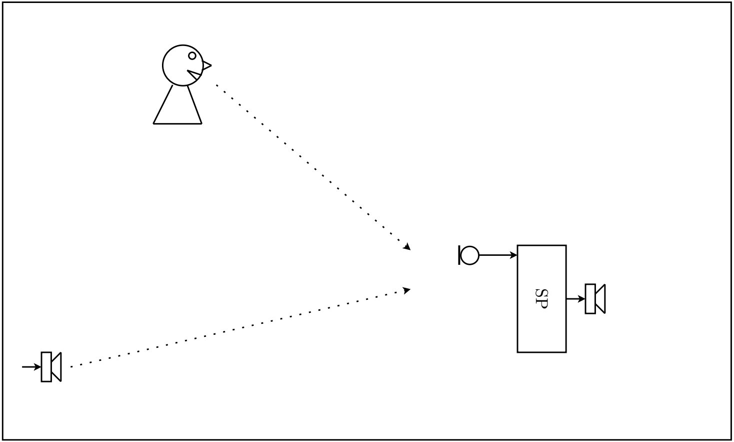
Welcome to Adaptive Array Signal Processing (5SSC0)

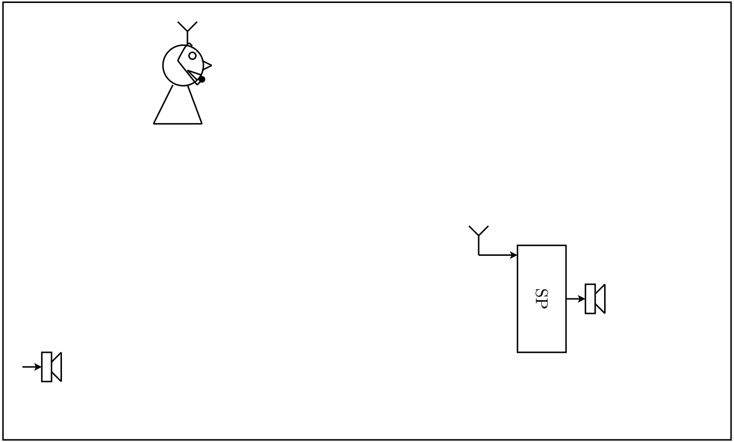
by Ruud van Sloun (lecturer), Iris Huijben (instructor), Julian Merkofer (instructor), and Vincent van de Schaft (instructor).

Motivation AASP example: Skype

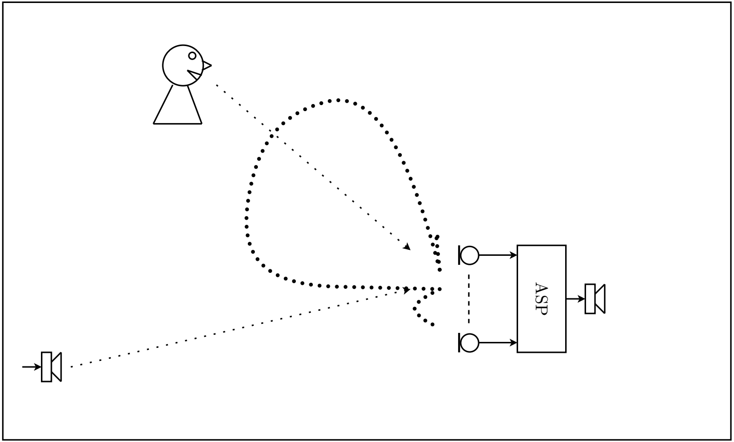
Motivation AASP example: Skype



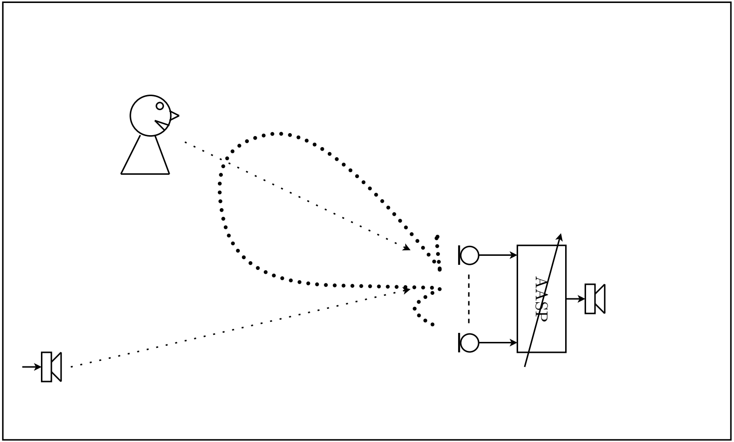
Motivation AASP example: Skype



Motivation AASP example: Skype

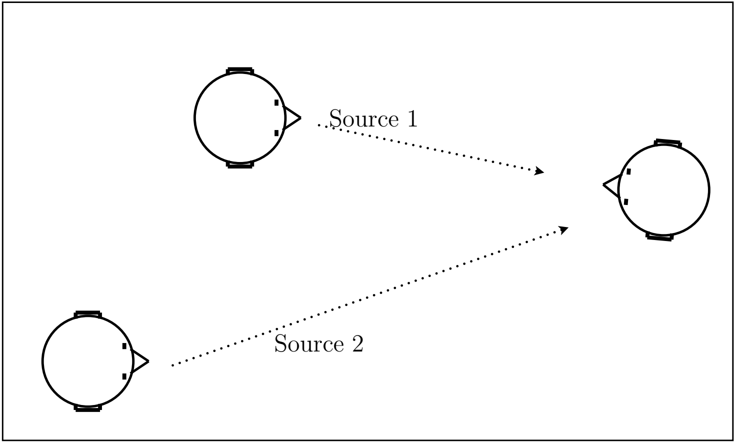


Motivation AASP example: Skype

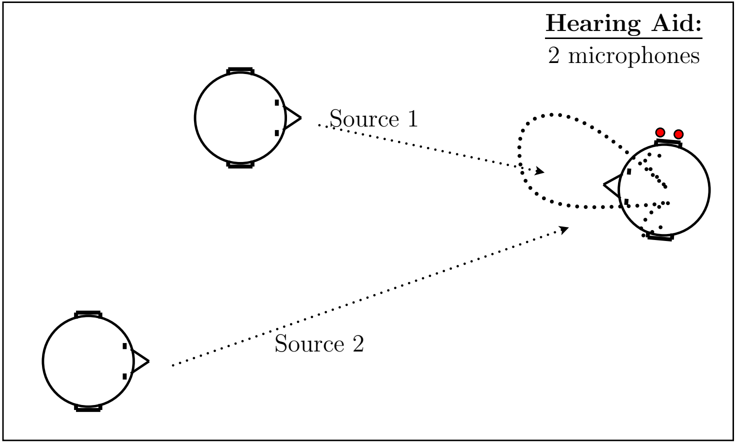


Motivation AASP example: Hearing Aid

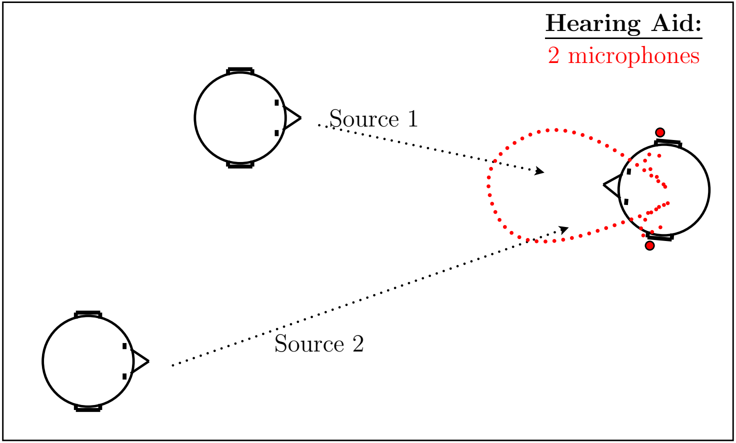
Motivation AASP example: Hearing Aid



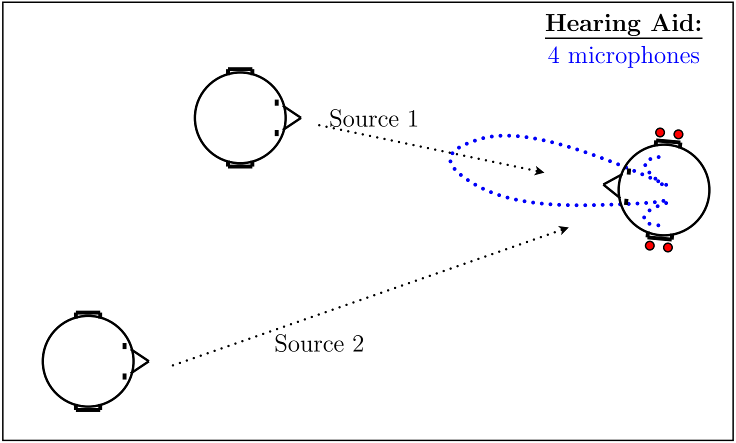
Motivation AASP example: Hearing Aid



Motivation AASP example: Hearing Aid



Motivation AASP example: Hearing Aid



Introduction

- **Main content:**

Introduction

- **Main content:**
  - Part 1A: **Adaptive Signal Processing** (*Single channel, FIR*)
  - Part 1B: **Array Signal Processing (ASP)** (*including DOA*)
  - Part 1C: **Adaptive Array Signal Processing (AASP)**
  - Part 2A: **Compressive sensing: theory**
  - Part 2B: **Compressive sensing: state of the art**

Introduction

- **Main content:**
  - Part 1A: **Adaptive Signal Processing** (*Single channel, FIR*)
  - Part 1B: **Array Signal Processing (ASP)** (*including DOA*)
  - Part 1C: **Adaptive Array Signal Processing (AASP)**
  - Part 2A: **Compressive sensing: theory**
  - Part 2B: **Compressive sensing: state of the art**
- **Assignments and oral exam:**
  1. Groups of 2 students each

Introduction

- **Main content:**
  - Part 1A: **Adaptive Signal Processing** (*Single channel, FIR*)
  - Part 1B: **Array Signal Processing (ASP)** (*including DOA*)
  - Part 1C: **Adaptive Array Signal Processing (AASP)**
  - Part 2A: **Compressive sensing: theory**
  - Part 2B: **Compressive sensing: state of the art**
- **Assignments and oral exam:**
  1. Groups of 2 students each
  2. Assignments 1A, 1B and 1C, 2:
     Fill in predefined documents

Introduction

- **Main content:**
  - Part 1A: **Adaptive Signal Processing** (*Single channel, FIR*)
  - Part 1B: **Array Signal Processing (ASP)** (*including DOA*)
  - Part 1C: **Adaptive Array Signal Processing (AASP)**
  - Part 2A: **Compressive sensing: theory**
  - Part 2B: **Compressive sensing: state of the art**
- **Assignments and oral exam:**
  1. Groups of 2 students each
  2. Assignments 1A, 1B and 1C, 2:
     Fill in predefined documents
  3. All reports to be finalized during course

Introduction

- **Main content:**
  - Part 1A: **Adaptive Signal Processing** (*Single channel, FIR*)
  - Part 1B: **Array Signal Processing (ASP)** (*including DOA*)
  - Part 1C: **Adaptive Array Signal Processing (AASP)**
  - Part 2A: **Compressive sensing: theory**
  - Part 2B: **Compressive sensing: state of the art**
- **Assignments and oral exam:**
  1. Groups of 2 students each
  2. Assignments 1A, 1B and 1C, 2:
     Fill in predefined documents
  3. All reports to be finalized during course
  4. Oral exam: During exam week of Q3

Introduction

- **Main content:**
  - Part 1A: **Adaptive Signal Processing** (*Single channel, FIR*)
  - Part 1B: **Array Signal Processing (ASP)** (*including DOA*)
  - Part 1C: **Adaptive Array Signal Processing (AASP)**
  - Part 2A: **Compressive sensing: theory**
  - Part 2B: **Compressive sensing: state of the art**
- **Assignments and oral exam:**
  1. Groups of 2 students each
  2. Assignments 1A, 1B and 1C, 2:
     Fill in predefined documents
  3. All reports to be finalized during course
  4. Oral exam: During exam week of Q3 **To pass 5SSC0** $\Rightarrow$ **ORAL** ≥ 5

Background material

1. Book:
   Dimitris G. Manolakis, Vinay K. Ingle and Stephen M. Kogon, "Statistical and Adaptive Array Signal Processing: Spectral estimation, signal modeling, adaptive filtering and array processing"
   - Optimum linear filters (Chapter 6)
   - Adaptive filters (Chapter 10)
   - Array processing (Chapter 11)
2. Book:
   Duarte and Eldar, "Structured Compressed Sensing: From Theory to Applications", from Sec. I to II-C (`https://arxiv.org/pdf/1106.6224.pdf`)

Background material

1. Book:
   Dimitris G. Manolakis, Vinay K. Ingle and Stephen M. Kogon, "Statistical and Adaptive Array Signal Processing: Spectral estimation, signal modeling, adaptive filtering and array processing"
   - Optimum linear filters (Chapter 6)
   - Adaptive filters (Chapter 10)
   - Array processing (Chapter 11)
2. Book:
   Duarte and Eldar, "Structured Compressed Sensing: From Theory to Applications", from Sec. I to II-C (`https://arxiv.org/pdf/1106.6224.pdf`)
3. These slides

Background material

1. Book:
   Dimitris G. Manolakis, Vinay K. Ingle and Stephen M. Kogon, "Statistical and Adaptive Array Signal Processing: Spectral estimation, signal modeling, adaptive filtering and array processing"
   - Optimum linear filters (Chapter 6)
   - Adaptive filters (Chapter 10)
   - Array processing (Chapter 11)
2. Book:
   Duarte and Eldar, "Structured Compressed Sensing: From Theory to Applications", from Sec. I to II-C (`https://arxiv.org/pdf/1106.6224.pdf`)
3. These slides
4. Documents: Study guide/ Course organization.

Background material

1. Book:
   Dimitris G. Manolakis, Vinay K. Ingle and Stephen M. Kogon, "Statistical and Adaptive Array Signal Processing: Spectral estimation, signal modeling, adaptive filtering and array processing"
   - Optimum linear filters (Chapter 6)
   - Adaptive filters (Chapter 10)
   - Array processing (Chapter 11)
2. Book:
   Duarte and Eldar, "Structured Compressed Sensing: From Theory to Applications", from Sec. I to II-C (`https://arxiv.org/pdf/1106.6224.pdf`)
3. These slides
4. Documents: Study guide/ Course organization.
5. Necessary Matlab code

Background material

1. Book:
   Dimitris G. Manolakis, Vinay K. Ingle and Stephen M. Kogon, "Statistical and Adaptive Array Signal Processing: Spectral estimation, signal modeling, adaptive filtering and array processing"
   - Optimum linear filters (Chapter 6)
   - Adaptive filters (Chapter 10)
   - Array processing (Chapter 11)
2. Book:
   Duarte and Eldar, "Structured Compressed Sensing: From Theory to Applications", from Sec. I to II-C (`https://arxiv.org/pdf/1106.6224.pdf`)
3. These slides
4. Documents: Study guide/ Course organization.
5. Necessary Matlab code

All relevant material (except book) available via Canvas (Modules):

Background material

1. Book:
   Dimitris G. Manolakis, Vinay K. Ingle and Stephen M. Kogon, "Statistical and Adaptive Array Signal Processing: Spectral estimation, signal modeling, adaptive filtering and array processing"
   - Optimum linear filters (Chapter 6)
   - Adaptive filters (Chapter 10)
   - Array processing (Chapter 11)
2. Book:
   Duarte and Eldar, "Structured Compressed Sensing: From Theory to Applications", from Sec. I to II-C (`https://arxiv.org/pdf/1106.6224.pdf`)
3. These slides
4. Documents: Study guide/ Course organization.
5. Necessary Matlab code

All relevant material (except book) available via Canvas (Modules):

Register for a group in Canvas

Deadlines and credits

Code	Deadline / Date	Credits
1A	February 20, 09:00	10
1B	March 6, 09:00	10
1C	March 20, 09:00	10
2	April 10, 09:00	20
Oral	tbd (April 10-21)	50
Total		100

Deadlines and credits

Code	Deadline / Date	Credits
1A	February 20, 09:00	10
1B	March 6, 09:00	10
1C	March 20, 09:00	10
2	April 10, 09:00	20
Oral	tbd (April 10-21)	50
Total		100

To pass 5SSC0 $\Rightarrow$ **ORAL** ≥ 5

Adaptive Signal Processing

(Part IA)

Content part I

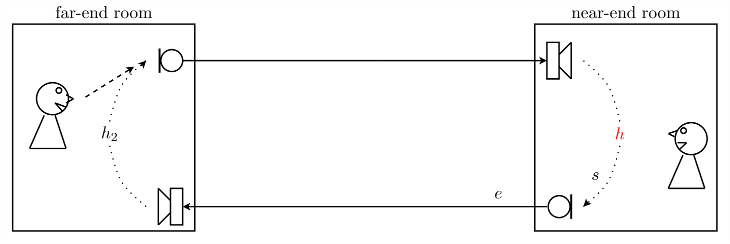
Focus on single channel adaptive algorithms using FIR structures

Content part I

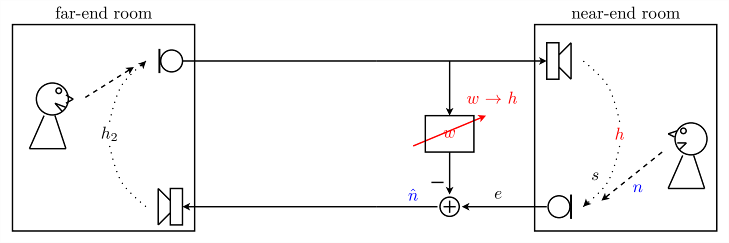
Focus on single channel adaptive algorithms using FIR structures

- Applications
- Minimum Mean Squared Error (MMSE)
- Constrained MMSE
- Least Squares (LS)
- Steepest Descent Algorithm (SGD)
- LMS variants: (Complex) (N)LMS, Constrained LMS
- Newton algorithm
- Recursive Least Squares (RLS)
- Frequency Domain Adaptive Filter (FDAF)
- Summary

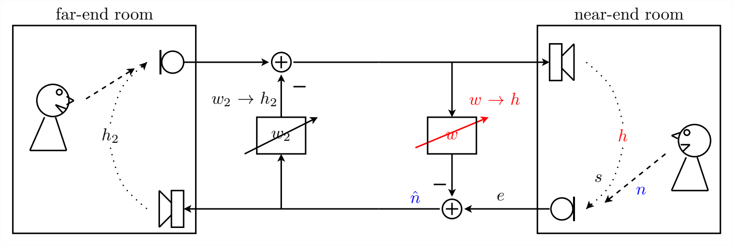
Applications: Acoustic Echo Cancelation



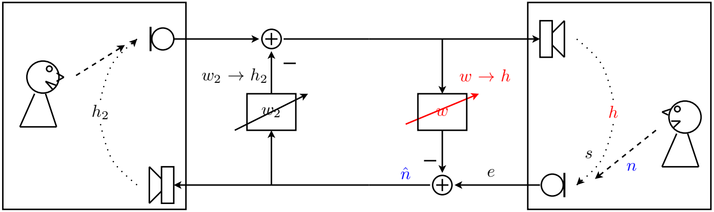
Applications: Acoustic Echo Cancelation

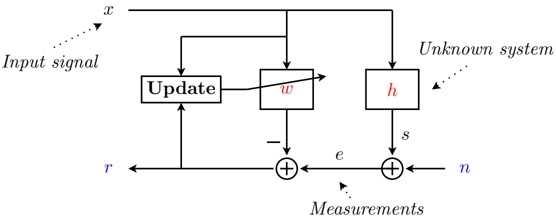


Applications: Acoustic Echo Cancelation



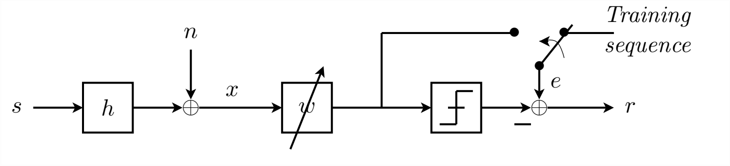
Applications: Acoustic Echo Cancelation



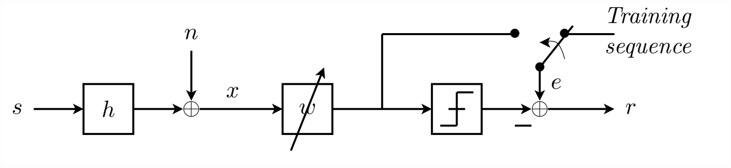


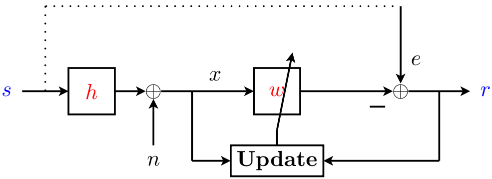
System Identification: $w \to h \Rightarrow r \to n$

Applications: Equalization



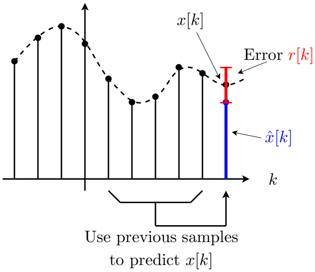
Applications: Equalization



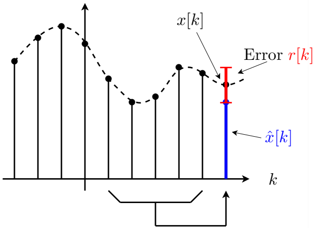


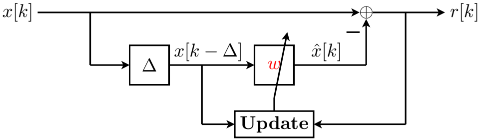
Signal correction/ Inverse modelling: $w \rightarrow h^{-1} \Rightarrow r \rightarrow s$

