR_a}{SNR_e} = \frac{\boldsymbol{w}^H \boldsymbol{a}(\theta)\boldsymbol{a}^H(\theta)\boldsymbol{w}}{\boldsymbol{w}^H \boldsymbol{w}}$$

- In the particular case of a ULA with N sensors and $\boldsymbol{w} = \boldsymbol{a}(\theta)$ the beamforming gain is described by

$$G \triangleq \frac{\boldsymbol{w}^H \boldsymbol{a}(\theta)\boldsymbol{a}^H(\theta)\boldsymbol{w}}{\boldsymbol{w}^H \boldsymbol{w}} = N,$$

where we have exploited the fact that $\boldsymbol{a}^H(\theta)\boldsymbol{a}(\theta) = N$.

- The beamformer $\boldsymbol{w} = \boldsymbol{a}(\theta)$ is known as the conventional beamformer or delay-and-sum beamformer.

Important notes:

- For spatially white noise, the conventional beamformer is optimal in the sense that it maximizes the SNR.
- In the presence of interference, we need beamformers which maximize the signal-to-interference-plus-noise ratio.
- Beamformers must often be adaptive in order to address the statistical behavior of the interference.

Important notes:

- The spacing between sensor elements should be greater or equal to $\lambda/2$ to avoid spatial ambiguities.

  - Example with $f_c = 10GHz$: $\lambda = 3$ cm (X band in radar systems)
  - Example with $f_c = 1GHz$: $\lambda = 30$ cm (communication systems)

- The aperture, known as the area used to gather the signals should be maximized.

B. Optimum beamforming

Principles:

- Methods that employ the statistics of the sensor array to compute the beamforming weights.
- Optimum beamformers employ some criterion (MVDR, SINR, etc) based on the statistical knowledge of data.
- Adaptive beamformers employ estimates of the statistics of the data.

Signal model

- The discrete-time signal observed by the sensor array is

$$\boldsymbol{x}[i] = \underbrace{s_d[i]\boldsymbol{a_d}(\theta)}_{\boldsymbol{s}[i]} + \underbrace{\sum_{k=1,k\neq d}^{D} s_k[i]\boldsymbol{a_k}(\theta)}_{\text{interference } \boldsymbol{j}[i]} + \boldsymbol{n}[i], \in \mathbb{C}^{N}$$

where $s_d[i]$ is the desired signal and $E[|s_d[i]|^2] = \sigma_d{}^2$.

- The signal component with interference and noise is given by

$$\boldsymbol{x_{j+n}}[i] = \boldsymbol{j}[i] + \boldsymbol{n}[i], \in \mathbb{C}^{N}$$

which is often modeledas a wide-sense stationary stochastic process with zero mean.

Statistical characterization

- The covariance matrix of the received signal is given by

$$\boldsymbol{R} = E\left[\boldsymbol{x}[i]\boldsymbol{x}^H[i]\right] = {\sigma_d}^2\boldsymbol{a_d}(\theta){\boldsymbol{a_d}}^{\boldsymbol{H}}(\theta) + \boldsymbol{R}_j + \boldsymbol{R_n},$$

where $\boldsymbol{R}_j = E\left[\boldsymbol{j}[i]\boldsymbol{j}^H[i]\right]$ is the interference covariance matrix and $\boldsymbol{R_n} = E\left[\boldsymbol{n}[i]\boldsymbol{n}^H[i]\right]$ is the covariance of the noise.

- The covariance matrix of the interference plus noise is given by

$$\boldsymbol{R} = E\left[\boldsymbol{x_{j+n}}[i]{\boldsymbol{x_{j+n}}}^H[i]\right] = \boldsymbol{R}_j + \boldsymbol{R_n}.$$

- When the noise is spatially white with variance ${\sigma_n}^2$, we have

$$\boldsymbol{R_n} = E\left[\boldsymbol{n}[i]\boldsymbol{n}^H[i]\right] = {\sigma_n}^2\boldsymbol{I}$$

Design of the optimum beamformer

- Let us first consider the maximization of the SINR at the output of each sensor element as expressed by

$$\boldsymbol{w_o} = \arg\ \max\ \mathrm{SINR}_e\ ,$$

where the SINR_e at each sensor is given by

$$\mathrm{SINR}_e = \frac{{\sigma_s}^2}{{\sigma_j}^2+{\sigma_n}^2},$$

with ${\sigma_s}^2$, ${\sigma_j}^2$ and ${\sigma_n}^2$ being the powers of the signal, inerference and noise.

Design of the optimum beamformer

- Let us now consider the maximization of the SINR at the output of the beamformer as expressed by

$$\boldsymbol{w_o} = \arg\ \max\ \mathrm{SINR}_{out}\ ,$$

where the SINR_{out} at the output of the beamformer is given by

$$\mathrm{SINR}_{out} = \frac{E\left[\boldsymbol{w}^H s[i]\boldsymbol{s}^H[\boldsymbol{i}]\boldsymbol{w}\right]}{E\left[\boldsymbol{w}^H \boldsymbol{x}_{j+n}[i]\boldsymbol{x}_{j+n}{}^H[\boldsymbol{i}]\boldsymbol{w}\right]} = \frac{\boldsymbol{w}^H\boldsymbol{R}_S\boldsymbol{w}}{\boldsymbol{w}^H\boldsymbol{R}_{j+n}\boldsymbol{w}},$$

where $\boldsymbol{R}_s = E\left[\boldsymbol{s}[i]\boldsymbol{s}^H[i]\right]$ is the covariance matrix of the desired signal $\boldsymbol{s}[i] = s_d[i]\boldsymbol{a_d}(\theta)$.

MVDR beamforming

- An equivalent solution: to minimize the interference plus noise at the output of the beamformer subject to $\boldsymbol{w}^H \boldsymbol{R}_s \boldsymbol{w}$ being constant.

- Design of MVDR beamformer:

$$\boldsymbol{w_o} = \arg\min \boldsymbol{w}^H \boldsymbol{R}_{j+n} \boldsymbol{w}, \qquad \text{subject to } \boldsymbol{w}^H \boldsymbol{a}(\theta_d) = 1$$

- The above solution is known as minimum variance distortionless response (MVDR) beamformer.

- The MVDR beamformer maximizes the SINR matching the signal response to the sensor array to the direction θ_d.

- The MVDR solution can be considered an optimal spatial filter.

MVDR beamforming: practical aspects

- A key problem with the MVDR beamformer is that it requires $\boldsymbol{R}_{j+n}$, which is difficult to obtain in practice.

- In practice, we employ $\boldsymbol{R}$ instead of $\boldsymbol{R}_{j+n}$.

- Alternative MVDR Design:

$$\boldsymbol{w_o} = \arg\min \; \boldsymbol{w}^H \boldsymbol{R} \boldsymbol{w}, \qquad \text{subject to } \boldsymbol{w}^H \boldsymbol{a}(\theta_d) = 1$$

- The above design is sometimes called minimum power distortionless response (MPDR) beamformer.

Derivation of MVDR beamformer

- Consider the constrained optimization given by

$$\boldsymbol{w_o} = \arg\ \min\ \boldsymbol{w}^H \boldsymbol{R}\boldsymbol{w}, \qquad \text{subject to } \boldsymbol{w}^H \boldsymbol{a}(\theta_d) = 1$$

- Using the method of Lagrange multipliers, we can transform the constrained optimization above into an unconstrained one.

- The Lagrangian function is given by

$$\mathcal{L}(\boldsymbol{w}, \lambda) = \boldsymbol{w}^H \boldsymbol{R}\boldsymbol{w} + \lambda(\boldsymbol{w}^H \boldsymbol{a}(\theta_d) - 1) + (\boldsymbol{a}^H(\theta_d)\boldsymbol{w} - 1)\lambda^*$$

Derivation of MVDR beamformer

- Let us compute the gradients of the Lagrangian given by

$$\mathcal{L}(\boldsymbol{w},\lambda) = \boldsymbol{w}^H\boldsymbol{R}\boldsymbol{w} + \lambda(\boldsymbol{w}^H\boldsymbol{a}(\theta_d) - 1) + (\boldsymbol{a}^H(\theta_d)\boldsymbol{w} - 1)\lambda^*,$$

which yield

$$\nabla\mathcal{L}(\boldsymbol{w},\lambda)_{\boldsymbol{w}^*} = \boldsymbol{R}\boldsymbol{w} + \lambda\boldsymbol{a}(\theta_d)$$

$$\nabla\mathcal{L}(\boldsymbol{w},\lambda)_{\lambda} = \boldsymbol{w}^H\boldsymbol{a}(\theta_d) - 1$$

- Equating the terms to zero and solving for $\boldsymbol{w}$, we obtain

$$\boldsymbol{w_o} = \frac{\boldsymbol{R}^{-1}\boldsymbol{a}(\theta_d)}{\boldsymbol{a}^H(\theta_d)\boldsymbol{R}^{-1}\boldsymbol{a}(\theta_d)}$$

- The minimum variance can be obtained by substituting $\boldsymbol{w_o}$ in the cost function $J(\boldsymbol{w}) = \boldsymbol{w}^H\boldsymbol{R}\boldsymbol{w}$, which yields

$$J(\boldsymbol{w_o}) = \boldsymbol{w_o}^H\boldsymbol{R}\boldsymbol{w_o}$$

$$= \frac{1}{\boldsymbol{a}^H(\theta_d)\boldsymbol{R}^{-1}\boldsymbol{a}(\theta_d)}$$

where $J(\boldsymbol{w_o})$ is known as the spatial power spectrum.

Practical computation of the MVDR beamformer

- In practice, $\boldsymbol{R}$ is unknown and must be estimated by:

$$\widehat{\boldsymbol{R}}[i] = \frac{1}{i}\sum_{l=1}^{i} \boldsymbol{x}[l]\boldsymbol{x}^H[l],$$

where i refers to the number of snapshots or data support.