Applications: Signal prediction



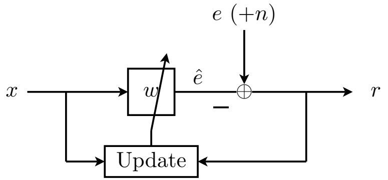
Applications: Signal prediction



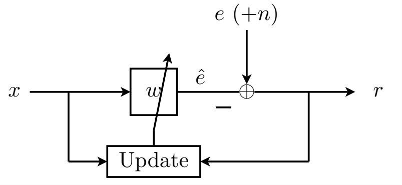


Signal prediction: Predict $x[k]$ from $x[k-\Delta]$

General Adaptive Filter model



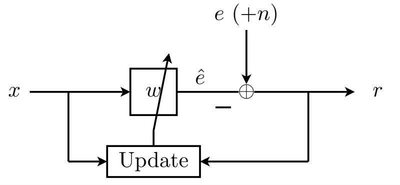
General Adaptive Filter model



Notes:

- Signals x and e correlated

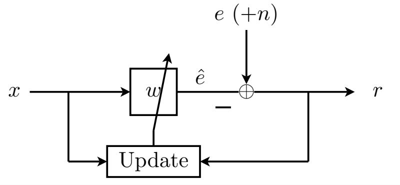
General Adaptive Filter model



Notes:

- Signals x and e correlated
- "Noise" n not correlated with other signals

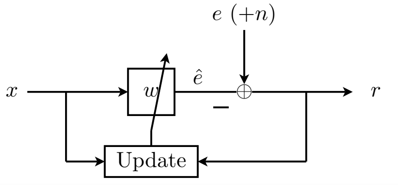
General Adaptive Filter model



Notes:

- Signals x and e correlated
- "Noise" n not correlated with other signals
- Pragmatic choices:
  - All signals in average zero
  - Filter w: FIR

General Adaptive Filter model

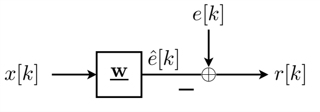


Notes:

- Signals x and e correlated
- "Noise" n not correlated with other signals
- Pragmatic choices:
  - All signals in average zero
  - Filter w: FIR
- Calculation of weight of filter w:
  - Use quadratic cost function: $J = f(r^2)$
  - First fixed weights (MMSE, LS), then adaptive

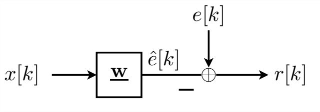
Fixed weights: MMSE

General Minimum Mean Squared Error (MMSE) model:



Fixed weights: MMSE

General Minimum Mean Squared Error (MMSE) model:

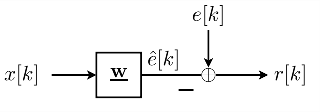


Goal:

Given N samples $\underline{x}[k] = (x[k], x[k-1], \cdots, x[k-N+1])^t$ calculate coefficients fixed filter $\underline{w} = (w_0, w_1, \cdots, w_{N-1})^t$ such that Mean Squared Error (MSE) $J = E\left\{r^2[k]\right\} = E\{(e[k] - \hat{e}[k])^2\}$ is minimized.

Fixed weights: MMSE

General Minimum Mean Squared Error (MMSE) model:



Goal:

Given N samples $\underline{x}[k] = (x[k], x[k-1], \cdots, x[k-N+1])^t$ calculate coefficients <u>fixed</u> filter $\underline{w} = (w_0, w_1, \cdots, w_{N-1})^t$ such that Mean Squared Error (MSE) $J = E\left\{r^2[k]\right\} = E\{(e[k] - \hat{e}[k])^2\}$ is minimized.

MMSE Optimization problem:

$$\boxed{\begin{array}{c}\text{Given FIR samples } x[k-i] \text{ for } i = 0, 1, \cdots N-1 \\ \underline{w}_o = \arg\min_{\underline{w}} \left(E\left\{r^2[k]\right\}\right)\end{array}}$$

Fixed weights: MMSE

$$
\begin{aligned}
J &= E\{(e[k] - \underline{w}^t \cdot \underline{x}[k]) \cdot (e[k] - \underline{x}^t[k] \cdot \underline{w})\} \\
&= E\{e^2[k]\} - \underline{w}^t E\{\underline{x}[k]e[k]\} - E\{e[k]\underline{x}^t[k]\}\underline{w} + \underline{w}^t E\{\underline{x}[k]\underline{x}^t[k]\}\underline{w}
\end{aligned}
$$

Fixed weights: MMSE

$$\begin{aligned} J &= E\{(e[k] - \underline{w}^t \cdot \underline{x}[k]) \cdot (e[k] - \underline{x}^t[k] \cdot \underline{w})\} \\ &= E\{e^2[k]\} - \underline{w}^t E\{\underline{x}[k] e[k]\} - E\{e[k] \underline{x}^t[k]\} \underline{w} + \underline{w}^t E\{\underline{x}[k] \underline{x}^t[k]\} \underline{w} \end{aligned}$$

$$\Rightarrow \boxed{J = E\{e^2[k]\} - \underline{w}^t \underline{r}_{ex} - \underline{r}_{ex}^t \underline{w} + \underline{w}^t \mathrm{R}_x \underline{w}}$$

Fixed weights: MMSE

$$\begin{aligned} J &= E\{(e[k] - \underline{w}^t \cdot \underline{x}[k]) \cdot (e[k] - \underline{x}^t[k] \cdot \underline{w})\} \\ &= E\{e^2[k]\} - \underline{w}^t E\{\underline{x}[k] e[k]\} - E\{e[k]\underline{x}^t[k]\}\underline{w} + \underline{w}^t E\{\underline{x}[k]\underline{x}^t[k]\}\underline{w} \end{aligned}$$

$$\Rightarrow \boxed{J = E\{e^2[k]\} - \underline{w}^t \underline{r}_{ex} - \underline{r}_{ex}^t \underline{w} + \underline{w}^t \mathrm{R}_x \underline{w}}$$

with cross correlation $\rho_{ex}[\tau] = E\{e[k]x[k-\tau]\}$:

$$\underline{r}_{ex} = E\{e[k]\underline{x}[k]\} = (\rho_{ex}[0], \rho_{ex}[1], \cdots, \rho_{ex}[N-1])^t$$

Fixed weights: MMSE

$$
\begin{aligned}
J &= E\{(e[k] - \underline{w}^t \cdot \underline{x}[k]) \cdot (e[k] - \underline{x}^t[k] \cdot \underline{w})\} \\
&= E\{e^2[k]\} - \underline{w}^t E\{\underline{x}[k]e[k]\} - E\{e[k]\underline{x}^t[k]\}\underline{w} + \underline{w}^t E\{\underline{x}[k]\underline{x}^t[k]\}\underline{w}
\end{aligned}
$$

$$
\Rightarrow \boxed{J = E\{e^2[k]\} - \underline{w}^t \underline{r}_{ex} - \underline{r}_{ex}^t \underline{w} + \underline{w}^t \mathrm{R}_x \underline{w}}
$$

with cross correlation $\rho_{ex}[\tau] = E\{e[k]x[k-\tau]$:

$$
\underline{r}_{ex} = E\{e[k]\underline{x}[k]\} = (\rho_{ex}[0], \rho_{ex}[1], \cdots, \rho_{ex}[N-1])^t
$$

and autocorrelation: $\rho_x[\tau] = E\{x[k]x[k-\tau]\} = rho_x[-\tau]$

$$
\mathrm{R}_x = E\{\underline{x}[k]\underline{x}^t[k]\} = \begin{pmatrix} \rho_x[0] & \rho_x[1] & \cdots & \rho_x[N-1] \\ \rho_x[1] & \rho_x[0] & \cdots & \rho_x[N-2] \\ \vdots & \vdots & \vdots & \vdots \\ \rho_x[N-1] & \rho_x[N-2] & \cdots & \rho_x[0] \end{pmatrix}
$$

Fixed weights: MMSE

$$J = E\{e^2[k]\} - \underline{\mathrm{w}}^t \underline{\mathrm{r}}_{ex} - \underline{\mathrm{r}}_{ex}^t \underline{\mathrm{w}} + \underline{\mathrm{w}}^t \mathrm{R}_x \underline{\mathrm{w}}$$

Fixed weights: MMSE

$$J = E\{e^2[k]\} - \underline{w}^t \underline{r}_{ex} - \underline{r}_{ex}^t \underline{w} + \underline{w}^t \mathrm{R}_x \underline{w}$$

$$\Rightarrow \text{ Optimum: } \underline{\nabla} = \frac{\mathrm{d}J}{\mathrm{d}\underline{w}} = -2(\underline{r}_{ex} - \mathrm{R}_x \underline{w}) = \underline{0}$$

Fixed weights: MMSE

$$J = E\{e^2[k]\} - \underline{w}^t \underline{r}_{ex} - \underline{r}_{ex}^t \underline{w} + \underline{w}^t \mathrm{R}_x \underline{w}$$

$$\Rightarrow \text{ Optimum: } \underline{\nabla} = \frac{\mathrm{d}J}{\mathrm{d}\underline{w}} = -2(\underline{r}_{ex} - \mathrm{R}_x \underline{w}) = \underline{0}$$

$\Rightarrow$ **Normal Equations** $\boxed{\mathrm{R}_x \cdot \underline{w} = \underline{r}_{ex}}$

Fixed weights: MMSE

$$J = E\{e^2[k]\} - \underline{w}^t \underline{r}_{ex} - \underline{r}_{ex}^t \underline{w} + \underline{w}^t \mathrm{R}_x \underline{w}$$

$$\Rightarrow \text{ Optimum: } \underline{\nabla} = \frac{\mathrm{d}J}{\mathrm{d}\underline{w}} = -2(\underline{r}_{ex} - \mathrm{R}_x \underline{w}) = \underline{0}$$

$\Rightarrow$ **Normal Equations** $\boxed{\mathrm{R}_x \cdot \underline{w} = \underline{r}_{ex}}$

$\Rightarrow$ **Wiener filter** $\boxed{\underline{w}_o = \mathrm{R}_x^{-1} \cdot \underline{r}_{ex}}$

Fixed weights: MMSE

$$J = E\{e^2[k]\} - \underline{w}^t \underline{r}_{ex} - \underline{r}_{ex}^t \underline{w} + \underline{w}^t \mathrm{R}_x \underline{w}$$

$$\Rightarrow \text{ Optimum: } \underline{\nabla} = \frac{\mathrm{d}J}{\mathrm{d}\underline{w}} = -2(\underline{r}_{ex} - \mathrm{R}_x \underline{w}) = \underline{0}$$

$\Rightarrow$ **Normal Equations** $\boxed{\mathrm{R}_x \cdot \underline{w} = \underline{r}_{ex}}$

$\Rightarrow$ **Wiener filter** $\boxed{\underline{w}_o = \mathrm{R}_x^{-1} \cdot \underline{r}_{ex}}$

General expression: $\boxed{J = J_{min} + (\underline{w} - \underline{w}_o)^t \cdot \mathrm{R}_x \cdot (\underline{w} - \underline{w}_o)}$

Fixed weights: MMSE

$$J = E\{e^2[k]\} - \underline{w}^t \underline{r}_{ex} - \underline{r}_{ex}^t \underline{w} + \underline{w}^t \mathrm{R}_x \underline{w}$$

$$\Rightarrow \text{ Optimum: } \underline{\bigtriangledown} = \frac{\mathrm{d}J}{\mathrm{d}\underline{w}} = -2(\underline{r}_{ex} - \mathrm{R}_x \underline{w}) = \underline{0}$$

$\Rightarrow$ **Normal Equations** $\boxed{\mathrm{R}_x \cdot \underline{w} = \underline{r}_{ex}}$

$\Rightarrow$ **Wiener filter** $\boxed{\underline{w}_o = \mathrm{R}_x^{-1} \cdot \underline{r}_{ex}}$

General expression: $\boxed{J = J_{min} + (\underline{w} - \underline{w}_o)^t \cdot \mathrm{R}_x \cdot (\underline{w} - \underline{w}_o)}$

$$J_{min} = J_{\underline{w}=\underline{w}_o} = E\{r^2[k]\} = E\{r[k] \cdot e[k]\} = E\{e^2[k]\} - \underline{r}_{ex}^t \mathrm{R}_x^{-1} \underline{r}_{ex}$$

Fixed weights: MMSE

$$J = E\{e^2[k]\} - \underline{w}^t \underline{r}_{ex} - \underline{r}_{ex}^t \underline{w} + \underline{w}^t R_x \underline{w}$$

$$\Rightarrow \text{ Optimum: } \underline{\nabla} = \frac{dJ}{d\underline{w}} = -2(\underline{r}_{ex} - R_x \underline{w}) = \underline{0}$$

$\Rightarrow$ **Normal Equations** $\quad \boxed{R_x \cdot \underline{w} = \underline{r}_{ex}}$

$\Rightarrow$ **Wiener filter** $\quad \boxed{\underline{w}_o = R_x^{-1} \cdot \underline{r}_{ex}}$

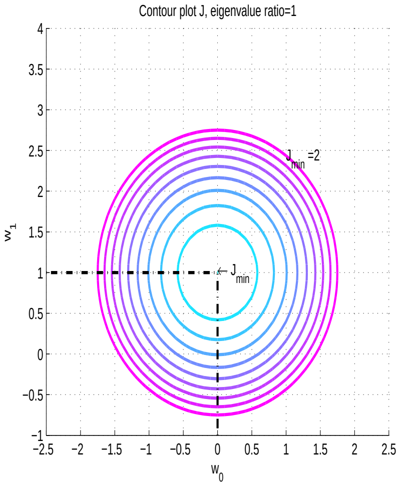
General expression: $\boxed{J = J_{min} + (\underline{w} - \underline{w}_o)^t \cdot R_x \cdot (\underline{w} - \underline{w}_o)}$

$$J_{min} = J_{\underline{w}=\underline{w}_o} = E\{r^2[k]\} = E\{r[k] \cdot e[k]\} = E\{e^2[k]\} - \underline{r}_{ex}^t R_x^{-1} \underline{r}_{ex}$$

From general expression $\Rightarrow$ J quadratic in $\underline{w}$ thus $\underline{w}_o$ really minimum

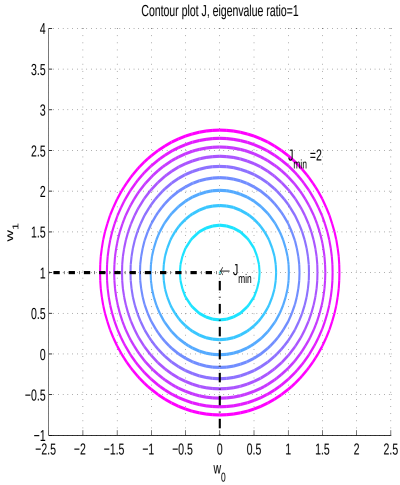
Fixed weights: MMSE

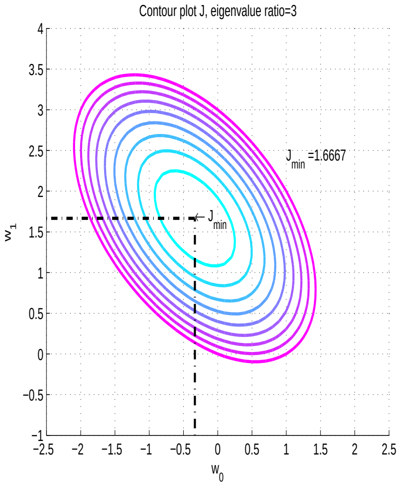
Contour plots $J = J_{min} + (\underline{w} - \underline{w}_o)^t \cdot R_x \cdot (\underline{w} - \underline{w}_o)$



Fixed weights: MMSE

Contour plots $J = J_{min} + (\underline{w} - \underline{w}_o)^t \cdot R_x \cdot (\underline{w} - \underline{w}_o)$





Eigenvalues: see Appendix

Two MMSE variants

Two MMSE variants

- **Complex MMSE:**

Setup with complex signals and weights

Similar result as before:

$$\boxed{\underline{\mathrm{w}}_o = \mathrm{R}_x^{-1} \cdot \underline{\mathrm{r}}_{e^* x}}$$

with $\underline{\mathrm{r}}_{e^* x} = E\{e^*[k]\underline{\mathrm{x}}[k]\}$ and $\mathrm{R}_x = E\{\underline{\mathrm{x}}[k] \cdot \underline{\mathrm{x}}^h[k]\}$ (h=hermetian)

Two MMSE variants

- **Complex MMSE:**

Setup with complex signals and weights

Similar result as before:

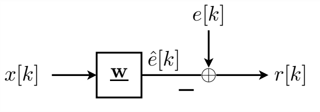
$$\boxed{\underline{\mathrm{w}}_o = \mathrm{R}_x^{-1} \cdot \underline{\mathrm{r}}_{e^*x}}$$

with $\underline{\mathrm{r}}_{e^*x} = E\{e^*[k]\underline{\mathrm{x}}[k]\}$ and $\mathrm{R}_x = E\{\underline{\mathrm{x}}[k] \cdot \underline{\mathrm{x}}^h[k]\}$ (h=hermetian)

- **Constrained MMSE:**

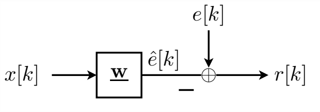
Setup with set of constraints on weights

Constrained MMSE



Goal: Minimize $E\{r^2[k]\}$ subject to M constraints

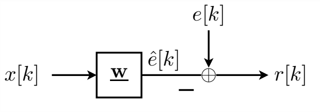
Constrained MMSE



Goal: Minimize $E\{r^2[k]\}$ subject to M constraints

Example: $$\sum_{i=0}^{N-1} w_i = 1$$

Constrained MMSE

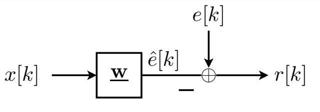


Goal: Minimize $E\{r^2[k]\}$ subject to M constraints

Example:

$$\sum_{i=0}^{N-1} w_i = 1 \quad \Leftrightarrow \quad (1, 1, \cdots, 1)^t \cdot \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_{N-1} \end{pmatrix} = 1$$

Constrained MMSE



Goal: Minimize $E\{r^2[k]\}$ subject to M constraints

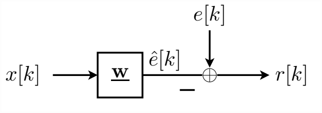
<u>Example:</u>

$$\sum_{i=0}^{N-1} w_i = 1 \quad \Leftrightarrow \quad (1,1,\cdots,1)^t \cdot \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_{N-1} \end{pmatrix} = 1$$

<u>General:</u>

$$\begin{aligned} \underline{c}_1^t \cdot \underline{w} &= f_1 \\ \underline{c}_2^t \cdot \underline{w} &= f_2 \\ &\vdots \\ \underline{c}_M^t \cdot \underline{w} &= f_M \end{aligned}$$

Constrained MMSE



Goal: Minimize $E\{r^2[k]\}$ subject to M constraints

Example:

$$\sum_{i=0}^{N-1} w_i = 1 \quad \Leftrightarrow \quad (1, 1, \cdots, 1)^t \cdot \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_{N-1} \end{pmatrix} = 1$$

General:

$$\begin{matrix} \underline{c}_1^t \cdot \underline{w} = f_1 \\ \underline{c}_2^t \cdot \underline{w} = f_2 \\ \vdots \\ \underline{c}_M^t \cdot \underline{w} = f_M \end{matrix} \quad \Leftrightarrow \quad \begin{pmatrix} \underline{c}_1^t \\ \underline{c}_2^t \\ \vdots \\ \underline{c}_M^t \end{pmatrix} \cdot \underline{w} = \underline{f} \quad \Leftrightarrow \quad C^t \cdot \underline{w} = \underline{f}$$

Constrained MMSE

Some notes on solving $\mathsf{C}^t \cdot \underline{\mathsf{f}} = \underline{\mathsf{w}}$

Constrained MMSE

Some notes on solving $\quad \mathsf{C}^t \cdot \underline{\mathsf{f}} = \underline{\mathsf{w}}$

$$\begin{aligned} M \times 1 \text{ constraint vector} \quad &: \quad \underline{\mathsf{f}} = (f_1, f_2, \cdots, f_M)^t \\ N \times M \text{ constraint matrix} \quad &: \quad \mathsf{C} = (\underline{\mathsf{c}}_1, \underline{\mathsf{c}}_2, \cdots, \underline{\mathsf{c}}_M)^t \end{aligned}$$

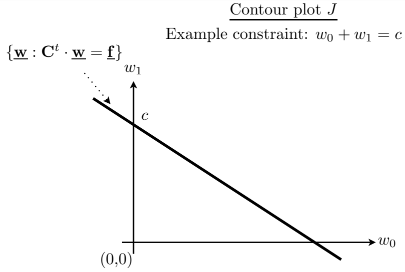
Note: $\quad M$ independent constraints $\Rightarrow$ C full rank

Constrained MMSE

Some notes on solving $\quad \mathsf{C}^t \cdot \underline{\mathsf{f}} = \underline{\mathsf{w}}$

$$\begin{array}{rcl} M \times 1 \text{ constraint vector} & : & \underline{\mathsf{f}} = (f_1, f_2, \cdots, f_M)^t \\ N \times M \text{ constraint matrix} & : & \mathsf{C} = (\underline{\mathsf{c}}_1, \underline{\mathsf{c}}_2, \cdots, \underline{\mathsf{c}}_M)^t \end{array}$$

Note: $\quad M$ independent constraints $\Rightarrow$ C full rank



Constrained MMSE

Solutions of $\mathrm{C}^t \cdot \underline{\mathrm{w}} = \underline{\mathrm{f}}$

Constrained MMSE

Solutions of $\mathrm{C}^t \cdot \underline{\mathrm{w}} = \underline{\mathrm{f}}$

- <u>Case $N = M$:</u>

  $\Rightarrow$ $\underline{\mathrm{w}}^c = (\mathrm{C}^t)^{-1} \cdot \underline{\mathrm{f}}$

  $\Rightarrow$ *No degrees of freedom left for MMSE*

Constrained MMSE

Solutions of $\mathrm{C}^t \cdot \underline{\mathrm{w}} = \underline{\mathrm{f}}$

- Case $N = M$:

  $\Rightarrow$ $\underline{\mathrm{w}}^c = (\mathrm{C}^t)^{-1} \cdot \underline{\mathrm{f}}$

  $\Rightarrow$ *No degrees of freedom left for MMSE*

- Case $N > M$:

  $\Rightarrow$ Possible solution $\underline{\mathrm{w}}^c = (\mathrm{C}^t)^\dagger \cdot \underline{\mathrm{f}}$
  Appendix: Generalized inverse $(\mathrm{C}^t)^\dagger = \mathrm{C} \cdot (\mathrm{C}^t \cdot \mathrm{C})^{-1}$

  $\Rightarrow$ $N - M$ *degrees of freedom left over for MMSE*

Constrained MMSE

Solutions of $\mathrm{C}^t \cdot \underline{\mathrm{w}} = \underline{\mathrm{f}}$

- Case $N = M$:

  $\Rightarrow$ $\underline{\mathrm{w}}^c = (\mathrm{C}^t)^{-1} \cdot \underline{\mathrm{f}}$

  $\Rightarrow$ *No degrees of freedom left for MMSE*

- Case $N > M$:

  $\Rightarrow$ Possible solution $\underline{\mathrm{w}}^c = (\mathrm{C}^t)^\dagger \cdot \underline{\mathrm{f}}$
  Appendix: Generalized inverse $(\mathrm{C}^t)^\dagger = \mathrm{C} \cdot (\mathrm{C}^t \cdot \mathrm{C})^{-1}$

  $\Rightarrow$ *$N - M$ degrees of freedom left over for MMSE*

- Case $N < M$:

  $\Rightarrow$ *Conflicting solutions*

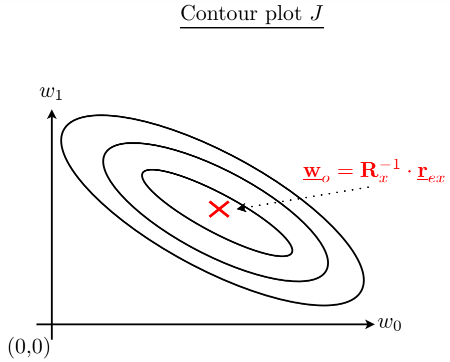
  $\Rightarrow$ Choose e.g. minimum norm solution

Constrained MMSE

We can't reach $\underline{w}_o$, but we can do better than $\underline{w}^c$:

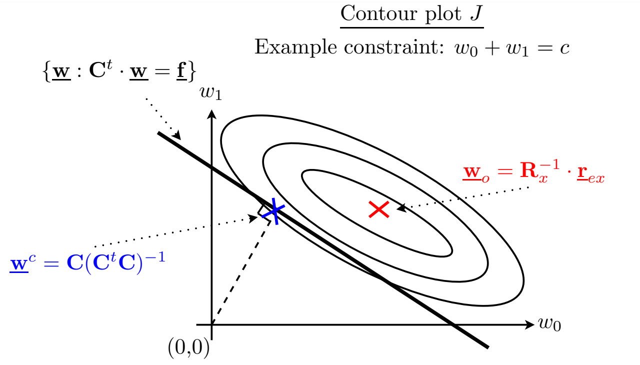
Constrained MMSE

We can't reach $\underline{w}_o$, but we can do better than $\underline{w}^c$:



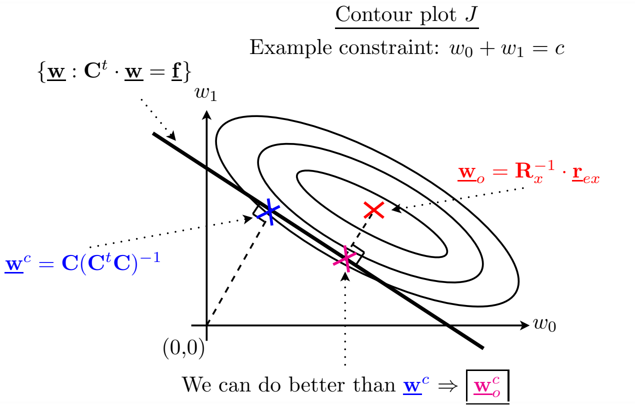
Constrained MMSE

We can't reach $\underline{w}_o$, but we can do better than $\underline{w}^c$:



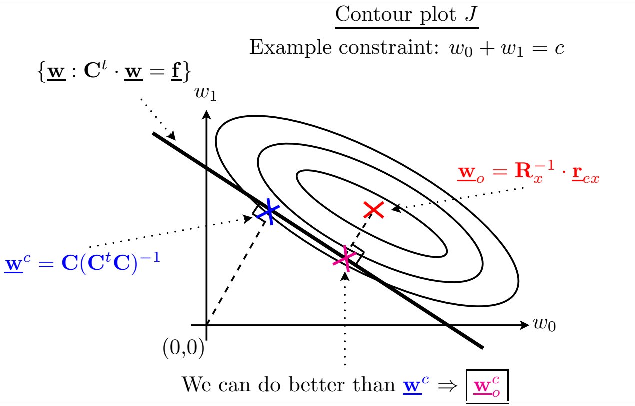
Constrained MMSE

We can't reach $\underline{w}_o$, but we can do better than $\underline{w}^c$:



Constrained MMSE

We can't reach $\underline{w}_o$, but we can do better than $\underline{w}^c$:



Use $N - M$ degrees of freedom to improve result: $\underline{w}^c \Rightarrow \underline{w}_o^c$

Constrained MMSE

Use Lagrange multipliers

Constrained MMSE

Use Lagrange multipliers

Performance index:

$$\begin{aligned} J^c &= E\{r^2\} + \underline{\lambda}^t(\mathrm{C}^t\underline{\mathrm{w}} - \underline{\mathrm{f}}) \\ &= E\{e^2\} - \underline{\mathrm{w}}^t\underline{\mathrm{r}}_{ex} - \underline{\mathrm{r}}_{ex}^t\underline{\mathrm{w}} + \underline{\mathrm{w}}^t\mathrm{R}_x\underline{\mathrm{w}} + \underline{\lambda}^t(\mathrm{C}^t\underline{\mathrm{w}} - \underline{\mathrm{f}}) \end{aligned}$$

Constrained MMSE

Use Lagrange multipliers

Performance index:

$$
\begin{aligned}
J^c &= E\{r^2\} + \underline{\lambda}^t(\mathrm{C}^t\underline{\mathrm{w}} - \underline{\mathrm{f}}) \\
&= E\{e^2\} - \underline{\mathrm{w}}^t\underline{\mathrm{r}}_{ex} - \underline{\mathrm{r}}_{ex}^t\underline{\mathrm{w}} + \underline{\mathrm{w}}^t\mathrm{R}_x\underline{\mathrm{w}} + \underline{\lambda}^t(\mathrm{C}^t\underline{\mathrm{w}} - \underline{\mathrm{f}})
\end{aligned}
$$

Gradient vector:

$$\frac{\mathrm{d}J^c}{\mathrm{d}\underline{\mathrm{w}}} = -2\underline{\mathrm{r}}_{ex} + 2\mathrm{R}_x\underline{\mathrm{w}} + \mathrm{C}\underline{\lambda}$$

$$\frac{\mathrm{d}J^c}{\mathrm{d}\underline{\mathrm{w}}} = \underline{0} \;\Rightarrow\; \underline{\mathrm{w}}_o^c = \mathrm{R}_x^{-1}\underline{\mathrm{r}}_{ex} - \frac{1}{2}\mathrm{R}_x^{-1}\mathrm{C}\underline{\lambda}$$

Constrained MMSE

Use Lagrange multipliers

Performance index:

$$\begin{aligned} J^c &= E\{r^2\} + \underline{\lambda}^t(C^t\underline{w} - \underline{f}) \\ &= E\{e^2\} - \underline{w}^t\underline{r}_{ex} - \underline{r}_{ex}^t\underline{w} + \underline{w}^t R_x \underline{w} + \underline{\lambda}^t(C^t\underline{w} - \underline{f}) \end{aligned}$$

Gradient vector:

$$\frac{dJ^c}{d\underline{w}} = -2\underline{r}_{ex} + 2R_x\underline{w} + C\underline{\lambda}$$

$$\frac{dJ^c}{d\underline{w}} = \underline{0} \;\Rightarrow\; \underline{w}_o^c = R_x^{-1}\underline{r}_{ex} - \frac{1}{2}R_x^{-1}C\underline{\lambda}$$

Furthermore in optimum: $C^t\underline{w}_o^c = \underline{f}$

Constrained MMSE

Use Lagrange multipliers

Performance index:

$$\begin{aligned} J^c &= E\{r^2\} + \underline{\lambda}^t(\mathrm{C}^t\underline{\mathrm{w}} - \underline{\mathrm{f}}) \\ &= E\{e^2\} - \underline{\mathrm{w}}^t\underline{\mathrm{r}}_{ex} - \underline{\mathrm{r}}_{ex}^t\underline{\mathrm{w}} + \underline{\mathrm{w}}^t\mathrm{R}_x\underline{\mathrm{w}} + \underline{\lambda}^t(\mathrm{C}^t\underline{\mathrm{w}} - \underline{\mathrm{f}}) \end{aligned}$$

Gradient vector:

$$\frac{\mathrm{d}J^c}{\mathrm{d}\underline{\mathrm{w}}} = -2\underline{\mathrm{r}}_{ex} + 2\mathrm{R}_x\underline{\mathrm{w}} + \mathrm{C}\underline{\lambda}$$

$$\frac{\mathrm{d}J^c}{\mathrm{d}\underline{\mathrm{w}}} = \underline{0} \;\Rightarrow\; \underline{\mathrm{w}}_o^c = \mathrm{R}_x^{-1}\underline{\mathrm{r}}_{ex} - \frac{1}{2}\mathrm{R}_x^{-1}\mathrm{C}\underline{\lambda}$$

Furthermore in optimum: $\mathrm{C}^t\underline{\mathrm{w}}_o^c = \underline{\mathrm{f}}$

Combine last two equations:

$$\Rightarrow \quad \underline{\lambda} = 2(\mathrm{C}^t\mathrm{R}_x^{-1}\mathrm{C})^{-1}(\mathrm{C}^t\mathrm{R}_x^{-1}\underline{\mathrm{r}}_{ex} - \underline{\mathrm{f}})$$

Constrained MMSE

$$\Rightarrow \quad \boxed{\underline{w}_o^c = \underline{w}_o + R_x^{-1}C(C^tR_x^{-1}C)^{-1}(\underline{f} - C^t\underline{w}_o)}$$

with

$$\underline{w}_o = R_x^{-1}\underline{r}_{ex}$$

Constrained MMSE

$$\Rightarrow \quad \boxed{\underline{w}_o^c = \underline{w}_o + R_x^{-1} C (C^t R_x^{-1} C)^{-1} (\underline{f} - C^t \underline{w}_o)}$$

with

$$\underline{w}_o = R_x^{-1} \underline{r}_{ex}$$

<u>Similar result:</u>

$$\boxed{\underline{w}_o^c = R_x^{-1} C (C^t R_x^{-1} C)^{-1} \underline{f}}$$

Constrained MMSE

$$\Rightarrow \quad \boxed{\underline{w}_o^c = \underline{w}_o + R_x^{-1}C(C^t R_x^{-1} C)^{-1}(\underline{f} - C^t \underline{w}_o)}$$

with

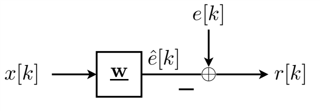
$$\underline{w}_o = R_x^{-1}\underline{r}_{ex}$$

<u>Similar result:</u>

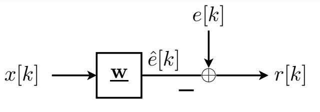
$$\boxed{\underline{w}_o^c = R_x^{-1}C(C^t R_x^{-1} C)^{-1}\underline{f}}$$

Check: $$C^t \underline{w}_o^c = C^t \underline{w}_o + (C^t R_x^{-1} C)(C^t R_x^{-1} C)^{-1}(\underline{f} - C^t \underline{w}_o) = \underline{f}$$

Least Squares (LS)



Least Squares (LS)



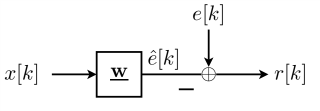
Different quadratic cost functions:

- Mean Square Error (MSE):

$$J_{mse} = E\{r^2[k]\} = E\{(e[k] - \underline{w}^t \underline{x}[k])^2\}$$

$\Rightarrow$ Minimum MSE (MMSE) = Wiener

Least Squares (LS)



<u>Different quadratic cost functions:</u>

- Mean Square Error (MSE):

$$J_{mse} = E\{r^2[k]\} = E\{(e[k] - \underline{w}^t \underline{x}[k])^2\}$$

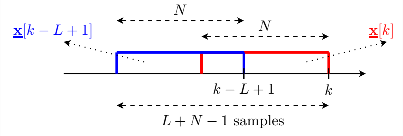
$\Rightarrow$ Minimum MSE (MMSE) = Wiener

- **Least Square (LS):** If statistical information is not available

$\Rightarrow$
Use criterion based on data (thus without $E\{\cdot\}$)

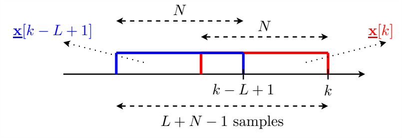
LS

Collect L (≥ 1) data vectors $\underline{x}[k-i]$ (each of length N)



LS

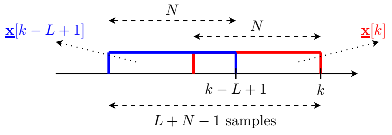
Collect L (≥ 1) data vectors $\underline{\mathrm{x}}[k-i]$ (each of length N)



<u>Available data (for $i = 0, 1, \cdots, L-1$):</u>

LS

Collect L (≥ 1) data vectors $\underline{x}[k-i]$ (each of length N)



Available data (for $i = 0, 1, \cdots, L-1$):

- *Input signal samples/ vectors* $\underline{x}[k-i]$

$$\underline{x}^t[k-i] = (x[k-i], x[k-i-1], \cdots, x[k-i-N+1])^t$$

- *Reference signal samples:* $e[k-i]$

- *Residual signal samples:* $r[k-i] = e[k-i] - \underline{x}^t[k-i] \cdot \underline{w}$

LS

Notation:

$$\mathrm{X}[k] = \begin{pmatrix} \underline{\mathrm{x}}^t[k] \\ \underline{\mathrm{x}}^t[k-1] \\ \vdots \\ \underline{\mathrm{x}}^t[k-L+1] \end{pmatrix} \qquad \underline{\mathrm{w}} = \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_{N-1} \end{pmatrix}$$

$$\underline{\mathrm{e}}[k] = \begin{pmatrix} e[k] \\ e[k-1] \\ \vdots \\ e[k-L+1] \end{pmatrix} \qquad \underline{\mathrm{r}}[k] = \begin{pmatrix} r[k] \\ r[k-1] \\ \vdots \\ r[k-L+1] \end{pmatrix}$$

LS

Notation:

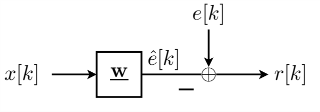
$$X[k] = \begin{pmatrix} \underline{x}^t[k] \\ \underline{x}^t[k-1] \\ \vdots \\ \underline{x}^t[k-L+1] \end{pmatrix} \qquad \underline{w} = \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_{N-1} \end{pmatrix}$$

$$\underline{e}[k] = \begin{pmatrix} e[k] \\ e[k-1] \\ \vdots \\ e[k-L+1] \end{pmatrix} \qquad \underline{r}[k] = \begin{pmatrix} r[k] \\ r[k-1] \\ \vdots \\ r[k-L+1] \end{pmatrix}$$

Simplified notation (skip time indices):

$$\underline{r} = \underline{e} - X \cdot \underline{w}$$

LS



LS problem formulation:

$$\boxed{\underline{w}_{ls,o} = \arg\min_{\underline{w}} |\underline{e} - X \cdot \underline{w}|^2}$$

LS

$$J_{ls} = \sum_{i=0}^{L-1} r^2[k-i] = \underline{\mathrm{r}}^t \cdot \underline{\mathrm{r}} = (\underline{\mathrm{e}}^t - \underline{\mathrm{w}}^t \mathrm{X}^t) \cdot (\underline{\mathrm{e}} - \mathrm{X}\underline{\mathrm{w}})$$

LS

$$
\begin{aligned}
J_{ls} &= \sum_{i=0}^{L-1} r^2[k-i] = \underline{\mathrm{r}}^t \cdot \underline{\mathrm{r}} = (\underline{\mathrm{e}}^t - \underline{\mathrm{w}}^t \mathrm{X}^t) \cdot (\underline{\mathrm{e}} - \mathrm{X}\underline{\mathrm{w}}) \\
&= \underline{\mathrm{e}}^t \underline{\mathrm{e}} + \underline{\mathrm{w}}^t \mathrm{X}^t \mathrm{X} \underline{\mathrm{w}} - \underline{\mathrm{w}}^t \mathrm{X}^t \underline{\mathrm{e}} - \underline{\mathrm{e}}^t \mathrm{X} \underline{\mathrm{w}}
\end{aligned}
$$

LS

$$\begin{aligned} J_{ls} &= \sum_{i=0}^{L-1} r^2[k-i] = \underline{r}^t \cdot \underline{r} = (\underline{e}^t - \underline{w}^t X^t) \cdot (\underline{e} - X\underline{w}) \\ &= \underline{e}^t \underline{e} + \underline{w}^t X^t X \underline{w} - \underline{w}^t X^t \underline{e} - \underline{e}^t X \underline{w} \end{aligned}$$

Minimum by setting gradient equal to zero:

$$\frac{\mathrm{d}J_{ls}}{\mathrm{d}\underline{w}} = \underline{\bigtriangledown}_{ls} = -2(X^t \underline{e} - X^t X \cdot \underline{w}) = \underline{0}$$

LS

$$
\begin{aligned}
J_{ls} &= \sum_{i=0}^{L-1} r^2[k-i] = \underline{\mathrm{r}}^t \cdot \underline{\mathrm{r}} = (\underline{\mathrm{e}}^t - \underline{\mathrm{w}}^t \mathrm{X}^t) \cdot (\underline{\mathrm{e}} - \mathrm{X}\underline{\mathrm{w}}) \\
&= \underline{\mathrm{e}}^t \underline{\mathrm{e}} + \underline{\mathrm{w}}^t \mathrm{X}^t \mathrm{X} \underline{\mathrm{w}} - \underline{\mathrm{w}}^t \mathrm{X}^t \underline{\mathrm{e}} - \underline{\mathrm{e}}^t \mathrm{X} \underline{\mathrm{w}}
\end{aligned}
$$

Minimum by setting gradient equal to zero:

$$
\frac{\mathrm{d}J_{ls}}{\mathrm{d}\underline{\mathrm{w}}} = \underline{\bigtriangledown}_{ls} = -2(\mathrm{X}^t \underline{\mathrm{e}} - \mathrm{X}^t \mathrm{X} \cdot \underline{\mathrm{w}}) = \underline{0}
$$

With $\overline{\mathrm{R}} = \mathrm{X}^t \mathrm{X}$ and $\underline{\bar{\mathrm{r}}} = \mathrm{X}^t \underline{\mathrm{e}}$

LS

$$
\begin{aligned}
J_{ls} &= \sum_{i=0}^{L-1} r^2[k-i] = \underline{r}^t \cdot \underline{r} = (\underline{e}^t - \underline{w}^t X^t) \cdot (\underline{e} - X\underline{w}) \\
&= \underline{e}^t \underline{e} + \underline{w}^t X^t X \underline{w} - \underline{w}^t X^t \underline{e} - \underline{e}^t X \underline{w}
\end{aligned}
$$

Minimum by setting gradient equal to zero:

$$
\frac{\mathrm{d}J_{ls}}{\mathrm{d}\underline{w}} = \underline{\nabla}_{ls} = -2(X^t \underline{e} - X^t X \cdot \underline{w}) = \underline{0}
$$

With $\overline{R} = X^t X$ and $\underline{\bar{r}} = X^t \underline{e}$

$\Rightarrow$ **Normal Equations** $\boxed{\overline{R}_x \cdot \underline{w} = \underline{\bar{r}}_{ex}}$

$\Rightarrow$ **Wiener filter** $\boxed{\underline{w}_{ls,o} = \overline{R}_x^{-1} \cdot \underline{\bar{r}}_{ex}}$