- Substituting $\widehat{\boldsymbol{R}}[i]$in to the expression of the MVDR beamformer, we obtain the solution known as sample matrix inversion (SMI):

$$\boldsymbol{w}_{\text{SMI}} = \frac{\widehat{\boldsymbol{R}}^{-1}[i]\boldsymbol{a}(\theta_d)}{\boldsymbol{a}^H(\theta_d)\widehat{\boldsymbol{R}}^{-1}[i]\boldsymbol{a}(\theta_d)},$$

- The SINR loss of the SMI solution is given by

$$L_{\text{SMI}} = \frac{SINR_{\text{SMI}}}{SINR_{\text{o}}} = \frac{{\boldsymbol{w}_{\text{SMI}}}^H\boldsymbol{R}_s\boldsymbol{w}_{\text{SMI}}}{{\boldsymbol{w}_{\text{SMI}}}^H\boldsymbol{R}_{j+n}\boldsymbol{w}_{\text{SMI}}} \cdot \frac{{\boldsymbol{w}_{\text{o}}}^H\boldsymbol{R}_s\boldsymbol{w}_{\text{o}}}{{\boldsymbol{w}_{\text{o}}}^H\boldsymbol{R}_{j+n}\boldsymbol{w}_{\text{o}}},$$

which corresponds to 3dB for i=2N.

C. Robust techniques

- Problems with the MVDR solution:
  - Signal mismatch $\hat{\boldsymbol{a}}(\theta_d) \neq \boldsymbol{a}(\theta_d)$ due to estimation errors, sensor imperfections, calibration errors, etc
  - Short data records or insufficient number of snapshots to obtain $\hat{\boldsymbol{R}}\,[i]$.

- Modeling of signal mismatch:

$$\hat{\boldsymbol{a}}(\theta_d) = \boldsymbol{a}(\theta_d) + \boldsymbol{e}$$

where $\boldsymbol{e}$ is a vector with errors

- The error vector $\boldsymbol{e}$ can be modeled as coherent local scattering, incoherent scattering and as randomly generated.

Robust design of beamformers

- Robust designs: effective approaches to mitigating signal mismatches and short data records.

- The first robust approach: diagonal loading.

- Main principles of diagonal loading:

    - Insertion of a diagonal loading of the type $\epsilon \boldsymbol{I}$ in $\widehat{\boldsymbol{R}}\,[i]$.
    - Reduction of the eigenvalue spread.
    - Improvement of the beampattern.

H. Cox, R. M. Zeskind, and M. H. Owen, "Robust adaptive beamforming," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-35, pp. 1365–1376, Oct. 1987

Robust design of beamformers

- Let us consider a robust design based on diagonal loading:

$$\boldsymbol{w}_{\boldsymbol{RO}} = \arg\min \ \boldsymbol{w}^H \boldsymbol{R} \boldsymbol{w} + \epsilon \boldsymbol{w}^H \boldsymbol{w}, \quad \text{subject to } \boldsymbol{w}^H \hat{\boldsymbol{a}}(\theta_d) = 1,$$

where ${\sigma^2}_n \leq \epsilon \leq 10{\sigma^2}_n$.

- Consider the method of Lagrange multipliers and the Lagrangian function given by

$$\mathcal{L}(\boldsymbol{w}, \lambda) = \boldsymbol{w}^H \boldsymbol{R} \boldsymbol{w} + \epsilon \boldsymbol{w}^H \boldsymbol{w} + \lambda(\boldsymbol{w}^H \hat{\boldsymbol{a}}(\theta_d) - 1) + \left(\hat{\boldsymbol{a}}^{\boldsymbol{H}}(\theta_d)\boldsymbol{w} - 1\right)\lambda^*,$$

Robust design of beamformers

- Let us compute the gradients of the Lagrangian given by

$$\mathcal{L}(\boldsymbol{w},\lambda) = \boldsymbol{w}^H \boldsymbol{R}\boldsymbol{w} + \epsilon \boldsymbol{w}^H \boldsymbol{w} + \lambda(\boldsymbol{w}^H \hat{\boldsymbol{a}}(\theta_d) - 1) + \left(\hat{\boldsymbol{a}}^{\boldsymbol{H}}(\theta_d)\boldsymbol{w} - 1\right)\lambda^*,$$