LS: Correspondence with MMSE

Use time-averaging (ergodicity):

$$\begin{aligned}
\hat{\mathrm{R}}_x &= \frac{1}{L}\sum_{i=0}^{L-1} \underline{\mathrm{x}}[k-i]\cdot\underline{\mathrm{x}}^t[k-i] = \frac{1}{L}\mathrm{X}^t\cdot\mathrm{X} = \frac{1}{L}\overline{\mathrm{R}}_x \\
\hat{\underline{\mathrm{r}}}_{ex} &= \frac{1}{L}\sum_{i=0}^{L-1} \underline{\mathrm{x}}[k-i]\cdot e[k-i] = \frac{1}{L}\mathrm{X}^t\cdot\underline{\mathrm{e}} = \frac{1}{L}\overline{\underline{\mathrm{r}}}_{ex}
\end{aligned}$$

LS: Correspondence with MMSE

Use time-averaging (ergodicity):

$$\hat{\mathrm{R}}_x = \frac{1}{L}\sum_{i=0}^{L-1} \underline{\mathrm{x}}[k-i] \cdot \underline{\mathrm{x}}^t[k-i] = \frac{1}{L}\mathrm{X}^t \cdot \mathrm{X} = \frac{1}{L}\overline{\mathrm{R}}_x$$

$$\hat{\underline{\mathrm{r}}}_{ex} = \frac{1}{L}\sum_{i=0}^{L-1} \underline{\mathrm{x}}[k-i] \cdot e[k-i] = \frac{1}{L}\mathrm{X}^t \cdot \underline{\mathrm{e}} = \frac{1}{L}\overline{\underline{\mathrm{r}}}_{ex}$$

with $\hat{\mathrm{R}}_x$ estimate of R_x and $\hat{\underline{\mathrm{r}}}_{ex}$ estimate of $\underline{\mathrm{r}}_{ex}$

$$\Rightarrow \quad \hat{\underline{\mathrm{w}}}_{mmse} = \left(\frac{1}{L}\overline{\mathrm{R}}_x\right)^{-1} \cdot \left(\frac{1}{L}\overline{\underline{\mathrm{r}}}_{ex}\right) = \overline{\mathrm{R}}_x^{-1} \cdot \overline{\underline{\mathrm{r}}}_{ex} = \underline{\mathrm{w}}_{ls}$$

LS: Correspondence with MMSE

Use time-averaging (ergodicity):

$$\begin{aligned}
\hat{\mathrm{R}}_x &= \frac{1}{L}\sum_{i=0}^{L-1} \underline{x}[k-i]\cdot\underline{x}^t[k-i] = \frac{1}{L}\mathrm{X}^t\cdot\mathrm{X} = \frac{1}{L}\overline{\mathrm{R}}_x \\
\hat{\underline{\mathrm{r}}}_{ex} &= \frac{1}{L}\sum_{i=0}^{L-1} \underline{x}[k-i]\cdot e[k-i] = \frac{1}{L}\mathrm{X}^t\cdot\underline{e} = \frac{1}{L}\overline{\underline{\mathrm{r}}}_{ex}
\end{aligned}$$

with $\hat{\mathrm{R}}_x$ estimate of R_x and $\hat{\underline{\mathrm{r}}}_{ex}$ estimate of $\underline{\mathrm{r}}_{ex}$

$$\Rightarrow \quad \hat{\underline{\mathrm{w}}}_{mmse} = \left(\frac{1}{L}\overline{\mathrm{R}}_x\right)^{-1}\cdot\left(\frac{1}{L}\overline{\underline{\mathrm{r}}}_{ex}\right) = \overline{\mathrm{R}}_x^{-1}\cdot\overline{\underline{\mathrm{r}}}_{ex} = \underline{\mathrm{w}}_{ls}$$

Finally note that for ergodic processes:

$$\lim_{L\to\infty}\frac{1}{L}\overline{\mathrm{R}}_x = \mathrm{R}_x \;;\; \lim_{L\to\infty}\frac{1}{L}\overline{\underline{\mathrm{r}}}_{ex} = \underline{\mathrm{r}}_{ex} \;;\; \lim_{L\to\infty}\underline{\mathrm{w}}_{ls} = \underline{\mathrm{w}}_{mmse}$$

Steepest gradient descent (SGD)

Problem: Optimal Wiener involves R_x^{-1}

Steepest gradient descent (SGD)

Problem: Optimal Wiener involves R_x^{-1}

To avoid this inversion, estimate optimum *iteratively*

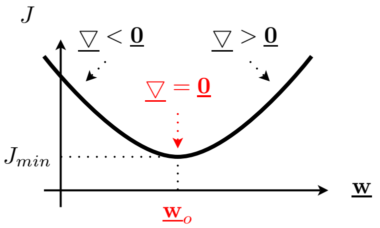
Steepest gradient descent (SGD)

Problem: Optimal Wiener involves R_x^{-1}

To avoid this inversion, estimate optimum *iteratively*

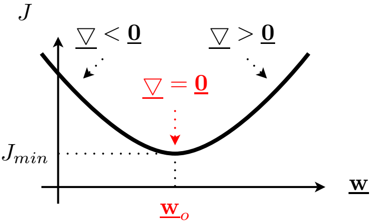
Goal: Decrease J each new iteration

SGD



J
$\underline{\nabla} < \underline{\mathbf{0}}$
$\underline{\nabla} > \underline{\mathbf{0}}$
$\underline{\nabla} = \underline{\mathbf{0}}$
J_{min}
$\underline{\mathbf{w}}$
$\underline{\mathbf{w}}_o$

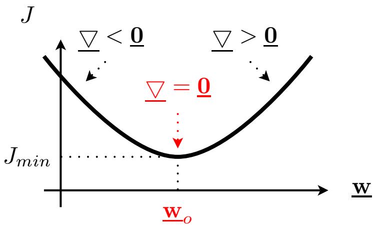
SGD



SGD principle: Update in negative gradient direction

$\Leftrightarrow \underline{w} \doteq \underline{w} - \alpha \underline{\nabla}$ with adaptation constant $\alpha \geq 0$

SGD

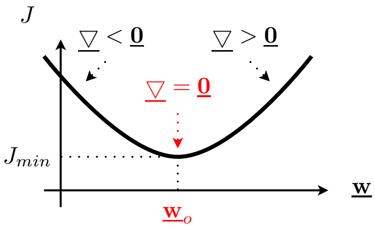


SGD principle: Update in negative gradient direction

$$\Leftrightarrow \underline{w} \doteq \underline{w} - \alpha\underline{\nabla} \text{ with adaptation constant } \alpha \geq 0$$

With $\underline{\nabla} = -2(\underline{r}_{ex} - R_x\underline{w}[k])$

SGD



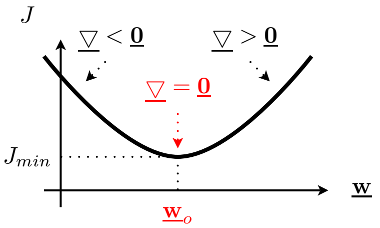
SGD principle: Update in negative gradient direction

$\Leftrightarrow \underline{w} \doteq \underline{w} - \alpha \underline{\nabla}$ with adaptation constant $\alpha \geq 0$

With $\underline{\nabla} = -2(\underline{r}_{ex} - R_x \underline{w}[k]) \;\Rightarrow$ **SGD algorithm:**

$$\boxed{\underline{w}[k+1] = \underline{w}[k] + 2\alpha(\underline{r}_{ex} - R_x \underline{w}[k])}$$

SGD



SGD principle: Update in negative gradient direction

$$\Leftrightarrow \underline{w} \doteq \underline{w} - \alpha \underline{\nabla} \text{ with adaptation constant } \alpha \geq 0$$

With $\underline{\nabla} = -2(\underline{r}_{ex} - R_x \underline{w}[k]) \Rightarrow$ **SGD algorithm:**

$$\boxed{\underline{w}[k+1] = \underline{w}[k] + 2\alpha(\underline{r}_{ex} - R_x \underline{w}[k])}$$

Notes: 1) No matrix inversion needed! 2) Usually $\underline{w}[0] = \underline{0}$

SGD

SGD converges to Wiener solution:

$$\boxed{\lim_{k\to\infty} \underline{\mathrm{w}}[k] \simeq \mathrm{R}_x^{-1} \cdot \underline{\mathrm{r}}_{ex}}$$

SGD

SGD converges to Wiener solution:

$$\boxed{\lim_{k\to\infty} \underline{w}[k] \simeq R_x^{-1} \cdot \underline{r}_{ex}}$$

'Proof':

For $k \to \infty$ we have:

$$\underline{w}[k+1] \simeq \underline{w}[k] \simeq \underline{w}[\infty]$$

SGD

SGD converges to Wiener solution:

$$\boxed{\lim_{k\to\infty} \underline{w}[k] \simeq R_x^{-1} \cdot \underline{r}_{ex}}$$

'Proof':

For $k \to \infty$ we have:

$$\begin{aligned} & \underline{w}[k+1] \simeq \underline{w}[k] \simeq \underline{w}[\infty] \\ \text{SGD} \quad \Rightarrow \quad & \underline{w}[\infty] \simeq \underline{w}[\infty] + 2\alpha(\underline{r}_{ex} - R_x \underline{w}[\infty]) \end{aligned}$$

SGD

SGD converges to Wiener solution:

$$\boxed{\lim_{k\to\infty} \underline{w}[k] \simeq R_x^{-1} \cdot \underline{r}_{ex}}$$

'Proof':

For $k \to \infty$ we have:

$$\begin{aligned} & & \underline{w}[k+1] \simeq \underline{w}[k] \simeq \underline{w}[\infty] \\ \text{SGD} & \Rightarrow & \underline{w}[\infty] \simeq \underline{w}[\infty] + 2\alpha(\underline{r}_{ex} - R_x \underline{w}[\infty]) \\ & \Rightarrow & \underline{w}[\infty] \simeq R_x^{-1} \cdot \underline{r}_{ex} \end{aligned}$$

SGD

SGD converges to Wiener solution:

$$\boxed{\lim_{k\to\infty} \underline{w}[k] \simeq R_x^{-1} \cdot \underline{r}_{ex}}$$

'Proof':

For $k \to \infty$ we have:

$$\begin{aligned} & & \underline{w}[k+1] \simeq \underline{w}[k] \simeq \underline{w}[\infty] \\ \text{SGD} & \Rightarrow & \underline{w}[\infty] \simeq \underline{w}[\infty] + 2\alpha(\underline{r}_{ex} - R_x \underline{w}[\infty]) \\ & \Rightarrow & \underline{w}[\infty] \simeq R_x^{-1} \cdot \underline{r}_{ex} \end{aligned}$$

For exact proof we need **stability analysis**

Stability SGD

Define difference weight vector: $\underline{\text{d}}[k] = \underline{\text{w}}[k] - \underline{\text{w}}_o$

Stability SGD

Define difference weight vector: $\underline{d}[k] = \underline{w}[k] - \underline{w}_o$

$$\underline{w}[k+1] \quad = \quad \underline{w}[k] + 2\alpha(\underline{r}_{ex} - R_x \underline{w}[k])$$

Stability SGD

Define difference weight vector: $\underline{d}[k] = \underline{w}[k] - \underline{w}_o$

$$\begin{aligned}
\underline{w}[k+1] &= \underline{w}[k] + 2\alpha(\underline{r}_{ex} - R_x\underline{w}[k]) \\
\underline{w}[k+1] - \underline{w}_o &= (I - 2\alpha R_x)\cdot \underline{w}[k] - \underline{w}_o + 2\alpha\underline{r}_{ex}
\end{aligned}$$

Stability SGD

Define difference weight vector: $\underline{d}[k] = \underline{w}[k] - \underline{w}_o$

$$
\begin{aligned}
\underline{w}[k+1] &= \underline{w}[k] + 2\alpha(\underline{r}_{ex} - R_x\underline{w}[k]) \\
\underline{w}[k+1] - \underline{w}_o &= (I - 2\alpha R_x) \cdot \underline{w}[k] - \underline{w}_o + 2\alpha\underline{r}_{ex} \\
\Rightarrow \ \underline{d}[k+1] &= (I - 2\alpha R_x) \cdot \underline{d}[k]
\end{aligned}
$$

Stability SGD

Define difference weight vector: $\underline{d}[k] = \underline{w}[k] - \underline{w}_o$

$$\begin{aligned} \underline{w}[k+1] &= \underline{w}[k] + 2\alpha(\underline{r}_{ex} - R_x\underline{w}[k]) \\ \underline{w}[k+1] - \underline{w}_o &= (I - 2\alpha R_x) \cdot \underline{w}[k] - \underline{w}_o + 2\alpha\underline{r}_{ex} \\ \Rightarrow \underline{d}[k+1] &= (I - 2\alpha R_x) \cdot \underline{d}[k] \end{aligned}$$

Recursion:

$$\underline{d}[k] = (I - 2\alpha R_x) \cdot \underline{d}[k-1] = \cdots = (I - 2\alpha R_x)^k \cdot \underline{d}[0]$$

Stability SGD

Define difference weight vector: $\underline{\mathrm{d}}[k] = \underline{\mathrm{w}}[k] - \underline{\mathrm{w}}_o$

$$\begin{aligned} \underline{\mathrm{w}}[k+1] &= \underline{\mathrm{w}}[k] + 2\alpha(\underline{\mathrm{r}}_{ex} - \mathrm{R}_x\underline{\mathrm{w}}[k]) \\ \underline{\mathrm{w}}[k+1] - \underline{\mathrm{w}}_o &= (\mathrm{I} - 2\alpha\mathrm{R}_x) \cdot \underline{\mathrm{w}}[k] - \underline{\mathrm{w}}_o + 2\alpha\underline{\mathrm{r}}_{ex} \\ \Rightarrow \ \underline{\mathrm{d}}[k+1] &= (\mathrm{I} - 2\alpha\mathrm{R}_x) \cdot \underline{\mathrm{d}}[k] \end{aligned}$$

Recursion:

$$\underline{\mathrm{d}}[k] = (\mathrm{I} - 2\alpha\mathrm{R}_x) \cdot \underline{\mathrm{d}}[k-1] = \cdots = (\mathrm{I} - 2\alpha\mathrm{R}_x)^k \cdot \underline{\mathrm{d}}[0]$$

$$\text{Stable iff:} \quad \lim_{k\to\infty} (\mathrm{I} - 2\alpha\mathrm{R}_x)^k = 0$$

Stability SGD

Define difference weight vector: $\underline{d}[k] = \underline{w}[k] - \underline{w}_o$

$$\begin{aligned} \underline{w}[k+1] &= \underline{w}[k] + 2\alpha(\underline{r}_{ex} - R_x\underline{w}[k]) \\ \underline{w}[k+1] - \underline{w}_o &= (I - 2\alpha R_x) \cdot \underline{w}[k] - \underline{w}_o + 2\alpha\underline{r}_{ex} \\ \Rightarrow \ \underline{d}[k+1] &= (I - 2\alpha R_x) \cdot \underline{d}[k] \end{aligned}$$

Recursion:

$$\underline{d}[k] = (I - 2\alpha R_x) \cdot \underline{d}[k-1] = \cdots = (I - 2\alpha R_x)^k \cdot \underline{d}[0]$$

$$\text{Stable iff:} \quad \lim_{k \to \infty} (I - 2\alpha R_x)^k = 0$$

Note:

When stable $\Rightarrow \underline{d}[\infty] = \underline{0} \Rightarrow \underline{w}[\infty] \simeq$ Wiener

Stability SGD

<u>How</u> do weights converge:

Stability SGD

How do weights converge:

Use eigenvalue decomposition (see Appendix):

Stability SGD

<u>How</u> do weights converge:

Use eigenvalue decomposition (see Appendix):

With $Q^h \cdot Q = Q \cdot Q^h = I$ and $R_x = Q \Lambda Q^h$

Stability SGD

<u>How</u> do weights converge:

Use eigenvalue decomposition (see Appendix):

With $\mathsf{Q}^h \cdot \mathsf{Q} = \mathsf{Q} \cdot \mathsf{Q}^h = \mathsf{I}$ and $\mathsf{R}_x = \mathsf{Q}\Lambda\mathsf{Q}^h$

$$
\begin{aligned}
\Rightarrow \quad (\mathsf{I} - 2\alpha \mathsf{R}_x)^k &= (\mathsf{Q}\mathsf{Q}^h - 2\alpha \mathsf{Q}\Lambda\mathsf{Q}^h)^k \\
&= \mathsf{Q}(\mathsf{I} - 2\alpha\Lambda)^k \mathsf{Q}^h
\end{aligned}
$$

Stability SGD

<u>How</u> do weights converge:

Use eigenvalue decomposition (see Appendix):

With $\mathsf{Q}^h \cdot \mathsf{Q} = \mathsf{Q} \cdot \mathsf{Q}^h = \mathsf{I}$ and $\mathsf{R}_x = \mathsf{Q}\Lambda\mathsf{Q}^h$

$$
\begin{aligned}
\Rightarrow \quad (\mathsf{I} - 2\alpha\mathsf{R}_x)^k &= (\mathsf{Q}\mathsf{Q}^h - 2\alpha\mathsf{Q}\Lambda\mathsf{Q}^h)^k \\
&= \mathsf{Q}(\mathsf{I} - 2\alpha\Lambda)^k\mathsf{Q}^h
\end{aligned}
$$

Change of variables: $\underline{\mathsf{D}}[k] = \mathsf{Q}^h \cdot \underline{\mathsf{d}}[k]$

Stability SGD

<u>How</u> do weights converge:

Use eigenvalue decomposition (see Appendix):

With $\mathrm{Q}^h \cdot \mathrm{Q} = \mathrm{Q} \cdot \mathrm{Q}^h = \mathrm{I}$ and $\mathrm{R}_x = \mathrm{Q}\Lambda\mathrm{Q}^h$

$$
\begin{aligned}
\Rightarrow \quad (\mathrm{I} - 2\alpha \mathrm{R}_x)^k &= (\mathrm{Q}\mathrm{Q}^h - 2\alpha \mathrm{Q}\Lambda\mathrm{Q}^h)^k \\
&= \mathrm{Q}(\mathrm{I} - 2\alpha\Lambda)^k \mathrm{Q}^h
\end{aligned}
$$

Change of variables: $\underline{\mathrm{D}}[k] = \mathrm{Q}^h \cdot \underline{\mathrm{d}}[k]$

$$
\underline{\mathrm{d}}[k] = (\mathrm{I} - 2\alpha \mathrm{R}_x)^k \underline{\mathrm{d}}[0] \quad \Rightarrow \quad \underline{\mathrm{D}}[k] = (\mathrm{I} - 2\alpha\Lambda)^k \underline{\mathrm{D}}[0]
$$

Stability SGD

<u>How</u> do weights converge:

Use eigenvalue decomposition (see Appendix):

With $\mathsf{Q}^h \cdot \mathsf{Q} = \mathsf{Q} \cdot \mathsf{Q}^h = \mathsf{I}$ and $\mathsf{R}_x = \mathsf{Q}\Lambda\mathsf{Q}^h$

$$\begin{aligned} \Rightarrow \quad (\mathsf{I} - 2\alpha \mathsf{R}_x)^k &= (\mathsf{Q}\mathsf{Q}^h - 2\alpha \mathsf{Q}\Lambda\mathsf{Q}^h)^k \\ &= \mathsf{Q}(\mathsf{I} - 2\alpha\Lambda)^k \mathsf{Q}^h \end{aligned}$$

Change of variables: $\underline{\mathsf{D}}[k] = \mathsf{Q}^h \cdot \underline{\mathsf{d}}[k]$

$$\underline{\mathsf{d}}[k] = (\mathsf{I} - 2\alpha \mathsf{R}_x)^k \underline{\mathsf{d}}[0] \quad \Rightarrow \quad \underline{\mathsf{D}}[k] = (\mathsf{I} - 2\alpha\Lambda)^k \underline{\mathsf{D}}[0]$$

Recursion stable iff: $\lim_{k \to \infty} (\mathsf{I} - 2\alpha\Lambda)^k = 0$

Stability SGD

$$\text{Recursion stable iff: } \lim_{k\to\infty} (\mathsf{I} - 2\alpha\Lambda)^k = 0$$

Stability SGD

$$\text{Recursion stable iff:} \quad \lim_{k \to \infty} (\mathsf{I} - 2\alpha\Lambda)^k = 0$$

Both matrices I and Λ diagonal

Stability SGD

$$\text{Recursion stable iff:} \quad \lim_{k\to\infty} (\mathsf{I} - 2\alpha\Lambda)^k = 0$$

Both matrices I and Λ diagonal $\Rightarrow$ Stable iff:

$$|1 - 2\alpha\lambda_i| < 1 \quad \Leftrightarrow \quad 0 < \alpha < \frac{1}{\lambda_i} \quad \text{for } i = 0, 1, \cdots, N-1$$

Stability SGD

Recursion stable iff: $\lim_{k\to\infty} (\mathbf{I} - 2\alpha\Lambda)^k = 0$

Both matrices I and Λ diagonal $\Rightarrow$ Stable iff:

$$|1 - 2\alpha\lambda_i| < 1 \quad \Leftrightarrow \quad 0 < \alpha < \frac{1}{\lambda_i} \quad \text{for } i = 0, 1, \cdots, N-1$$

Thus SGD algorithm stable iff: $\boxed{0 < \alpha < \frac{1}{\lambda_{max}}}$

Stability SGD

$$\text{Recursion stable iff:} \quad \lim_{k\to\infty} (\mathrm{I} - 2\alpha\Lambda)^k = 0$$

Both matrices I and Λ diagonal $\Rightarrow$ Stable iff:

$$|1 - 2\alpha\lambda_i| < 1 \quad \Leftrightarrow \quad 0 < \alpha < \frac{1}{\lambda_i} \quad \text{for } i = 0, 1, \cdots, N-1$$

$$\text{Thus SGD algorithm stable iff:} \quad \boxed{0 < \alpha < \frac{1}{\lambda_{max}}}$$