,

and equate the the terms to zero which yields

$$\nabla \mathcal{L}(\boldsymbol{w},\lambda)_{\boldsymbol{w}^*} = \boldsymbol{R}\boldsymbol{w} + \epsilon \boldsymbol{w} + \lambda \boldsymbol{a}(\theta_d) = \boldsymbol{0}$$

$$\Longrightarrow \boldsymbol{w} = (\boldsymbol{R} + \epsilon \boldsymbol{I})^{-\boldsymbol{1}}(-\lambda)\ \hat{\boldsymbol{a}}(\theta_d)$$

$$\nabla \mathcal{L}(\boldsymbol{w},\lambda)_{\lambda} = \boldsymbol{w}^H \hat{\boldsymbol{a}}(\theta_d) - 1 = \boldsymbol{0}$$

$$\Longrightarrow \boldsymbol{w}^H \hat{\boldsymbol{a}}(\theta_d) = 1$$

Robust design of beamformers

- Substituting $\boldsymbol{w}$ in $\boldsymbol{w}^H\hat{\boldsymbol{a}}(\theta_d) - 1$, we have

$$\lambda = -\left(\hat{\boldsymbol{a}}^{\boldsymbol{H}}(\theta_d)(\boldsymbol{R} + \epsilon\boldsymbol{I})^{-\boldsymbol{1}}\hat{\boldsymbol{a}}(\theta_d)\right),^{-1},$$

- Substituting λ in $\boldsymbol{w}$, we obtain the expression for the robust beamformer with diagonal loading:

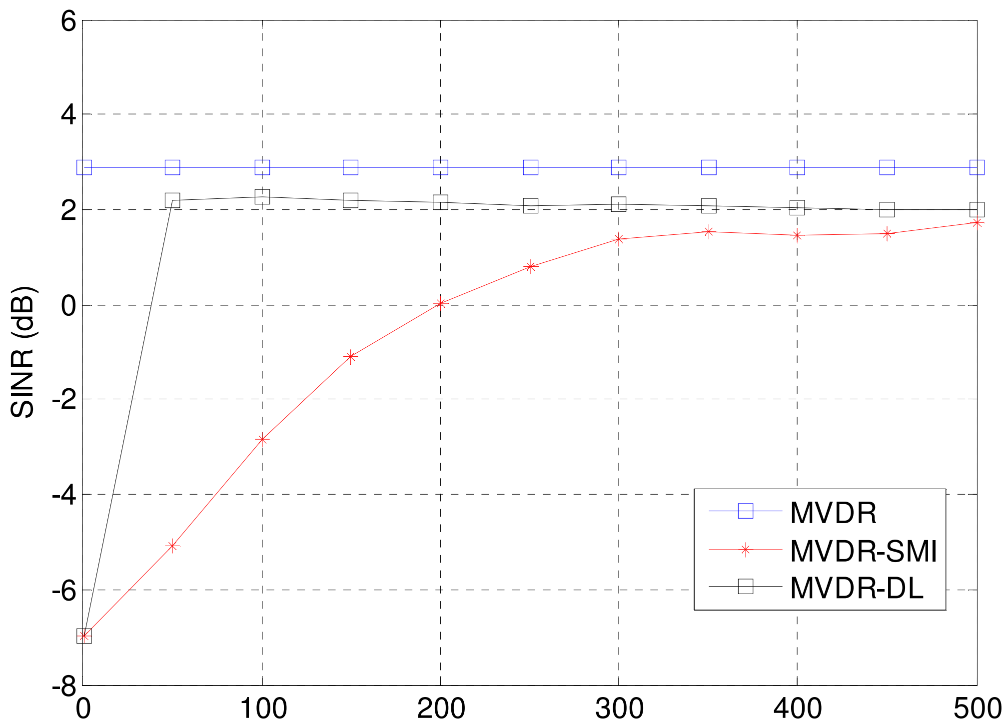
$$\boldsymbol{w}_{RO} = \frac{(\boldsymbol{R}+\epsilon\boldsymbol{I})^{-\boldsymbol{1}}\,\hat{\boldsymbol{a}}(\theta_d)}{\hat{\boldsymbol{a}}^{\boldsymbol{H}}(\theta_d)(\boldsymbol{R}+\epsilon\boldsymbol{I})^{-\boldsymbol{1}}\hat{\boldsymbol{a}}(\theta_d)},$$

where ϵ needs to be adjusted.

- Since there is no closed form solution for ϵ techniques that can automatically adjust ϵ are of interest to designers.

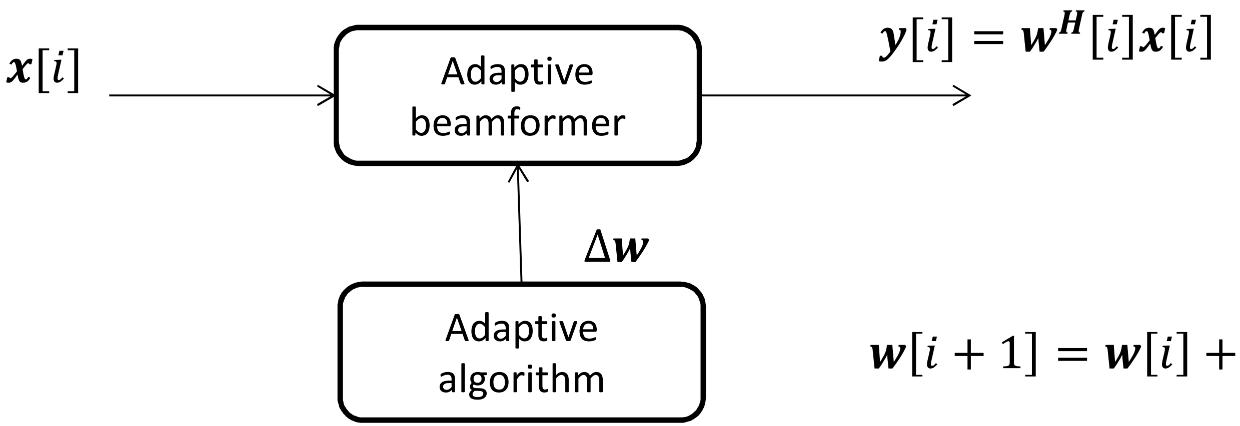
Example: Robust beamforming

- N = 10 sensors
- SNR = 5 dB
- D = 3 interfering signals
- Signal mismatch:
  - generated with Gaussian random variables
  - Variance equal to 10% of the noise variance.
- Diagonal loading (DL):
  - 10 x noise variance



D. Adaptive algorithms

- When the statistics of the sensor data change the use of adaptive algorithms to compute the MVDR solution is desirable.
- The main idea is to employ a recursive algorithm to iteratively compute the set of parameters of the beamformer.



$$\boldsymbol{w}[i+1] = \boldsymbol{w}[i] + \Delta\boldsymbol{w}$$

MVDR-LMS algorithm

- Consider the following optimization problem:

$$\hat{\boldsymbol{w}}\,[i] = \arg\,\min_{\boldsymbol{w}[i]}\ \boldsymbol{w}^H[i]\ \underbrace{\hat{\boldsymbol{R}}[i]}_{\boldsymbol{x}[i]\boldsymbol{x}^H[i]}\ \boldsymbol{w}[i], \qquad \text{subject to } \boldsymbol{w}^H[i]\hat{\boldsymbol{a}}(\theta_d) = 1.$$

- Using the method of Lagrange multipliers, we have

$$\mathcal{L}(\boldsymbol{w}[i], \lambda) = \boldsymbol{w}^H[i]\boldsymbol{x}[i]\boldsymbol{x}^H[i]\boldsymbol{w}[i] + \lambda(\boldsymbol{w}^H[i]\hat{\boldsymbol{a}}(\theta_d) - 1) + \left(\hat{\boldsymbol{a}}^{\boldsymbol{H}}(\theta_d)\boldsymbol{w}[i] - 1\right)\lambda^*,$$

- Computing the gradient terms of the Lagrangian, we obtain

$$\nabla\mathcal{L}(\boldsymbol{w}, \lambda)_{\boldsymbol{w}^*} = \boldsymbol{x}[i]\boldsymbol{x}^H[i]\boldsymbol{w}[i] + \lambda\hat{\boldsymbol{a}}(\theta_d)$$

$$\nabla\mathcal{L}(\boldsymbol{w}, \lambda)_{\lambda} = \boldsymbol{w}^H[i]\hat{\boldsymbol{a}}(\theta_d) - 1$$

MVDR-LMS algorithm

- Let us now resort to a gradient descent rule for computing $\boldsymbol{w}[i]$, which yields

$$\hat{\boldsymbol{w}}[i+1] = \hat{\boldsymbol{w}}[i] - \mu \nabla \mathcal{L}(\boldsymbol{w}, \lambda)_{\boldsymbol{w}^*}$$
$$= \hat{\boldsymbol{w}}[i] - \mu\big(\boldsymbol{x}[i]\boldsymbol{x}^H[i]\hat{\boldsymbol{w}}[i] + \lambda \hat{\boldsymbol{a}}(\theta_d)\big),$$

where μ is the step size.

- Substituting the above recursion into the contraint $\hat{\boldsymbol{w}}^H[i+1]\hat{\boldsymbol{a}}(\theta_d) = 1$, we obtain

$$\underbrace{\hat{\boldsymbol{w}}^H[i]\hat{\boldsymbol{a}}(\theta_d)}_{1} - \mu\big(\boldsymbol{x}[i]\boldsymbol{x}^H[i]\hat{\boldsymbol{w}}[i] + \lambda \hat{\boldsymbol{a}}(\theta_d)\big)^H \hat{\boldsymbol{a}}(\theta_d) = 1$$
$$-\mu \hat{\boldsymbol{w}}^H[i]\boldsymbol{x}[i]\boldsymbol{x}^H[i]\hat{\boldsymbol{a}}(\theta_d) - \lambda^* \mu \hat{\boldsymbol{a}}^H(\theta_d)\hat{\boldsymbol{a}}(\theta_d) = 0$$
$$\lambda^* = (-)\big(\hat{\boldsymbol{a}}^H(\theta_d)\hat{\boldsymbol{a}}(\theta_d)\big)^{-1} \hat{\boldsymbol{w}}^H[i]\boldsymbol{x}[i]\boldsymbol{x}^H[i]\hat{\boldsymbol{a}}(\theta_d)$$