For adaptation constant α in this region:

$$\lim_{k\to\infty} \underline{\mathrm{w}}[k] = \underline{\mathrm{w}}_o = \mathrm{R}_x^{-1} \cdot \underline{\mathrm{r}}_{ex}$$

Stability SGD

$$\text{Recursion stable iff:} \quad \lim_{k\to\infty} (\mathrm{I} - 2\alpha\Lambda)^k = 0$$

Both matrices I and Λ diagonal $\Rightarrow$ Stable iff:

$$|1 - 2\alpha\lambda_i| < 1 \quad \Leftrightarrow \quad 0 < \alpha < \frac{1}{\lambda_i} \quad \text{for } i = 0, 1, \cdots, N-1$$

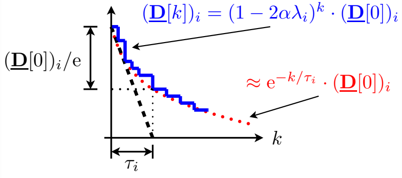
$$\text{Thus SGD algorithm stable iff:} \quad \boxed{0 < \alpha < \frac{1}{\lambda_{max}}}$$

For adaptation constant α in this region:

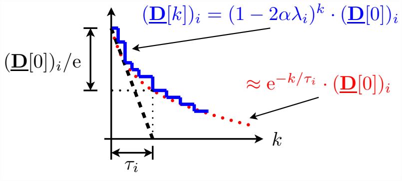
$$\lim_{k\to\infty} \underline{\mathrm{w}}[k] = \underline{\mathrm{w}}_o = \mathrm{R}_x^{-1} \cdot \underline{\mathrm{r}}_{ex}$$

$$J_{\underline{\mathrm{w}}=\underline{\mathrm{w}}_o} = E\{r^2[k]\} = J_{min} = E\{e^2\} - \underline{\mathrm{r}}_{ex}^t \mathrm{R}_x^{-1} \underline{\mathrm{r}}_{ex}$$

Convergence SGD

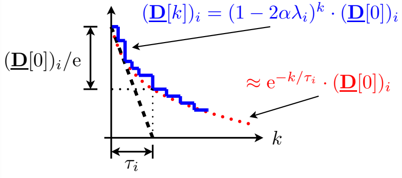


Convergence SGD



$$\mathrm{e}^{-k/\tau_i} \cdot (\underline{D}[0])_i \approx (1 - 2\alpha\lambda_i)^k \cdot (\underline{D}[0])_i \quad \Rightarrow$$

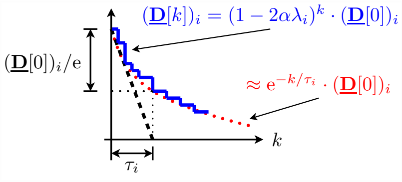
Convergence SGD



$$\mathrm{e}^{-k/\tau_i} \cdot (\underline{D}[0])_i \approx (1 - 2\alpha\lambda_i)^k \cdot (\underline{D}[0])_i \quad \Rightarrow$$

Average behavior: $\tau_{av,i} = \dfrac{-1}{\ln(1 - 2\alpha\lambda_i)}$

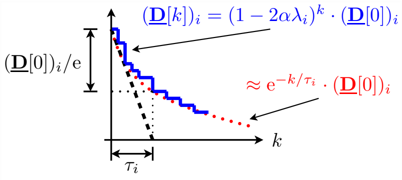
Convergence SGD



$$e^{-k/\tau_i} \cdot (\underline{D}[0])_i \approx (1-2\alpha\lambda_i)^k \cdot (\underline{D}[0])_i \quad \Rightarrow$$

Average behavior: $\tau_{av,i} = \dfrac{-1}{\ln(1-2\alpha\lambda_i)}$ $\Rightarrow$ For small α $\boxed{\tau_{av,i} \approx \dfrac{1}{2\alpha\lambda_i}}$

Convergence SGD

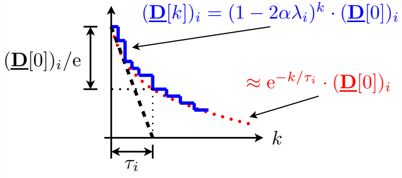


$$\mathrm{e}^{-k/\tau_i} \cdot (\underline{\mathrm{D}}[0])_i \approx (1-2\alpha\lambda_i)^k \cdot (\underline{\mathrm{D}}[0])_i \quad \Rightarrow$$

Average behavior: $\tau_{av,i} = \dfrac{-1}{\ln(1-2\alpha\lambda_i)}$ $\Rightarrow$ For small α $\boxed{\tau_{av,i} \approx \dfrac{1}{2\alpha\lambda_i}}$

Note:
Overall time constant depends on eigenvalue spread $\Gamma_x = \lambda_{max}/\lambda_{min}$. Thus, the larger Γ_x the longer it takes for adaptation.

Convergence SGD



$$\mathrm{e}^{-k/\tau_i} \cdot (\underline{\mathrm{D}}[0])_i \approx (1-2\alpha\lambda_i)^k \cdot (\underline{\mathrm{D}}[0])_i \quad \Rightarrow$$

Average behavior: $\tau_{av,i} = \dfrac{-1}{\ln(1-2\alpha\lambda_i)}$ $\Rightarrow$ For small α $\boxed{\tau_{av,i} \approx \dfrac{1}{2\alpha\lambda_i}}$

Note:
Overall time constant depends on eigenvalue spread $\Gamma_x = \lambda_{max}/\lambda_{min}$. Thus, the larger Γ_x the longer it takes for adaptation.

Q: What happens for white noise?

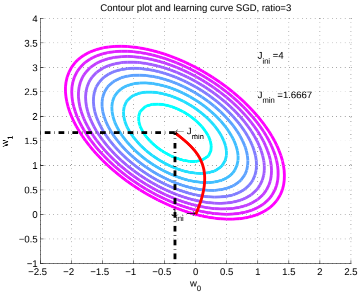
Convergence SGD

Example with $\Gamma_x = \lambda_{max}/\lambda_{min} = 3$

Convergence SGD

Example with $\Gamma_x = \lambda_{max}/\lambda_{min} = 3$

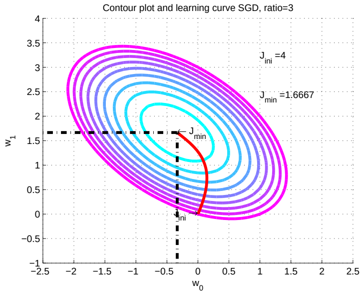
Learning curve in contour plot J



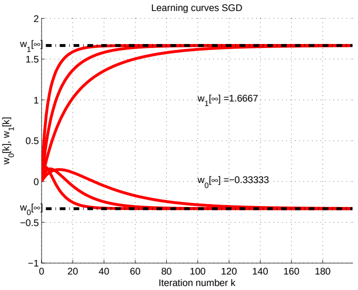
Convergence SGD

Example with $\Gamma_x = \lambda_{max}/\lambda_{min} = 3$

Learning curve in contour plot J



Learning curves for different α



Least Mean Square (LMS)

Motivation: SGD not practical. Gradient assumes **known** R_x and $\underline{r}_{ex}$

Least Mean Square (LMS)

Motivation: SGD not practical. Gradient assumes **known** R_x and $\underline{r}_{ex}$

LMS principle: Use instantaneous estimate of gradient:

$$\begin{aligned}\hat{\underline{\nabla}}[k] &= -2\left(e[k]\underline{x}[k] - \underline{x}[k]\underline{x}^t[k]\underline{w}[k]\right) \\ &= -2\underline{x}[k]\left(e[k] - \underline{x}^t[k]\underline{w}[k]\right) = -2\underline{x}[k]r[k]\end{aligned}$$

Least Mean Square (LMS)

Motivation: SGD not practical. Gradient assumes **known** R_x and $\underline{r}_{ex}$

LMS principle: Use instantaneous estimate of gradient:

$$\begin{aligned} \underline{\hat{\nabla}}[k] &= -2\left(e[k]\underline{x}[k] - \underline{x}[k]\underline{x}^t[k]\underline{w}[k]\right) \\ &= -2\underline{x}[k]\left(e[k] - \underline{x}^t[k]\underline{w}[k]\right) = -2\underline{x}[k]r[k] \end{aligned}$$

With $\underline{w} \doteq \underline{w} - \underline{\hat{\nabla}} \Rightarrow$ LMS algorithm (Widrow, 1975):

$$\boxed{\begin{array}{lcl} k = 0 & : & \underline{w}[0] = \underline{0} \quad \text{(usually)} \\ k > 0 & : & \hat{e}[k] = \underline{w}^t[k] \cdot \underline{x}[k] \\ & & r[k] = e[k] - \hat{e}[k] \\ & & \underline{w}[k+1] = \underline{w}[k] + 2\alpha\underline{x}[k]r[k] \end{array}}$$

Least Mean Square (LMS)

Motivation: SGD not practical. Gradient assumes **known** R_x and $\underline{r}_{ex}$

LMS principle: Use instantaneous estimate of gradient:

$$\begin{aligned}\underline{\hat{\nabla}}[k] &= -2\left(e[k]\underline{x}[k] - \underline{x}[k]\underline{x}^t[k]\underline{w}[k]\right) \\ &= -2\underline{x}[k]\left(e[k] - \underline{x}^t[k]\underline{w}[k]\right) = -2\underline{x}[k]r[k]\end{aligned}$$

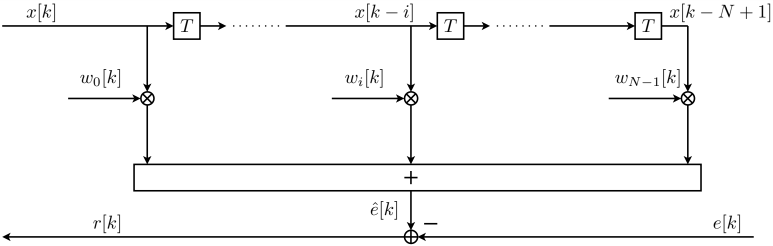
With $\underline{w} \doteq \underline{w} - \underline{\hat{\nabla}} \Rightarrow$ LMS algorithm (Widrow, 1975):

$$\boxed{\begin{array}{lcl} k = 0 & : & \underline{w}[0] = \underline{0} \quad \text{(usually)} \\ k > 0 & : & \hat{e}[k] = \underline{w}^t[k] \cdot \underline{x}[k] \\ & & r[k] = e[k] - \hat{e}[k] \\ & & \underline{w}[k+1] = \underline{w}[k] + 2\alpha\underline{x}[k]r[k] \end{array}}$$

Note: $\underline{w}^t[k] \cdot \underline{x}[k]$ is "convolution" and $\underline{x}[k]r[k]$ "correlation"

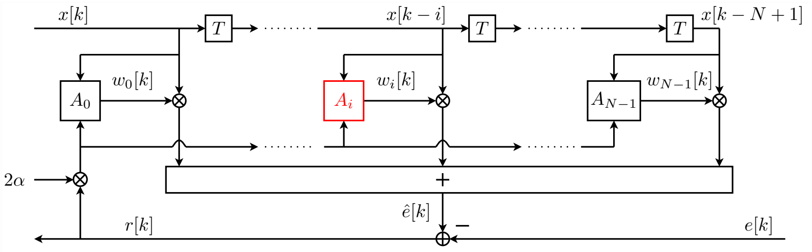
LMS

Realization scheme LMS algorithm:



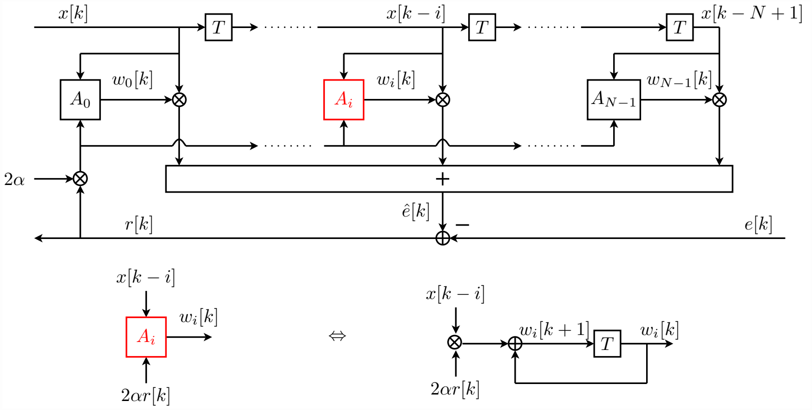
LMS

Realization scheme LMS algorithm:



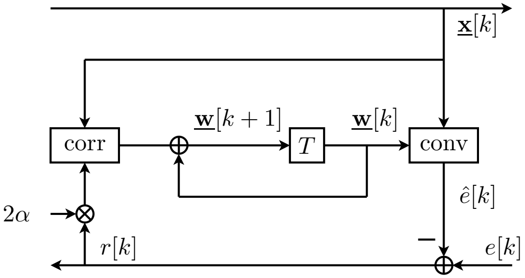
LMS

Realization scheme LMS algorithm:



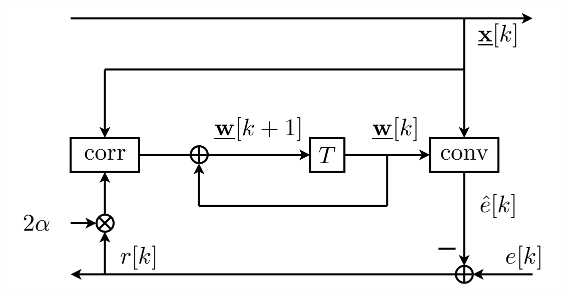
LMS

Simplified realization scheme LMS algorithm:



LMS

Simplified realization scheme LMS algorithm:



Notes:

- Simple, robust algorithm, complexity $O(2N)$
- LMS tries to "decorrelate" signals x and r
- In contrast to SGD: Weights fluctuate around optimal values

Two LMS variants

Two LMS variants

- **NLMS:** LMS with normalization by $\sigma_x^2 = E\{x^2[k]\}$:

$$\boxed{\underline{\mathrm{w}}[k+1] = \underline{\mathrm{w}}[k] + \frac{2\alpha}{\sigma_x^2}\underline{\mathrm{x}}[k]r[k]}$$

Two LMS variants

- **NLMS:** LMS with normalization by $\sigma_x^2 = E\{x^2[k]\}$:

$$\boxed{\underline{w}[k+1] = \underline{w}[k] + \frac{2\alpha}{\sigma_x^2}\underline{x}[k]r[k]}$$

In practice $\hat{\sigma}_x^2[k] \Rightarrow$ time-varying step size. E.g.:

- $\hat{\sigma}_x^2[k] = \beta\hat{\sigma}_x^2[k-1] + (1-\beta)\frac{\underline{x}^t[k]\underline{x}[k]}{N}$ with $0 < \beta < 1$
- $\hat{\sigma}_x^2[k] = \frac{\underline{x}^t[k]\underline{x}[k]}{N} + \epsilon$ with ϵ some small constant

Two LMS variants

- **NLMS:** LMS with normalization by $\sigma_x^2 = E\{x^2[k]\}$:

$$\boxed{\underline{\mathrm{w}}[k+1] = \underline{\mathrm{w}}[k] + \frac{2\alpha}{\sigma_x^2}\underline{\mathrm{x}}[k]r[k]}$$

  In practice $\hat{\sigma}_x^2[k] \Rightarrow$ time-varying step size. E.g.:
  - $\hat{\sigma}_x^2[k] = \beta\hat{\sigma}_x^2[k-1] + (1-\beta)\frac{\underline{\mathrm{x}}^t[k]\underline{\mathrm{x}}[k]}{N}$ with $0 < \beta < 1$
  - $\hat{\sigma}_x^2[k] = \frac{\underline{\mathrm{x}}^t[k]\underline{\mathrm{x}}[k]}{N} + \epsilon$ with ϵ some small constant
- **Complex LMS:** LMS for complex signals and weights:

$$\boxed{\underline{\mathrm{w}}[k+1] = \underline{\mathrm{w}}[k] + 2\alpha\underline{\mathrm{x}}[k]r^*[k]}$$

Newton

Convergence gradient based algorithms relies on coloration input:

$$\underline{\nabla} = -2\left(\underline{r}_{ex} - R_x \underline{w}[k]\right)$$

Newton

Convergence gradient based algorithms relies on coloration input:

$$\underline{\nabla} = -2\left(\underline{r}_{ex} - R_x \underline{w}[k]\right)$$

Solution Newton: $\underline{w}[k+1] = \underline{w}[k] - \alpha R_x^{-1} \underline{\nabla} \Rightarrow$

$$\boxed{\underline{w}[k+1] = \underline{w}[k] + 2\alpha R_x^{-1} \cdot \left(\underline{r}_{ex} - R_x \underline{w}[k]\right)}$$

Newton

Convergence gradient based algorithms relies on coloration input:

$$\underline{\nabla} = -2\left(\underline{\mathrm{r}}_{ex} - \mathrm{R}_x \underline{\mathrm{w}}[k]\right)$$

Solution Newton: $\underline{\mathrm{w}}[k+1] = \underline{\mathrm{w}}[k] - \alpha \mathrm{R}_x^{-1} \underline{\nabla} \Rightarrow$

$$\boxed{\underline{\mathrm{w}}[k+1] = \underline{\mathrm{w}}[k] + 2\alpha \mathrm{R}_x^{-1} \cdot \left(\underline{\mathrm{r}}_{ex} - \mathrm{R}_x \underline{\mathrm{w}}[k]\right)}$$

Convergence Newton:

$$\underline{\mathrm{d}}[k+1] = \left(\mathrm{I} - 2\alpha \mathrm{R}_x^{-1} \mathrm{R}_x\right) \underline{\mathrm{d}}[k] = (1-2\alpha)\underline{\mathrm{d}}[k] \quad \Rightarrow \quad \text{Convergence } 0 < \alpha < 1$$

Newton

Convergence gradient based algorithms relies on coloration input:

$$\underline{\nabla} = -2\left(\underline{r}_{ex} - R_x \underline{w}[k]\right)$$

Solution Newton: $\underline{w}[k+1] = \underline{w}[k] - \alpha R_x^{-1} \underline{\nabla} \Rightarrow$

$$\boxed{\underline{w}[k+1] = \underline{w}[k] + 2\alpha R_x^{-1} \cdot \left(\underline{r}_{ex} - R_x \underline{w}[k]\right)}$$

Convergence Newton:

$$\underline{d}[k+1] = \left(I - 2\alpha R_x^{-1} R_x\right) \underline{d}[k] = (1-2\alpha)\underline{d}[k] \quad \Rightarrow \quad \text{Convergence } 0 < \alpha < 1$$

Notes:

- R_x^{-1} causes whitening of input process

Newton

Convergence gradient based algorithms relies on coloration input:

$$\underline{\nabla} = -2\left(\underline{r}_{ex} - R_x \underline{w}[k]\right)$$

Solution Newton: $\underline{w}[k+1] = \underline{w}[k] - \alpha R_x^{-1} \underline{\nabla} \Rightarrow$

$$\boxed{\underline{w}[k+1] = \underline{w}[k] + 2\alpha R_x^{-1} \cdot \left(\underline{r}_{ex} - R_x \underline{w}[k]\right)}$$

Convergence Newton:

$$\underline{d}[k+1] = \left(I - 2\alpha R_x^{-1} R_x\right) \underline{d}[k] = (1-2\alpha)\underline{d}[k] \quad \Rightarrow \quad \text{Convergence } 0 < \alpha < 1$$

Notes:

- R_x^{-1} causes whitening of input process
- All weights have same convergence (in contrast to LMS, SGD)

Newton

Convergence gradient based algorithms relies on coloration input:

$$\underline{\nabla} = -2\left(\underline{r}_{ex} - R_x \underline{w}[k]\right)$$

Solution Newton: $\underline{w}[k+1] = \underline{w}[k] - \alpha R_x^{-1} \underline{\nabla} \Rightarrow$

$$\boxed{\underline{w}[k+1] = \underline{w}[k] + 2\alpha R_x^{-1} \cdot \left(\underline{r}_{ex} - R_x \underline{w}[k]\right)}$$

Convergence Newton:

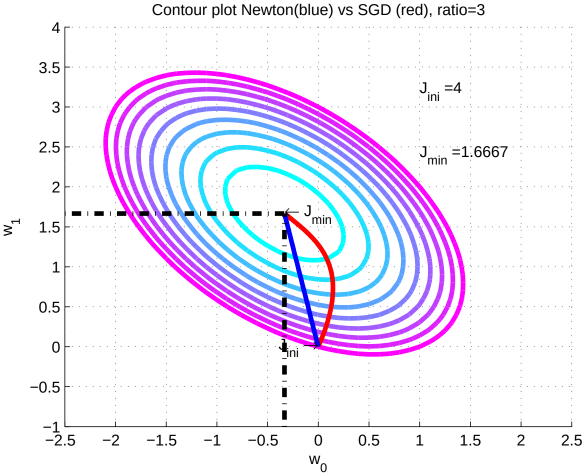
$$\underline{d}[k+1] = \left(I - 2\alpha R_x^{-1} R_x\right) \underline{d}[k] = (1-2\alpha)\underline{d}[k] \quad \Rightarrow \quad \text{Convergence } 0 < \alpha < 1$$

Notes:

- R_x^{-1} causes whitening of input process
- All weights have same convergence (in contrast to LMS, SGD)
- Newton $\equiv$ SGD with white noise input!