MVDR-LMS algorithm

- Substituting λ^* into the recursion for $\boldsymbol{w}[i]$, we obtain

$$\begin{aligned}\widehat{\boldsymbol{w}}[i+1] &= \widehat{\boldsymbol{w}}[i] \\ &\quad - \mu\Bigg(\boldsymbol{x}[i]\boldsymbol{x}^H[i]\widehat{\boldsymbol{w}}[i] \\ &\quad - \left(\widehat{\boldsymbol{a}}^H(\theta_d)\widehat{\boldsymbol{a}}(\theta_d)\right)^{-1}\widehat{\boldsymbol{a}}(\theta_d)\widehat{\boldsymbol{a}}^H(\theta_d)\boldsymbol{x}[i]\underbrace{\boldsymbol{x}^H[i]\widehat{\boldsymbol{w}}[i]}_{y^*[i]}\Bigg) \\ &= \widehat{\boldsymbol{w}}[i] - \mu y^*[i]\left(\boldsymbol{I} - \left(\widehat{\boldsymbol{a}}^H(\theta_d)\widehat{\boldsymbol{a}}(\theta_d)\right)^{-1}\widehat{\boldsymbol{a}}(\theta_d)\widehat{\boldsymbol{a}}^H(\theta_d)\right)\boldsymbol{x}[i] \\ &= \widehat{\boldsymbol{w}}[i] - \mu y^*[i]\boldsymbol{P}\boldsymbol{x}[i],\end{aligned}$$

where $\boldsymbol{P} = \left(\boldsymbol{I} - \left(\widehat{\boldsymbol{a}}^H(\theta_d)\widehat{\boldsymbol{a}}(\theta_d)\right)^{-1}\widehat{\boldsymbol{a}}(\theta_d)\widehat{\boldsymbol{a}}^H(\theta_d)\right) \in \mathbb{C}^N$ is a projection matrix which enforces the contraint.

Statistical analysis of the MVDR-LMS algorithm

- Very similar to the standard LMS algorithm.

- Stability condition:

$$0 \leq \mu \leq \frac{2}{\mathrm{tr}\left[\boldsymbol{R}_p\right]},$$

where $\boldsymbol{R}_p = \boldsymbol{P}^H \boldsymbol{R} \boldsymbol{P}$.

- The MSE is given by

$$\mathrm{MSE}[i] = \zeta_{\min} + \zeta_{\min} \frac{\frac{\mu}{2}\mathrm{tr}\left[\boldsymbol{R}_p\right]}{1-\frac{\mu}{2}\mathrm{tr}\left[\boldsymbol{R}_p\right]},$$

where $\zeta_{\min} = \frac{1}{\boldsymbol{a}^H(\theta_d)\hat{\boldsymbol{R}}^{-1}[i]\boldsymbol{a}(\theta_d)}$

Summary of the MVDR-LMS algorithm

Initialization:

N - number of sensors

$\hat{\boldsymbol{w}}[0] = \frac{\boldsymbol{a}(\theta_d)}{N}$

μ - step size

$$\boldsymbol{P} = \left(\boldsymbol{I} - (\hat{\boldsymbol{a}}^H(\theta_d)\hat{\boldsymbol{a}}(\theta_d))^{-1}\hat{\boldsymbol{a}}(\theta_d)\hat{\boldsymbol{a}}^H(\theta_d)\right)$$

Computations:

for i = 1, 2, ... Do

compute $y[i] = \hat{\boldsymbol{w}}^H[i]\boldsymbol{x}[i]$

compute $\hat{\boldsymbol{w}}[i+1] = \hat{\boldsymbol{w}}[i] - \mu y^*[i]\boldsymbol{P}\boldsymbol{x}[i]$

end

Computational complexity of the MVDR-LMS algorithm

Recursions	Additions	Multiplications
$y[i] = \hat{\boldsymbol{w}}^H[i]\boldsymbol{x}[i]$	N-1	N
$\boldsymbol{P}\boldsymbol{x}[i]$	2N-1	3N+1
$\hat{\boldsymbol{w}}[i+1] = \hat{\boldsymbol{w}}[i] - \mu y^*[i]\boldsymbol{P}\boldsymbol{x}[i]$	N	N+1
Total	4N-2	5N+2