Newton

Learning curves in contour plot: Newton vs. SGD



Newton: another view

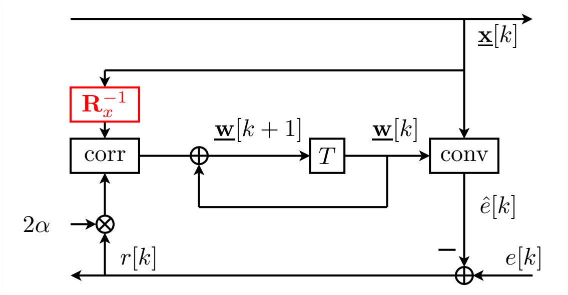
Replace $\underline{\nabla}$ by $\hat{\underline{\nabla}}_{LMS} = \underline{x}[k]r[k] \Rightarrow$ LMS/Newton:

$$\boxed{\underline{w}[k+1] = \underline{w}[k] + 2\alpha R_x^{-1}\underline{x}[k]r[k]}$$

Newton: another view

Replace $\underline{\nabla}$ by $\hat{\underline{\nabla}}_{LMS} = \underline{x}[k]r[k] \Rightarrow$ LMS/Newton:

$$\boxed{\underline{w}[k+1] = \underline{w}[k] + 2\alpha R_x^{-1} \underline{x}[k] r[k]}$$



Newton: Practical problem

Autocorrelation matrix R_x:

Newton: Practical problem

Autocorrelation matrix R_x:

- (In general) not known in advance
- May change during time (non-stationary process)
- Inversion is expensive (many MIPS)

Newton: Practical problem

Autocorrelation matrix R_x:

- (In general) not known in advance
- May change during time (non-stationary process)
- Inversion is expensive (many MIPS)

$\Rightarrow$ Complexity Newton algorithm huge

$\Rightarrow$ Need for efficient solution with estimate of R_x

$\Rightarrow$ Different algorithms, e.g. RLS, FDAF, etc.

Recursive Least Squares (RLS)

For data block length L fixed, Least Squares problem becomes:

$$\min_{\underline{w}[k]} \left|\underline{e}[k] - X[k] \cdot \underline{w}[k]\right|^2 \quad \Rightarrow \quad \underline{w}_{LS}[k] = \left(X^t[k]X[k]\right)^{-1} \left(X^t[k]\underline{e}[k]\right)$$

Recursive Least Squares (RLS)

For data block length L fixed, Least Squares problem becomes:

$$\min_{\underline{\mathrm{w}}[k]} \left|\underline{\mathrm{e}}[k] - \mathrm{X}[k] \cdot \underline{\mathrm{w}}[k]\right|^2 \quad \Rightarrow \quad \underline{\mathrm{w}}_{LS}[k] = \left(\mathrm{X}^t[k]\mathrm{X}[k]\right)^{-1} \left(\mathrm{X}^t[k]\underline{\mathrm{e}}[k]\right)$$

RLS: Find efficient recursive solution for LS problem from $k \rightarrow k+1$

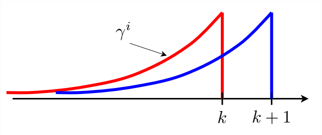
Recursive Least Squares (RLS)

For data block length L fixed, Least Squares problem becomes:

$$\min_{\underline{\mathrm{w}}[k]} \|\underline{\mathrm{e}}[k] - \mathrm{X}[k] \cdot \underline{\mathrm{w}}[k]\|^2 \quad \Rightarrow \quad \underline{\mathrm{w}}_{LS}[k] = \left(\mathrm{X}^t[k]\mathrm{X}[k]\right)^{-1} \left(\mathrm{X}^t[k]\underline{\mathrm{e}}[k]\right)$$

RLS: Find efficient recursive solution for LS problem from $k \to k+1$

Use exponential sliding window: Scale down data by factor γ



$$\text{Forgetting factor} \quad : \quad 0 < \gamma < 1$$
$$\text{'Memory'} \quad : \quad \frac{1}{1-\gamma}$$

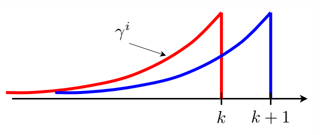
Recursive Least Squares (RLS)

For data block length L fixed, Least Squares problem becomes:

$$\min_{\underline{\mathrm{w}}[k]} |\underline{\mathrm{e}}[k] - \mathrm{X}[k] \cdot \underline{\mathrm{w}}[k]|^2 \quad \Rightarrow \quad \underline{\mathrm{w}}_{LS}[k] = \left(\mathrm{X}^t[k]\mathrm{X}[k]\right)^{-1} \left(\mathrm{X}^t[k]\underline{\mathrm{e}}[k]\right)$$

RLS: Find efficient recursive solution for LS problem from $k \rightarrow k+1$

Use exponential sliding window: Scale down data by factor γ



$$\begin{aligned} \text{Forgetting factor} &: \quad 0 < \gamma < 1 \\ \text{'Memory'} &: \quad \frac{1}{1-\gamma} \end{aligned}$$

$$\mathrm{X}[k] = \begin{pmatrix} \gamma^0 \underline{x}^t[k] \\ \cdots \\ \gamma^i \underline{x}^t[k-i] \\ \cdots \\ \gamma^k \underline{x}^t[0] \end{pmatrix} \text{ and } \underline{\mathrm{e}}[k] = \begin{pmatrix} \gamma^0 e[k] \\ \cdots \\ \gamma^i e[k-i] \\ \cdots \\ \gamma^k e[0] \end{pmatrix}$$

RLS algorithm

RLS algorithm

Initialization: $\underline{\bar{r}}_{ex}[0] = \underline{0}$; $\overline{R}_x^{-1}[0] = \delta^{-1} I$ with δ small

RLS algorithm

Initialization: $\underline{\overline{\mathrm{r}}}_{ex}[0] = \underline{0}$; $\overline{\mathrm{R}}_x^{-1}[0] = \delta^{-1}\mathrm{I}$ with δ small

For $k \geq 0$:

RLS algorithm

Initialization: $\underline{\bar{r}}_{ex}[0] = \underline{0}$; $\overline{R}_x^{-1}[0] = \delta^{-1}I$ with δ small

For $k \geq 0$:

$$\overline{R}_x^{-1}[k+1] \quad = \quad \gamma^{-2}\left(\overline{R}_x^{-1}[k] - \underline{g}[k+1] \cdot \underline{x}^t[k+1]\overline{R}_x^{-1}[k]\right)$$

RLS algorithm

Initialization: $\underline{\bar{r}}_{ex}[0] = \underline{0}$; $\overline{R}_x^{-1}[0] = \delta^{-1}I$ with δ small

For $k \geq 0$:

$$\underline{g}[k+1] = \frac{\overline{R}_x^{-1}[k]\underline{x}[k+1]}{\gamma^2 + \underline{x}^t[k+1]\overline{R}_x^{-1}[k]\underline{x}[k+1]}$$

$$\overline{R}_x^{-1}[k+1] = \gamma^{-2}\left(\overline{R}_x^{-1}[k] - \underline{g}[k+1]\cdot\underline{x}^t[k+1]\overline{R}_x^{-1}[k]\right)$$

RLS algorithm

Initialization: $\underline{\overline{r}}_{ex}[0] = \underline{0}$; $\overline{R}_x^{-1}[0] = \delta^{-1}I$ with δ small

For $k \geq 0$:

$$\begin{aligned}
\underline{g}[k+1] &= \frac{\overline{R}_x^{-1}[k]\underline{x}[k+1]}{\gamma^2 + \underline{x}^t[k+1]\overline{R}_x^{-1}[k]\underline{x}[k+1]} \\
\overline{R}_x^{-1}[k+1] &= \gamma^{-2}\left(\overline{R}_x^{-1}[k] - \underline{g}[k+1]\cdot\underline{x}^t[k+1]\overline{R}_x^{-1}[k]\right) \\
\underline{\overline{r}}_{ex}[k+1] &= \gamma^2\underline{\overline{r}}_{ex}[k] + \underline{x}[k+1]\cdot e[k+1]
\end{aligned}$$

RLS algorithm

Initialization: $\underline{\overline{r}}_{ex}[0] = \underline{0}$; $\overline{R}_x^{-1}[0] = \delta^{-1}I$ with δ small

For $k \geq 0$:

$$
\begin{aligned}
\underline{g}[k+1] &= \frac{\overline{R}_x^{-1}[k]\underline{x}[k+1]}{\gamma^2 + \underline{x}^t[k+1]\overline{R}_x^{-1}[k]\underline{x}[k+1]} \\
\overline{R}_x^{-1}[k+1] &= \gamma^{-2}\left(\overline{R}_x^{-1}[k] - \underline{g}[k+1]\cdot\underline{x}^t[k+1]\overline{R}_x^{-1}[k]\right) \\
\underline{\overline{r}}_{ex}[k+1] &= \gamma^2\underline{\overline{r}}_{ex}[k] + \underline{x}[k+1]\cdot e[k+1] \\
\underline{w}[k+1] &= \overline{R}_x^{-1}[k+1]\cdot\underline{r}_{ex}[k+1]
\end{aligned}
$$

RLS algorithm

Initialization: $\underline{\overline{r}}_{ex}[0] = \underline{0}$; $\overline{R}_x^{-1}[0] = \delta^{-1}I$ with δ small

For $k \geq 0$:

$$\begin{aligned}
\underline{g}[k+1] &= \frac{\overline{R}_x^{-1}[k]\underline{x}[k+1]}{\gamma^2 + \underline{x}^t[k+1]\overline{R}_x^{-1}[k]\underline{x}[k+1]} \\
\overline{R}_x^{-1}[k+1] &= \gamma^{-2}\left(\overline{R}_x^{-1}[k] - \underline{g}[k+1]\cdot\underline{x}^t[k+1]\overline{R}_x^{-1}[k]\right) \\
\underline{\overline{r}}_{ex}[k+1] &= \gamma^2\underline{\overline{r}}_{ex}[k] + \underline{x}[k+1]\cdot e[k+1] \\
\underline{w}[k+1] &= \overline{R}_x^{-1}[k+1]\cdot\underline{r}_{ex}[k+1]
\end{aligned}$$

Notes:

- $\underline{w}[\infty] \to \underline{w}_o$

RLS algorithm

Initialization: $\overline{\underline{r}}_{ex}[0] = \underline{0}$; $\overline{R}_x^{-1}[0] = \delta^{-1} I$ with δ small

For $k \geq 0$:

$$\begin{aligned}
\underline{g}[k+1] &= \frac{\overline{R}_x^{-1}[k]\underline{x}[k+1]}{\gamma^2 + \underline{x}^t[k+1]\overline{R}_x^{-1}[k]\underline{x}[k+1]} \\
\overline{R}_x^{-1}[k+1] &= \gamma^{-2}\left(\overline{R}_x^{-1}[k] - \underline{g}[k+1]\cdot\underline{x}^t[k+1]\overline{R}_x^{-1}[k]\right) \\
\overline{\underline{r}}_{ex}[k+1] &= \gamma^2\overline{\underline{r}}_{ex}[k] + \underline{x}[k+1]\cdot e[k+1] \\
\underline{w}[k+1] &= \overline{R}_x^{-1}[k+1]\cdot\underline{r}_{ex}[k+1]
\end{aligned}$$

Notes:

- $\underline{w}[\infty] \to \underline{w}_o$
- Complexity $O(N^2)$ per time update

RLS algorithm

Initialization: $\underline{\overline{r}}_{ex}[0] = \underline{0}$; $\overline{R}_x^{-1}[0] = \delta^{-1}I$ with δ small

For $k \geq 0$:

$$
\begin{aligned}
\underline{g}[k+1] &= \frac{\overline{R}_x^{-1}[k]\underline{x}[k+1]}{\gamma^2 + \underline{x}^t[k+1]\overline{R}_x^{-1}[k]\underline{x}[k+1]} \\
\overline{R}_x^{-1}[k+1] &= \gamma^{-2}\left(\overline{R}_x^{-1}[k] - \underline{g}[k+1]\cdot\underline{x}^t[k+1]\overline{R}_x^{-1}[k]\right) \\
\underline{\overline{r}}_{ex}[k+1] &= \gamma^2\underline{\overline{r}}_{ex}[k] + \underline{x}[k+1]\cdot e[k+1] \\
\underline{w}[k+1] &= \overline{R}_x^{-1}[k+1]\cdot\underline{r}_{ex}[k+1]
\end{aligned}
$$

Notes:

- $\underline{w}[\infty] \to \underline{w}_o$
- Complexity $O(N^2)$ per time update
- Window length increases when time increases!

RLS algorithm

Initialization: $\underline{\overline{r}}_{ex}[0] = \underline{0}$; $\overline{R}_x^{-1}[0] = \delta^{-1}I$ with δ small

For $k \geq 0$:

$$
\begin{aligned}
\underline{g}[k+1] &= \frac{\overline{R}_x^{-1}[k]\underline{x}[k+1]}{\gamma^2 + \underline{x}^t[k+1]\overline{R}_x^{-1}[k]\underline{x}[k+1]} \\
\overline{R}_x^{-1}[k+1] &= \gamma^{-2}\left(\overline{R}_x^{-1}[k] - \underline{g}[k+1]\cdot\underline{x}^t[k+1]\overline{R}_x^{-1}[k]\right) \\
\underline{\overline{r}}_{ex}[k+1] &= \gamma^2\underline{\overline{r}}_{ex}[k] + \underline{x}[k+1]\cdot e[k+1] \\
\underline{w}[k+1] &= \overline{R}_x^{-1}[k+1]\cdot\underline{r}_{ex}[k+1]
\end{aligned}
$$

Notes:

- $\underline{w}[\infty] \rightarrow \underline{w}_o$
- Complexity $O(N^2)$ per time update
- Window length increases when time increases!
- Exhibits unstable roundoff error accumulation

RLS algorithm

Initialization: $\underline{\bar{\mathrm{r}}}_{ex}[0] = \underline{0}$; $\bar{\mathrm{R}}_x^{-1}[0] = \delta^{-1}\mathrm{I}$ with δ small

For $k \geq 0$:

$$
\begin{aligned}
\underline{\mathrm{g}}[k+1] &= \frac{\bar{\mathrm{R}}_x^{-1}[k]\underline{\mathrm{x}}[k+1]}{\gamma^2 + \underline{\mathrm{x}}^t[k+1]\bar{\mathrm{R}}_x^{-1}[k]\underline{\mathrm{x}}[k+1]} \\
\bar{\mathrm{R}}_x^{-1}[k+1] &= \gamma^{-2}\left(\bar{\mathrm{R}}_x^{-1}[k] - \underline{\mathrm{g}}[k+1]\cdot\underline{\mathrm{x}}^t[k+1]\bar{\mathrm{R}}_x^{-1}[k]\right) \\
\underline{\bar{\mathrm{r}}}_{ex}[k+1] &= \gamma^2\underline{\bar{\mathrm{r}}}_{ex}[k] + \underline{\mathrm{x}}[k+1]\cdot e[k+1] \\
\underline{\mathrm{w}}[k+1] &= \bar{\mathrm{R}}_x^{-1}[k+1]\cdot\underline{\mathrm{r}}_{ex}[k+1]
\end{aligned}
$$

Notes:

- $\underline{\mathrm{w}}[\infty] \rightarrow \underline{\mathrm{w}}_o$
- Complexity $O(N^2)$ per time update
- Window length increases when time increases!
- Exhibits unstable roundoff error accumulation
- RLS is basis for many practical algorithms

RLS algorithm

Initialization: $\underline{\bar{r}}_{ex}[0] = \underline{0}$; $\overline{R}_x^{-1}[0] = \delta^{-1}I$ with δ small

For $k \geq 0$:

$$
\begin{aligned}
\underline{g}[k+1] &= \frac{\overline{R}_x^{-1}[k]\underline{x}[k+1]}{\gamma^2 + \underline{x}^t[k+1]\overline{R}_x^{-1}[k]\underline{x}[k+1]} \\
\overline{R}_x^{-1}[k+1] &= \gamma^{-2}\left(\overline{R}_x^{-1}[k] - \underline{g}[k+1]\cdot\underline{x}^t[k+1]\overline{R}_x^{-1}[k]\right) \\
\underline{\bar{r}}_{ex}[k+1] &= \gamma^2\underline{\bar{r}}_{ex}[k] + \underline{x}[k+1]\cdot e[k+1] \\
\underline{w}[k+1] &= \overline{R}_x^{-1}[k+1]\cdot\underline{r}_{ex}[k+1]
\end{aligned}
$$

Notes:

- $\underline{w}[\infty] \to \underline{w}_o$
- Complexity $O(N^2)$ per time update
- Window length increases when time increases!
- Exhibits unstable roundoff error accumulation
- RLS is basis for many practical algorithms
- Decorrelation takes place in algorithm

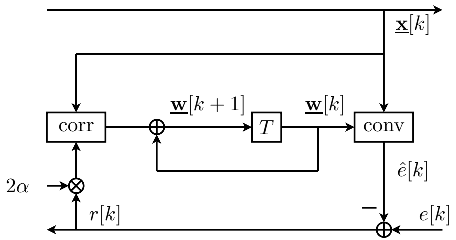
Frequency Domain Adaptive Filter (FDAF)

FDAF: Alternative for LMS/Newton and RLS

Frequency Domain Adaptive Filter (FDAF)

FDAF: Alternative for LMS/Newton and RLS

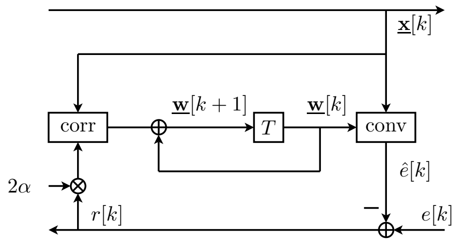
First step of derivation: Translate LMS to frequency domain



Frequency Domain Adaptive Filter (FDAF)

FDAF: Alternative for LMS/Newton and RLS

First step of derivation: Translate LMS to frequency domain



LMS weight update:

$$\underline{w}[k+1] = \underline{w}[k] + 2\alpha \underline{x}[k] r[k]$$

Filter output:

$$\hat{e}[k] = \underline{x}^t[k] \cdot \underline{w}[k]$$

FDAF

Transform vectors to frequency domain:

FDAF

Transform vectors to frequency domain:

$$\begin{aligned} \mathsf{F} \cdot \underline{\mathrm{x}}[k] = \underline{\mathrm{X}}[k] &= (X_0[k], X_1[k], \cdots, X_{N-1}[k])^t \\ \mathsf{F}^{-1} \cdot \underline{\mathrm{w}}[k] = \underline{\mathrm{W}}[k] &= (W_0[k], W_1[k], \cdots, W_{N-1}[k])^t \end{aligned}$$

with DFT matrix F, which has properties: $\mathsf{F} = \mathsf{F}^t$ and $\mathsf{F}^{-1} = \frac{1}{N}\mathsf{F}^*$

FDAF

Transform vectors to frequency domain:

$$\begin{aligned} \mathsf{F} \cdot \underline{x}[k] = \underline{X}[k] &= (X_0[k], X_1[k], \cdots, X_{N-1}[k])^t \\ \mathsf{F}^{-1} \cdot \underline{w}[k] = \underline{W}[k] &= (W_0[k], W_1[k], \cdots, W_{N-1}[k])^t \end{aligned}$$

with DFT matrix F, which has properties: $\mathsf{F} = \mathsf{F}^t$ and $\mathsf{F}^{-1} = \frac{1}{N}\mathsf{F}^*$

Apply filter operation in frequency domain:

$$\hat{e}[k] = \sum_{i=0}^{N-1} x[k-i] \cdot w_i[k] = \underline{x}^t \cdot \underline{w}[k]$$

FDAF

Transform vectors to frequency domain:

$$\begin{aligned} \mathrm{F} \cdot \underline{\mathrm{x}}[k] = \underline{\mathrm{X}}[k] &= (X_0[k], X_1[k], \cdots, X_{N-1}[k])^t \\ \mathrm{F}^{-1} \cdot \underline{\mathrm{w}}[k] = \underline{\mathrm{W}}[k] &= (W_0[k], W_1[k], \cdots, W_{N-1}[k])^t \end{aligned}$$

with DFT matrix F, which has properties: $\mathrm{F} = \mathrm{F}^t$ and $\mathrm{F}^{-1} = \frac{1}{N}\mathrm{F}^*$

Apply filter operation in frequency domain:

$$\begin{aligned} \hat{e}[k] &= \sum_{i=0}^{N-1} x[k-i] \cdot w_i[k] = \underline{\mathrm{x}}^t \cdot \underline{\mathrm{w}}[k] \\ &= \underline{\mathrm{x}}^t[k](\mathrm{F} \cdot \mathrm{F}^{-1})\underline{\mathrm{w}}[k] = (\mathrm{F}\underline{\mathrm{x}}[k])^t \cdot \left(\mathrm{F}^{-1}\underline{\mathrm{w}}[k]\right) \end{aligned}$$

FDAF

Transform vectors to frequency domain:

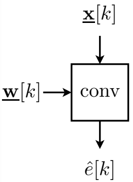
$$\begin{aligned} \mathsf{F} \cdot \underline{x}[k] = \underline{X}[k] &= (X_0[k], X_1[k], \cdots, X_{N-1}[k])^t \\ \mathsf{F}^{-1} \cdot \underline{w}[k] = \underline{W}[k] &= (W_0[k], W_1[k], \cdots, W_{N-1}[k])^t \end{aligned}$$

with DFT matrix F, which has properties: $\mathsf{F} = \mathsf{F}^t$ and $\mathsf{F}^{-1} = \frac{1}{N}\mathsf{F}^*$

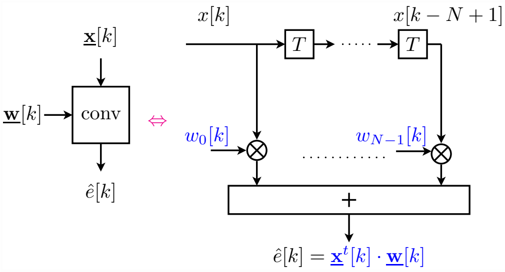
Apply filter operation in frequency domain:

$$\begin{aligned} \hat{e}[k] &= \sum_{i=0}^{N-1} x[k-i] \cdot w_i[k] = \underline{x}^t \cdot \underline{w}[k] \\ &= \underline{x}^t[k](\mathsf{F} \cdot \mathsf{F}^{-1})\underline{w}[k] = (\mathsf{F}\underline{x}[k])^t \cdot \left(\mathsf{F}^{-1}\underline{w}[k]\right) \\ &= \underline{X}^t[k] \cdot \underline{W}[k] = \sum_{l=0}^{N-1} X_l[k] W_l[k] \end{aligned}$$

FDAF

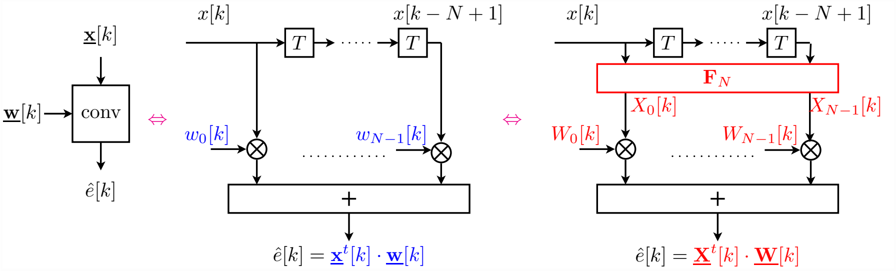


FDAF



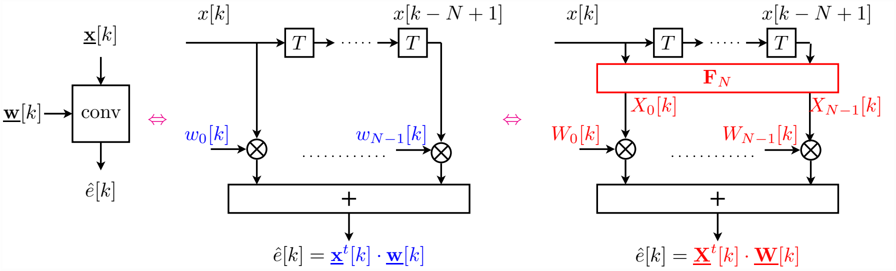
$\underline{\mathbf{x}}[k]$
$\underline{\mathbf{w}}[k]$
conv
$\hat{e}[k]$
$\Leftrightarrow$
$x[k]$
T
T
$x[k-N+1]$
$w_0[k]$
$w_{N-1}[k]$
$+$
$\hat{e}[k] = \underline{\mathbf{x}}^t[k] \cdot \underline{\mathbf{w}}[k]$

FDAF



$\underline{\mathbf{x}}[k]$
$\underline{\mathbf{w}}[k]$
conv
$\hat{e}[k]$
$\Leftrightarrow$
$x[k]$
$x[k-N+1]$
T
T
$w_0[k]$
$w_{N-1}[k]$
$+$
$\hat{e}[k] = \underline{\mathbf{x}}^t[k] \cdot \underline{\mathbf{w}}[k]$
$\Leftrightarrow$
$x[k]$
$x[k-N+1]$
T
T
$\mathbf{F}_N$
$X_0[k]$
$X_{N-1}[k]$
$W_0[k]$
$W_{N-1}[k]$
$+$
$\hat{e}[k] = \underline{\mathbf{X}}^t[k] \cdot \underline{\mathbf{W}}[k]$

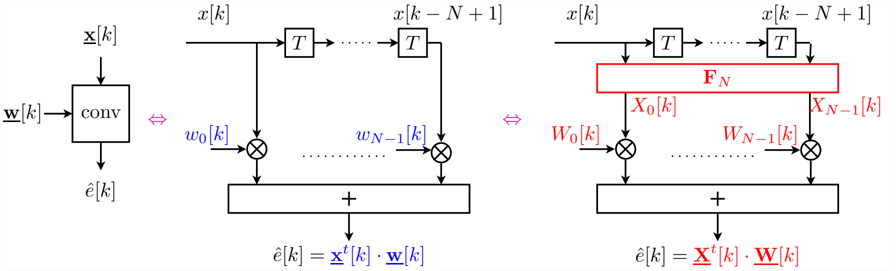
FDAF



Notes:

- Inverse DFT in definition of weights $W_l[k]$

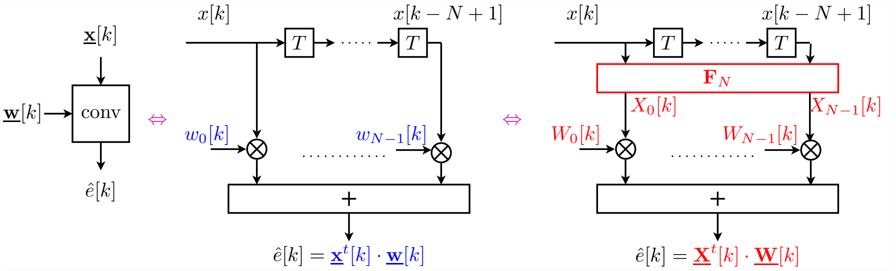
FDAF



Notes:

- Inverse DFT in definition of weights $W_l[k]$
- Use DFT symmetry to reduce complexity

FDAF



Notes:

- Inverse DFT in definition of weights $W_l[k]$
- Use DFT symmetry to reduce complexity
- For large N: Frequency bins "uncorrelated"

FDAF

Transform LMS to frequency domain (multiply update algorithm by F^{-1}*)*

FDAF

Transform LMS to frequency domain (multiply update algorithm by F^{-1}*)*

$$\mathsf{F}^{-1} \cdot \underline{\mathsf{w}}[k+1] \quad = \quad \mathsf{F}^{-1} \cdot \underline{\mathsf{w}}[k] + 2\alpha \mathsf{F}^{-1} \cdot \underline{\mathsf{x}}[k] r[k]$$

FDAF

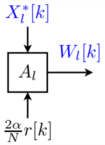
Transform LMS to frequency domain (multiply update algorithm by F^{-1}*)*

$$\begin{aligned} \mathsf{F}^{-1} \cdot \underline{\mathsf{w}}[k+1] &= \mathsf{F}^{-1} \cdot \underline{\mathsf{w}}[k] + 2\alpha \mathsf{F}^{-1} \cdot \underline{\mathsf{x}}[k] r[k] \\ \Leftrightarrow \quad \underline{\mathsf{W}}[k+1] &= \underline{\mathsf{W}}[k] + \frac{2\alpha}{N} \underline{\mathsf{X}}^*[k] r[k] \end{aligned}$$

FDAF

Transform LMS to frequency domain (multiply update algorithm by F^{-1}*)*

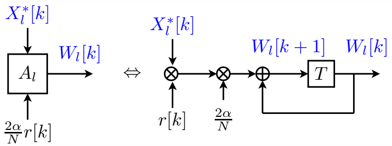
$$\begin{aligned} \mathsf{F}^{-1} \cdot \underline{\mathsf{w}}[k+1] &= \mathsf{F}^{-1} \cdot \underline{\mathsf{w}}[k] + 2\alpha \mathsf{F}^{-1} \cdot \underline{\mathsf{x}}[k] r[k] \\ \Leftrightarrow \quad \underline{\mathsf{W}}[k+1] &= \underline{\mathsf{W}}[k] + \frac{2\alpha}{N} \underline{\mathsf{X}}^{*}[k] r[k] \end{aligned}$$



FDAF

Transform LMS to frequency domain (multiply update algorithm by F^{-1}*)*

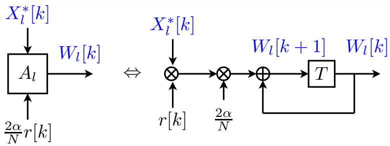
$$\begin{aligned} \mathsf{F}^{-1} \cdot \underline{w}[k+1] &= \mathsf{F}^{-1} \cdot \underline{w}[k] + 2\alpha \mathsf{F}^{-1} \cdot \underline{x}[k] r[k] \\ \Leftrightarrow \quad \underline{W}[k+1] &= \underline{W}[k] + \frac{2\alpha}{N} \underline{X}^*[k] r[k] \end{aligned}$$



FDAF

Transform LMS to frequency domain (multiply update algorithm by F^{-1}*)*

$$
\begin{aligned}
\mathsf{F}^{-1}\cdot \underline{w}[k+1] &= \mathsf{F}^{-1}\cdot \underline{w}[k] + 2\alpha \mathsf{F}^{-1}\cdot \underline{x}[k]r[k] \\
\Leftrightarrow \quad \underline{W}[k+1] &= \underline{W}[k] + \frac{2\alpha}{N}\underline{X}^*[k]r[k]
\end{aligned}
$$



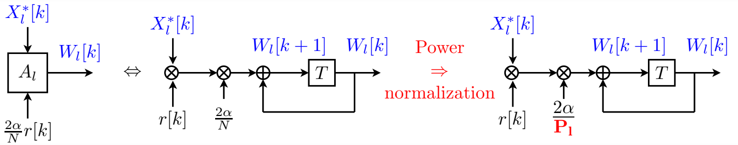
Improve convergence by Power normalization:

$$P_l = \frac{1}{N}E\{|X_l[k]|^2\}$$

FDAF

Transform LMS to frequency domain (multiply update algorithm by F^{-1}*)*

$$\begin{aligned}\mathsf{F}^{-1}\cdot\underline{w}[k+1] &= \mathsf{F}^{-1}\cdot\underline{w}[k]+2\alpha\mathsf{F}^{-1}\cdot\underline{x}[k]r[k]\\ \Leftrightarrow\quad \underline{W}[k+1] &= \underline{W}[k]+\frac{2\alpha}{N}\underline{X}^*[k]r[k]\end{aligned}$$



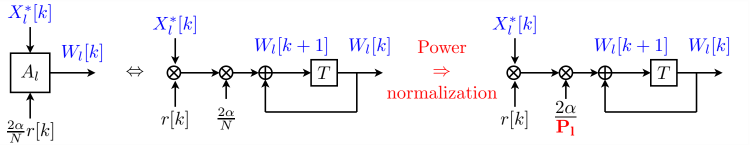
Improve convergence by Power normalization:

$$P_l=\frac{1}{N}E\{|X_l[k]|^2\}$$

FDAF

Transform LMS to frequency domain (multiply update algorithm by F^{-1})

$$
\begin{aligned}
\mathsf{F}^{-1} \cdot \underline{w}[k+1] &= \mathsf{F}^{-1} \cdot \underline{w}[k] + 2\alpha \mathsf{F}^{-1} \cdot \underline{x}[k] r[k] \\
\Leftrightarrow \quad \underline{W}[k+1] &= \underline{W}[k] + \frac{2\alpha}{N} \underline{X}^*[k] r[k]
\end{aligned}
$$



Improve convergence by Power normalization:

$$
P_l = \frac{1}{N} E\{|X_l[k]|^2\} \quad \text{e.g.:} \quad \hat{P}_l[k+1] = \beta \hat{P}_l[k] + (1-\beta) \frac{|X_l[k]|^2}{N}
$$

FDAF

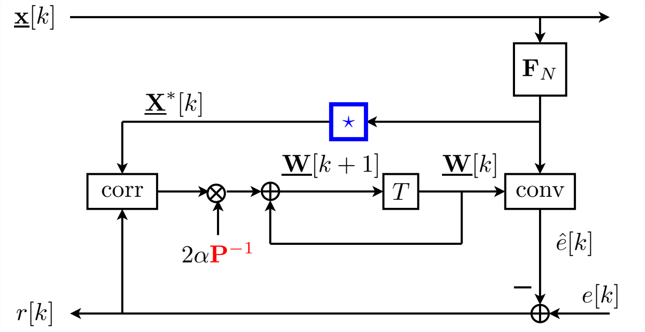
$$\underline{\textbf{FDAF algorithm}}\textbf{:} \qquad \underline{\mathrm{W}}[k+1] = \underline{\mathrm{W}}[k] + 2\alpha \mathrm{P}^{-1}\underline{\mathrm{X}}^{*}[k]r[k]$$

with $\mathrm{P} = \mathrm{diag}\{\underline{\mathrm{P}}\}$ and $(\underline{\mathrm{P}})_l = P_l = \frac{1}{N}E\{|X_l[k]|^2\}$

FDAF

FDAF algorithm: $\underline{\mathrm{W}}[k+1] = \underline{\mathrm{W}}[k] + 2\alpha \mathrm{P}^{-1} \underline{\mathrm{X}}^{*}[k] r[k]$

with $\mathrm{P} = \mathrm{diag}\{\underline{\mathrm{P}}\}$ and $(\underline{\mathrm{P}})_l = P_l = \frac{1}{N} E\{|X_l[k]|^2\}$



FDAF

Average behavior FDAF:

With $\underline{\mathsf{D}} = \mathsf{F}^{-1}\underline{\mathsf{d}} = \mathsf{F}^{-1}(\underline{\mathsf{w}} - \underline{\mathsf{w}}_o) = \underline{\mathsf{W}} - \underline{\mathsf{W}}_o$ FDAF becomes:

FDAF

Average behavior FDAF:

With $\underline{\mathsf{D}} = \mathsf{F}^{-1}\underline{\mathsf{d}} = \mathsf{F}^{-1}(\underline{\mathsf{w}} - \underline{\mathsf{w}}_o) = \underline{\mathsf{W}} - \underline{\mathsf{W}}_o$ FDAF becomes:

$$\underline{\mathsf{D}}[k+1] = \left(\mathsf{I} - \frac{2\alpha}{N}\mathsf{P}^{-1}\underline{\mathsf{X}}^*[k]\underline{\mathsf{X}}^t[k]\right)\underline{\mathsf{D}}[k] + \frac{2\alpha}{N}\mathsf{P}^{-1}\underline{\mathsf{X}}^*[k]r_{\mathrm{min}}[k]$$

FDAF

Average behavior FDAF:

With $\underline{D} = F^{-1}\underline{d} = F^{-1}(\underline{w} - \underline{w}_o) = \underline{W} - \underline{W}_o$ FDAF becomes:

$$\underline{D}[k+1] = \left(I - \frac{2\alpha}{N}P^{-1}\underline{X}^*[k]\underline{X}^t[k]\right)\underline{D}[k] + \frac{2\alpha}{N}P^{-1}\underline{X}^*[k]r_{\min}[k]$$

Different bins 'uncorrelated' $\Rightarrow \frac{E\{\underline{X}^*[k]\underline{X}^t[k]\}}{N} \approx P$

FDAF

Average behavior FDAF:

With $\underline{\mathrm{D}} = \mathrm{F}^{-1}\underline{\mathrm{d}} = \mathrm{F}^{-1}(\underline{\mathrm{w}} - \underline{\mathrm{w}}_o) = \underline{\mathrm{W}} - \underline{\mathrm{W}}_o$ FDAF becomes:

$$\underline{\mathrm{D}}[k+1] = \left(\mathrm{I} - \frac{2\alpha}{N}\mathrm{P}^{-1}\underline{\mathrm{X}}^*[k]\underline{\mathrm{X}}^t[k]\right)\underline{\mathrm{D}}[k] + \frac{2\alpha}{N}\mathrm{P}^{-1}\underline{\mathrm{X}}^*[k]r_{\mathrm{min}}[k]$$

Different bins 'uncorrelated' $\Rightarrow \frac{E\{\underline{\mathrm{X}}^*[k]\underline{\mathrm{X}}^t[k]\}}{N} \approx \mathrm{P}$

$\Rightarrow E\{\underline{\mathrm{D}}[k+1]\} \approx (1-2\alpha)\cdot E\{\underline{\mathrm{D}}[k]\} \Rightarrow \lim_{k\to\infty} E\{\underline{\mathrm{D}}[k]\} = \underline{0}$

FDAF

Average behavior FDAF:

With $\underline{\mathrm{D}} = \mathrm{F}^{-1}\underline{\mathrm{d}} = \mathrm{F}^{-1}(\underline{\mathrm{w}} - \underline{\mathrm{w}}_o) = \underline{\mathrm{W}} - \underline{\mathrm{W}}_o$ FDAF becomes:

$$\underline{\mathrm{D}}[k+1] \quad = \quad \left(\mathrm{I} - \frac{2\alpha}{N}\mathrm{P}^{-1}\underline{\mathrm{X}}^*[k]\underline{\mathrm{X}}^t[k]\right)\underline{\mathrm{D}}[k] + \frac{2\alpha}{N}\mathrm{P}^{-1}\underline{\mathrm{X}}^*[k]r_{\mathrm{min}}[k]$$

Different bins 'uncorrelated' $\Rightarrow \frac{E\{\underline{\mathrm{X}}^*[k]\underline{\mathrm{X}}^t[k]\}}{N} \approx \mathrm{P}$

$\Rightarrow E\{\underline{\mathrm{D}}[k+1]\} \approx (1-2\alpha)\cdot E\{\underline{\mathrm{D}}[k]\} \Rightarrow \lim_{k\to\infty} E\{\underline{\mathrm{D}}[k]\} = \underline{0}$

FDAF converges to Wiener solution: $\lim\limits_{k\to\infty} E\{\underline{\mathrm{W}}[k]\} = \underline{\mathrm{W}}_o = \mathrm{F}^{-1}\underline{\mathrm{w}}_o$

FDAF

Average behavior FDAF:

With $\underline{\mathrm{D}} = \mathrm{F}^{-1}\underline{\mathrm{d}} = \mathrm{F}^{-1}(\underline{\mathrm{w}} - \underline{\mathrm{w}}_o) = \underline{\mathrm{W}} - \underline{\mathrm{W}}_o$ FDAF becomes:

$$\underline{\mathrm{D}}[k+1] \quad = \quad \left(\mathrm{I} - \frac{2\alpha}{N}\mathrm{P}^{-1}\underline{\mathrm{X}}^*[k]\underline{\mathrm{X}}^t[k]\right)\underline{\mathrm{D}}[k] + \frac{2\alpha}{N}\mathrm{P}^{-1}\underline{\mathrm{X}}^*[k]r_{\mathrm{min}}[k]$$

Different bins 'uncorrelated' $\Rightarrow \frac{E\{\underline{\mathrm{X}}^*[k]\underline{\mathrm{X}}^t[k]\}}{N} \approx \mathrm{P}$

$\Rightarrow E\{\underline{\mathrm{D}}[k+1]\} \approx (1-2\alpha) \cdot E\{\underline{\mathrm{D}}[k]\} \Rightarrow \lim_{k\to\infty} E\{\underline{\mathrm{D}}[k]\} = \underline{0}$

FDAF converges to Wiener solution: $\displaystyle\lim_{k\to\infty} E\{\underline{\mathrm{W}}[k]\} = \underline{\mathrm{W}}_o = \mathrm{F}^{-1}\underline{\mathrm{w}}_o$

Notes:

- DFT (FFT) is fixed transform: Easy but not exact $\left(\frac{E\{\underline{\mathrm{X}}^*[k]\underline{\mathrm{X}}^t[k]\}}{N} \approx \mathrm{P}\right)$

FDAF

Average behavior FDAF:

With $\underline{\mathrm{D}} = \mathrm{F}^{-1}\underline{\mathrm{d}} = \mathrm{F}^{-1}(\underline{\mathrm{w}} - \underline{\mathrm{w}}_o) = \underline{\mathrm{W}} - \underline{\mathrm{W}}_o$ FDAF becomes:

$$\underline{\mathrm{D}}[k+1] \quad = \quad \left(\mathrm{I} - \frac{2\alpha}{N}\mathrm{P}^{-1}\underline{\mathrm{X}}^*[k]\underline{\mathrm{X}}^t[k]\right)\underline{\mathrm{D}}[k] + \frac{2\alpha}{N}\mathrm{P}^{-1}\underline{\mathrm{X}}^*[k]r_{\mathrm{min}}[k]$$

Different bins 'uncorrelated' $\Rightarrow \frac{E\{\underline{\mathrm{X}}^*[k]\underline{\mathrm{X}}^t[k]\}}{N} \approx \mathrm{P}$

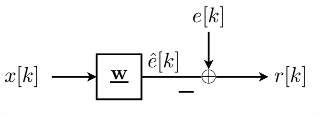
$\Rightarrow E\{\underline{\mathrm{D}}[k+1]\} \approx (1-2\alpha)\cdot E\{\underline{\mathrm{D}}[k]\} \Rightarrow \lim_{k\to\infty} E\{\underline{\mathrm{D}}[k]\} = \underline{0}$

FDAF converges to Wiener solution: $\lim\limits_{k\to\infty} E\{\underline{\mathrm{W}}[k]\} = \underline{\mathrm{W}}_o = \mathrm{F}^{-1}\underline{\mathrm{w}}_o$

Notes:

- DFT (FFT) is fixed transform: Easy but not exact ($\frac{E\{\underline{\mathrm{X}}^*[k]\underline{\mathrm{X}}^t[k]\}}{N} \approx \mathrm{P}$)
- FDAF equivalent to NLMS with white noise input