MVDR-RLS algorithm

- Consider the following optimization problem:

$$\hat{\boldsymbol{w}}[i] = \arg\ \min_{\boldsymbol{w}[i]}\ \boldsymbol{w}^H[i] \underbrace{\hat{\boldsymbol{R}}[i]}_{\sum_{l=1}^{i} \alpha^{i-l} \boldsymbol{x}[l]\boldsymbol{x}^H[l]} \boldsymbol{w}[i] + \delta \boldsymbol{w}^H[i]\boldsymbol{w}[i],$$

$$\text{subject to } \boldsymbol{w}^H[i]\hat{\boldsymbol{a}}(\theta_d) = 1.$$

- Using the method of Lagrange multipliers, we have

$$\mathcal{L}(\boldsymbol{w}[i], \lambda) = \boldsymbol{w}^H[i]\left(\sum_{l=1}^{i} \alpha^{i-l}\, \boldsymbol{x}[l]\boldsymbol{x}^H[l] + \delta \boldsymbol{I}\right)\boldsymbol{w}[i]$$
$$+\lambda(\boldsymbol{w}^H[i]\hat{\boldsymbol{a}}(\theta_d) - 1) + \left(\hat{\boldsymbol{a}}^{\boldsymbol{H}}(\theta_d)\boldsymbol{w}[i] - 1\right)\lambda^*,$$

where α is the forgetting factor.

- Computing the gradient terms of the Lagrangian, we obtain

$$\nabla\mathcal{L}(\boldsymbol{w}, \lambda)_{\boldsymbol{w}^*} = \left(\sum_{l=1}^{i} \alpha^{i-l}\, \boldsymbol{x}[l]\boldsymbol{x}^H[l] + \delta \boldsymbol{I}\right)\boldsymbol{w}[i] + \lambda\hat{\boldsymbol{a}}(\theta_d)$$
$$\nabla\mathcal{L}(\boldsymbol{w}, \lambda)_{\lambda} = \boldsymbol{w}^H[i]\hat{\boldsymbol{a}}(\theta_d) - 1$$

MVDR-RLS algorithm

- Equating the gradient terms to zero, we obtain

$$\nabla\mathcal{L}(\boldsymbol{w},\lambda)_{\boldsymbol{w}^*} = \underbrace{\left(\sum_{l=1}^{i}\alpha^{i-l}\,\boldsymbol{x}[l]\boldsymbol{x}^H[l] + \delta\boldsymbol{I}\right)}_{\widehat{\boldsymbol{R}}[i]=\boldsymbol{P}^{-1}[i]}\boldsymbol{w}[i] + \lambda\widehat{\boldsymbol{a}}(\theta_d) = \boldsymbol{0}$$

$$\Rightarrow \boldsymbol{w}[i] = -\lambda\widehat{\boldsymbol{R}[i]}^{-1}\widehat{\boldsymbol{a}}(\theta_d)$$

$$\nabla\mathcal{L}(\boldsymbol{w},\lambda)_{\lambda} = \boldsymbol{w}^H[i]\widehat{\boldsymbol{a}}(\theta_d) - 1 = 0$$

$$\Rightarrow \boldsymbol{w}^H[i]\widehat{\boldsymbol{a}}(\theta_d) = 1$$

- Substituting $\boldsymbol{w}[i]$ into the constraint $\boldsymbol{w}^H[i]\widehat{\boldsymbol{a}}(\theta_d) = 1$, we have

$$\lambda = -\left(\widehat{\boldsymbol{a}}^H(\theta_d)\widehat{\boldsymbol{R}[i]}^{-1}\widehat{\boldsymbol{a}}(\theta_d)\right)^{-1}$$

- Using the Lagrange multiplier λ in the expression for $\boldsymbol{w}[i]$, we get

$$\widehat{\boldsymbol{w}}[i] = \frac{\widehat{\boldsymbol{R}[i]}^{-1}\widehat{\boldsymbol{a}}(\theta_d)}{\widehat{\boldsymbol{a}}^H(\theta_d)\widehat{\boldsymbol{R}[i]}^{-1}\widehat{\boldsymbol{a}}(\theta_d)} = \gamma_d[i]\boldsymbol{P}[i]\widehat{\boldsymbol{a}}(\theta_d)$$

MVDR-RLS algorithm

- A recursive computation of $\widehat{\boldsymbol{R}[i]}^{-1} = \boldsymbol{P}[i]$can be performed as

$$\boldsymbol{R}[i] = \alpha\boldsymbol{R}[i-1] + \boldsymbol{x}[i]\boldsymbol{x}^H[i]$$

- Using the matrix inversion lemma, we have

$$\boldsymbol{k}[i] = \frac{\alpha^{-1}\boldsymbol{P}[i-1]\boldsymbol{x}[i]}{1 + \alpha^{-1}\boldsymbol{x}^H[i]\boldsymbol{P}[i-1]\boldsymbol{x}[i]},$$

$$\boldsymbol{P}[i] = \alpha^{-1}\boldsymbol{P}[i-1] - \alpha^{-1}\boldsymbol{k}[i]\boldsymbol{x}^H[i]\boldsymbol{P}[i-1],$$

where $\boldsymbol{k}[i] \in \mathbb{C}^N$ is the gain vector and the recursion for $\boldsymbol{P}[i]$ is the so-called Riccati equation.

MVDR-RLS algorithm

- Substituting $\boldsymbol{P}[i]$ in $\boldsymbol{w}[i]$, we obtain

$$\hat{\boldsymbol{w}}[i] = \gamma_d[i]\boldsymbol{P}[i]\hat{\boldsymbol{a}}(\theta_d)$$
$$= \gamma_d[i](\alpha^{-1}\boldsymbol{P}[i-1] - \alpha^{-1}\boldsymbol{k}[i]\boldsymbol{x}^H[i]\boldsymbol{P}[i-1])\hat{\boldsymbol{a}}(\theta_d)$$
$$= \gamma_d[i](\alpha^{-1}\boldsymbol{P}[i-1]\hat{\boldsymbol{a}}(\theta_d) - \alpha^{-1}\boldsymbol{k}[i]\boldsymbol{x}^H[i]\boldsymbol{P}[i-1]\hat{\boldsymbol{a}}(\theta_d))$$
$$= \left(\frac{\alpha^{-1}\gamma_d[i]}{\gamma_d[i-1]}\right)(\hat{\boldsymbol{w}}[i-1] - \boldsymbol{k}[i]\boldsymbol{x}^H[i]\hat{\boldsymbol{w}}[i-1])$$
$$= \left(\frac{\alpha^{-1}\gamma_d[i]}{\gamma_d[i-1]}\right)(\boldsymbol{I} - \boldsymbol{k}[i]\boldsymbol{x}^H[i])\hat{\boldsymbol{w}}[i-1],$$

where $\gamma_d[i] = \frac{1}{\hat{\boldsymbol{a}}^H(\theta_d)\widehat{\boldsymbol{R}[i]}^{-1}\hat{\boldsymbol{a}}(\theta_d)}$ is a gain factor that enforces the constraint.

Statistical analysis of the MVDR-RLS algorithm

- Very similar to the standard RLS algorithm.

- The MSE is given by

$$\mathrm{MSE}[i] = \zeta_{\min} + N\left(\frac{1-\alpha}{2}\right),$$

where $\zeta_{\min} = \frac{1}{\boldsymbol{a}^H(\theta_d)\widehat{\boldsymbol{R}}^{-1}[i]\boldsymbol{a}(\theta_d)}$

Summary of the MVDR-RLS algorithm

Initialization:

N - number of sensors

$\hat{\boldsymbol{w}}[0] = \frac{\boldsymbol{a}(\theta_d)}{N}$

α - forgetting factor

$\boldsymbol{P}[0] = \delta \boldsymbol{I}$

Computations:

for i = 1, 2, ... Do

compute $y[i] = \hat{\boldsymbol{w}}^H[i]\boldsymbol{x}[i]$

compute $\boldsymbol{k}[i] = \frac{\alpha^{-1}\boldsymbol{P}[i-1]\boldsymbol{x}[i]}{1+\alpha^{-1}\boldsymbol{x}^H[i]\boldsymbol{P}[i-1]\boldsymbol{x}[i]}$,

compute $\boldsymbol{P}[i] = \alpha^{-1}\boldsymbol{P}[i-1] - \alpha^{-1}\boldsymbol{k}[i]\boldsymbol{x}^H[i]\boldsymbol{P}[i-1]$

compute $\gamma_d[i] = \frac{1}{\hat{\boldsymbol{a}}^H(\theta_d)\widehat{\boldsymbol{R}[i]}^{-1}\hat{\boldsymbol{a}}(\theta_d)}$

compute $\hat{\boldsymbol{w}}[i] = \left(\frac{\alpha^{-1}\gamma_d[i]}{\gamma_d[i-1]}\right)(\boldsymbol{I} - \boldsymbol{k}[i]\boldsymbol{x}^H[i])\hat{\boldsymbol{w}}[i-1]$

end

Computational complexity of the MVDR-RLS algorithm

Recursions	Additions	Multiplications
$y[i] = \hat{\boldsymbol{w}}^H[i]\boldsymbol{x}[i]$	N-1	N
$\boldsymbol{P}[\mathrm{i}]\boldsymbol{x}[i]$	N^2-N	N^2
$\boldsymbol{k}[i] = \dfrac{\boldsymbol{P}[i-1]\boldsymbol{x}[i]}{\alpha + \boldsymbol{x}^H[i]\boldsymbol{P}[i-1]\boldsymbol{x}[i]}$	N	N
$\boldsymbol{P}[i] = \alpha^{-1}\boldsymbol{P}[i-1] - \alpha^{-1}\boldsymbol{k}[i]\boldsymbol{x}^H[i]\boldsymbol{P}[i-1]$	N^2	$2N^2$+N
$\hat{\boldsymbol{w}}[i] = \left(\dfrac{\alpha^{-1}\gamma_d[i]}{\gamma_d[i-1]}\right)(\boldsymbol{I} - \boldsymbol{k}[i]\boldsymbol{x}^H[i])\hat{\boldsymbol{w}}[i-1]$	N	N+1
Total	$2N^2$+4N-2	$3N^2$+4N+1

Example: MVDR beamforming

- N = 10 sensors
- SNR = 5 dB
- D = 3 interfering signals
- No signal mismatch