Summary



	MMSE	LS
Auto correlation	$R_x = E\{\underline{x}[k] \cdot \underline{x}^t[k]\}$	$\overline{R}_x = X^t \cdot X$
Cross correlation	$\underline{r}_{ex} = E\{e[k] \cdot \underline{x}[k]\}$	$\underline{\overline{r}}_{ex} = X^t \cdot \underline{e}$
Error J	$E\{r^2[k]\}$	$\sum_{i=0}^{L-1} r^2[k-i]$
Criterion	$\min_{\underline{w}}\{E\{r^2[k]\}\}$	$\min_{\underline{w}} \lvert\underline{e} - X \cdot \underline{w}\rvert^2$
Opt. solution $\underline{w}_o$	$R_x^{-1} \cdot \underline{r}_{ex}$	$\overline{R}_x^{-1} \cdot \underline{\overline{r}}_{ex}$
Min. error J_{min}	$E\{e^2\} - \underline{r}_{ex}^t R_x^{-1} \underline{r}_{ex}$	$\underline{e}^t \underline{e} - \underline{\overline{r}}_{ex}^t \overline{R}_x^{-1} \underline{\overline{r}}_{ex}$

Summary

Set of constraints: $\mathsf{C}^t \cdot \underline{\mathsf{w}} = \underline{\mathsf{f}}$

Solution for $N \geq M$: $\underline{\mathsf{w}}^c = \mathsf{C}(\mathsf{C}^t\mathsf{C})^{-1}\underline{\mathsf{f}}$

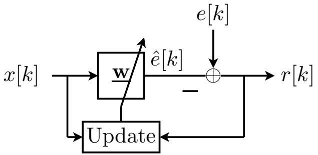
Solution for $N > M$ with MMSE:

$$\underline{\mathsf{w}}_o^c = \underline{\mathsf{w}}_o + \mathsf{R}_x^{-1}\mathsf{C}(\mathsf{C}^t\mathsf{R}_x^{-1}\mathsf{C})^{-1}(\underline{\mathsf{f}} - \mathsf{C}^t\underline{\mathsf{w}}_o)$$

Similar result:

$$\underline{\mathsf{w}}_o^c = \mathsf{R}_x^{-1}\mathsf{C}(\mathsf{C}^t\mathsf{R}_x^{-1}\mathsf{C})^{-1}\underline{\mathsf{f}}$$

Summary



Simple adaptive algorithms (no decorrelation):

$$\text{SGD} \quad : \quad \underline{w}[k+1] = \underline{w}[k] + 2\alpha(\underline{r}_{ex} - R_x\underline{w}[k])$$

$$\text{(complex)(N)LMS} \quad : \quad \underline{w}[k+1] = \underline{w}[k] + \frac{2\alpha}{\hat{\sigma}_x^2}\underline{x}[k]r^*[k]$$

Constrained LMS: $C^t \cdot \underline{w} = \underline{f}$

$$\underline{w}[k+1] = \tilde{P} \cdot \{\underline{w}[k] + 2\alpha\underline{x}[k]r[k]\} + C\left(C^t \cdot C\right)^{-1}\underline{f}$$

with

$$\tilde{P} = I - C\left(C^t \cdot C\right)^{-1} C^t \text{ and } \underline{w}[0] = C\left(C^t \cdot C\right)^{-1}\underline{f}$$

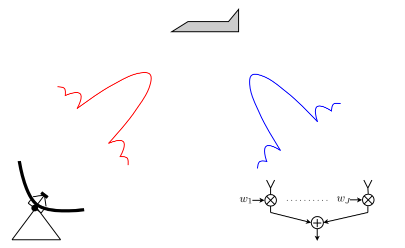
Summary

<u>*Algorithms with improved convergence:*</u>

$$\begin{aligned}
\text{LMS/Newton} &: \quad \underline{w}[k+1] = \underline{w}[k] + 2\alpha R_x^{-1}\underline{x}[k]r[k] \\
\text{Newton} &: \quad \underline{w}[k+1] = \underline{w}[k] + 2\alpha R_x^{-1} \cdot (\underline{r}_{ex} - R_x\underline{w}[k]) \\
\text{RLS} &: \quad \underline{g}[k+1] = \frac{\overline{R}_x^{-1}[k]\underline{x}[k+1]}{\gamma^2 + \underline{x}^t[k+1]\overline{R}_x^{-1}[k]\underline{x}[k+1]} \\
&\quad\ \overline{R}_x^{-1}[k+1] = \gamma^{-2}\left(\overline{R}_x^{-1}[k] - \underline{g}[k+1] \cdot \underline{x}^t[k+1]\overline{R}_x^{-1}[k]\right. \\
&\quad\ \underline{\overline{r}}_{ex}[k+1] = \gamma^2\underline{\overline{r}}_{ex}[k] + \underline{x}^t[k+1] \cdot e[k+1] \\
&\quad\ \underline{w}[k+1] = \overline{R}_x^{-1}[k+1] \cdot \underline{r}_{ex}[k+1]
\end{aligned}$$

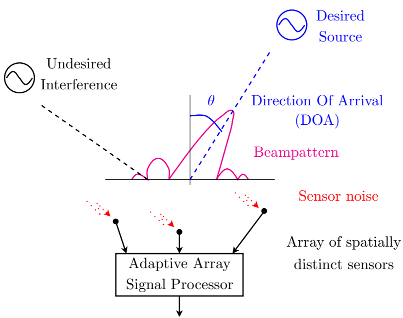
Array Signal Processing (Part IB)

Introduction

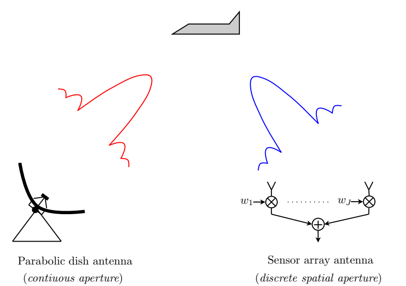


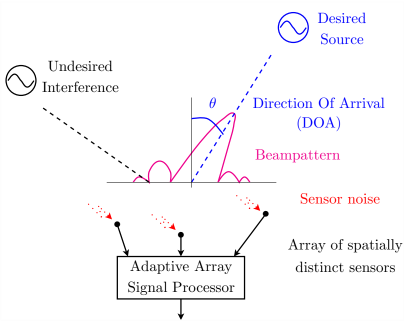
Parabolic dish antenna
(*contiuous aperture*)

Sensor array antenna
(*discrete spatial aperture*)



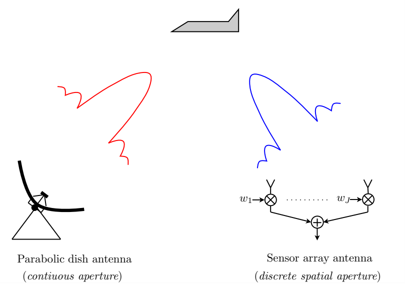
Introduction

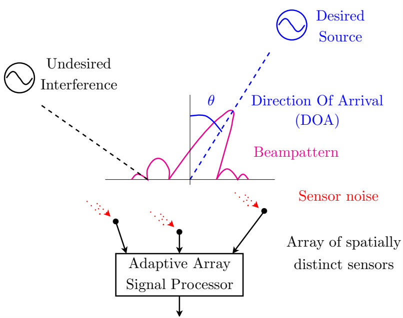




Beamforming: Spatio temporal filtering to either direct of block the radiation or reception of signals in specified directions

Introduction





Beamforming: Spatio temporal filtering to either direct of block the radiation or reception of signals in specified directions

Result Beamforming = **Spatial filtering**: Separate signals with possible overlapping frequencies but from different directions

Different scenario's (content first part):

- Bandwidth source
- Array geometry
- Far field vs. near field
- Direction Of Arrival (DOA)
- Discrete-time signal representation
- Array signal model
- ASP unit
- Spatial/ temporal filtering
- Broadband signals

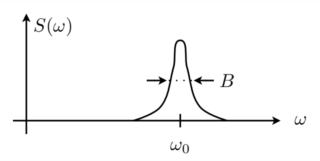
Scenario: Bandwidth source

Analytical representation: $s(t) = A(t)e^{j(\omega_o t + \phi(t))}$

Scenario: Bandwidth source

Analytical representation: $s(t) = A(t)e^{j(\omega_o t + \phi(t))}$

Narrowband: $A(t)$ and $\phi(t)$ vary slower than $e^{j\omega_0 t}$



Narrowband: $|\tau| \ll 1/B$

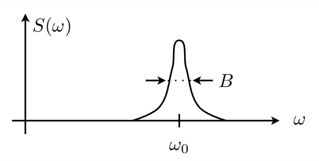
$A(t-\tau) \approx A(t) = 1$ (usually) $\Rightarrow$

$\phi(t-\tau) \approx \phi(t) = 0$ (usually)

Scenario: Bandwidth source

Analytical representation: $s(t) = A(t)e^{j(\omega_o t + \phi(t))}$

Narrowband: $A(t)$ and $\phi(t)$ vary slower than $e^{j\omega_0 t}$



Narrowband: $|\tau| \ll 1/B$

$A(t-\tau) \approx A(t) = 1$ (usually) $\Rightarrow$

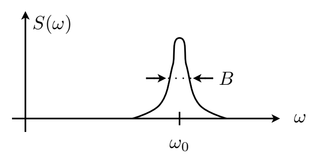
$\phi(t-\tau) \approx \phi(t) = 0$ (usually)

$\Rightarrow s(t-\tau) = A(t-\tau)e^{j\phi(t-\tau)}e^{j\omega_0(t-\tau)} \approx e^{-j\omega_o\tau} \cdot s(t)$

Scenario: Bandwidth source

Analytical representation: $s(t) = A(t)e^{j(\omega_o t + \phi(t))}$

Narrowband: $A(t)$ and $\phi(t)$ vary slower than $e^{j\omega_0 t}$



Narrowband: $|\tau| \ll 1/B$

$A(t-\tau) \approx A(t) = 1$ (usually) $\Rightarrow$

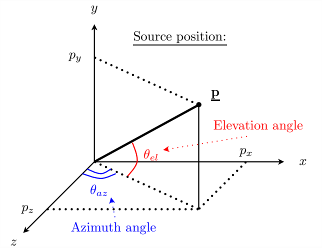
$\phi(t-\tau) \approx \phi(t) = 0$ (usually)

$$\Rightarrow s(t-\tau) = A(t-\tau)e^{j\phi(t-\tau)}e^{j\omega_0(t-\tau)} \approx e^{-j\omega_o\tau} \cdot s(t)$$

Thus for narrowband: Time delay $\equiv$ phase shift

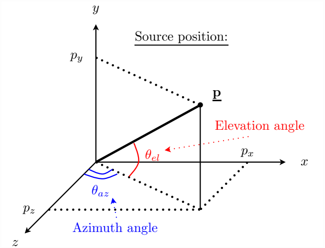
In this course mainly narrowband

Scenario: Array geometry



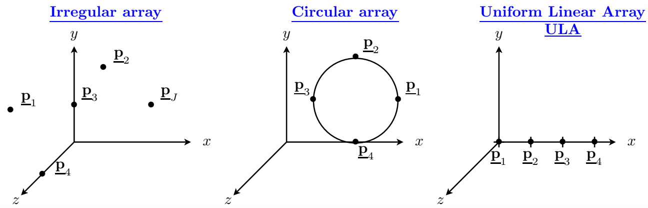
$$
\begin{aligned}
\underline{p} &= (p_x, p_y, p_z)^t \\
p_x &= ||\underline{p}|| \sin(\theta_{az}) \cos(\theta_{el}) \\
p_y &= ||\underline{p}|| \sin(\theta_{el}) \\
p_z &= ||\underline{p}|| \cos(\theta_{az}) \cos(\theta_{el})
\end{aligned}
$$

Scenario: Array geometry



$$
\begin{aligned}
\underline{p} &= (p_x, p_y, p_z)^t \\
p_x &= ||\underline{p}|| \sin(\theta_{az}) \cos(\theta_{el}) \\
p_y &= ||\underline{p}|| \sin(\theta_{el}) \\
p_z &= ||\underline{p}|| \cos(\theta_{az}) \cos(\theta_{el})
\end{aligned}
$$

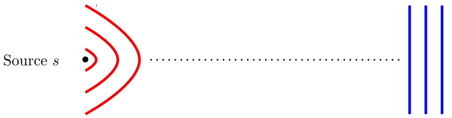
Array can be uniform, nonuniform, linear, circular, $\cdots$



In this course mainly: **ULA**

Scenario: Far field vs. near field

Array aperture: Volume (1D length) that collects incoming signal



Near field

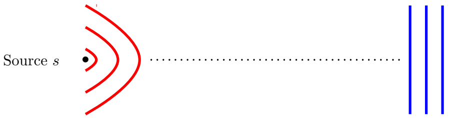
- Distance source - array $\ll$ Array aperture
- Sherical wavefront

Far field

- Distance source - array $\gg$ Array aperture
- Plane wavefront

Scenario: Far field vs. near field

Array aperture: Volume (1D length) that collects incoming signal



- Distance source - array $\ll$ Array aperture
- Sherical wavefront

- Distance source - array $\gg$ Array aperture
- Plane wavefront

Near field: Propagation for single frequency source

$$s(t, \underline{p}) = \frac{A}{||\underline{p}||^2} e^{j\omega(t - \frac{||\underline{p}||}{c})} \text{ with } \omega = 2\pi f \text{ and } f = \frac{c}{\lambda}$$

λ = wavelength, c = speed in medium (≈ 334 [m/sec] for sound in air)

$\Rightarrow$ Amplitude decays proportional to distance from source

Scenario: Far field vs. near field

In this course mainly far field:

$$s(t, \underline{p}_i) = A e^{j\omega(t-\tau_i)} = A e^{j\omega(t-\frac{\underline{v}^t \cdot \underline{p}_i}{c})} = A e^{j(\omega t - \underline{k}^t \cdot \underline{p}_i)}$$

with direction vector $\underline{v}$, wave number vector $\underline{k} = \frac{\omega}{c} \cdot \underline{v}$

Scenario: Far field vs. near field

In this course mainly far field:

$$s(t, \underline{p}_i) = Ae^{j\omega(t-\tau_i)} = Ae^{j\omega(t-\frac{\underline{v}^t \cdot \underline{p}_i}{c})} = Ae^{j(\omega t - \underline{k}^t \cdot \underline{p}_i)}$$

with direction vector $\underline{v}$, wave number vector $\underline{k} = \frac{\omega}{c} \cdot \underline{v}$

- ✓ $s(t, \underline{p}_i)$ describes propagation as function of both time and space
- ✓ Information is preserved while propagating

Scenario: Far field vs. near field

In this course mainly far field:

$$s(t, \underline{p}_i) = Ae^{j\omega(t-\tau_i)} = Ae^{j\omega(t-\frac{\underline{v}^t \cdot \underline{p}_i}{c})} = Ae^{j(\omega t - \underline{k}^t \cdot \underline{p}_i)}$$

with direction vector $\underline{v}$, wave number vector $\underline{k} = \frac{\omega}{c} \cdot \underline{v}$

- ✓ $s(t, \underline{p}_i)$ describes propagation as function of both time and space
- ✓ Information is preserved while propagating

⇒ Reconstruction band limited signal over all space and time by either:

- Temporally sampling at given location in space
- Spatially sampling at given instant of time
- **Combination**

Scenario: Far field vs. near field

In this course mainly far field:

$$s(t, \underline{p}_i) = A e^{j\omega(t-\tau_i)} = A e^{j\omega(t-\frac{\underline{v}^t \cdot \underline{p}_i}{c})} = A e^{j(\omega t - \underline{k}^t \cdot \underline{p}_i)}$$

with direction vector $\underline{v}$, wave number vector $\underline{k} = \frac{\omega}{c} \cdot \underline{v}$

- ✓ $s(t, \underline{p}_i)$ describes propagation as function of both time and space
- ✓ Information is preserved while propagating

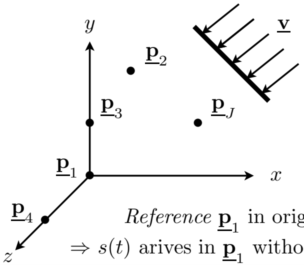
$\Rightarrow$ Reconstruction band limited signal over all space and time by either:

- ▶ Temporally sampling at given location in space
- ▶ Spatially sampling at given instant of time
- ▶ **Combination**

Spatially sampling:

$\boxed{\text{Basis for all aperture and sensor array processing techniques}}$

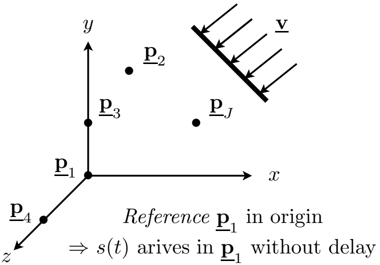
Scenario: Far field vs. near field



Reference $\underline{\mathbf{p}}_1$ in origin

$\Rightarrow s(t)$ arives in $\underline{\mathbf{p}}_1$ without delay

Scenario: Far field vs. near field



At position $\underline{p}_i$: $s(t - \tau_i) = s(t)e^{-j\omega\tau_i}$

with delay $\tau_i = \frac{\underline{v}^t \cdot \underline{p}_i}{c}$ and $\underline{v}$ is direction vector

$\omega = 2\pi f$, $f = \frac{c}{\lambda}$, $\lambda =$ wavelength, $c =$ speed in medium

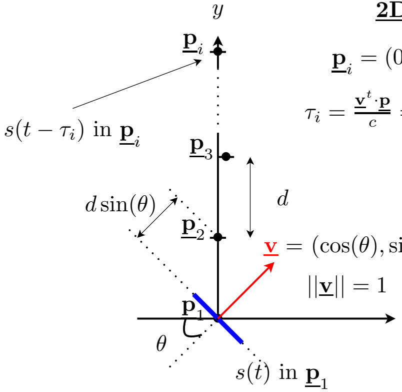
Scenario: Direction Of Arrival (DOA)

Location is 3D quantity. In practice often 2D

Scenario: Direction Of Arrival (DOA)

Location is 3D quantity. In practice often 2D

Example: Narrow band, far field 2D DOA for ULA:



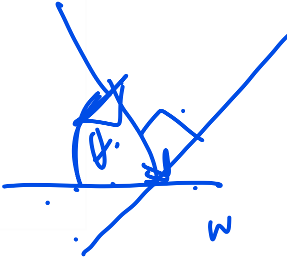
2D ULA:

$$\underline{\mathbf{p}}_i = (0, (i-1)\cdot d)^t$$

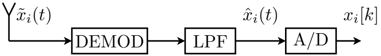
$$\tau_i = \frac{\underline{\mathbf{v}}^t \cdot \underline{\mathbf{p}}}{c} = (i-1)\frac{d\sin(\theta)}{c}$$

$$\underline{\mathbf{v}} = (\cos(\theta), \sin(\theta))^t$$

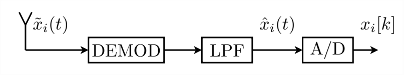
$$||\underline{\mathbf{v}}|| = 1$$



Scenario: Discrete-time signal representation



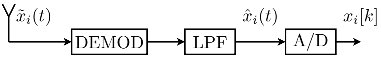
Scenario: Discrete-time signal representation



- ✓ Analog sensor signal at sensor i: $\tilde{x}_i(t)$
- ✓ Ideal demodulation and LPF results in baseband signal: $\hat{x}_i(t)$
- ✓ After A/D (complex valued) discrete-time signal: $x_i[k]$
- ✓ Analog signal at $\underline{p}_i$ for narrow band, far field case:

$$\hat{x}_i(t) = s(t - \tau_i) = s(t)e^{-j\omega\tau_i} \quad \text{with} \quad \tau_i = \frac{\underline{v}^t \cdot \underline{p}_i}{c}$$

Scenario: Discrete-time signal representation



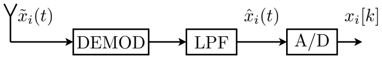
- ✓ Analog sensor signal at sensor i: $\tilde{x}_i(t)$
- ✓ Ideal demodulation and LPF results in baseband signal: $\hat{x}_i(t)$
- ✓ After A/D (complex valued) discrete-time signal: $x_i[k]$
- ✓ Analog signal at $\underline{p}_i$ for narrow band, far field case:

$$\hat{x}_i(t) = s(t - \tau_i) = s(t)\mathrm{e}^{-\mathrm{j}\omega\tau_i} \quad \text{with} \quad \tau_i = \frac{\underline{v}^t \cdot \underline{p}_i}{c}$$

- ✓ Discrete-time signal at $\underline{p}_i$ for ULA-case:

$$s[k]\mathrm{e}^{-\mathrm{j}\omega\tau_i} = s[k]\cdot\mathrm{e}^{-\mathrm{j}2\pi(i-1)\frac{d\sin(\theta)}{\lambda}} = s[k]\cdot a_i(\theta) \text{ with } a_i(\theta) = \mathrm{e}^{-\mathrm{j}2\pi(i-1)\frac{d\sin(\theta}{\lambda}}$$

Scenario: Discrete-time signal representation



- ✓ Analog sensor signal at sensor i: $\tilde{x}_i(t)$
- ✓ Ideal demodulation and LPF results in baseband signal: $\hat{x}_i(t)$
- ✓ After A/D (complex valued) discrete-time signal: $x_i[k]$
- ✓ Analog signal at $\underline{p}_i$ for narrow band, far field case:

$$\hat{x}_i(t) = s(t - \tau_i) = s(t)\mathrm{e}^{-\mathrm{j}\omega\tau_i} \quad \text{with} \quad \tau_i = \frac{\underline{v}^t \cdot \underline{p}_i}{c}$$

- ✓ Discrete-time signal at $\underline{p}_i$ for ULA-case:

$$s[k]\mathrm{e}^{-\mathrm{j}\omega\tau_i} = s[k]\cdot\mathrm{e}^{-\mathrm{j}2\pi(i-1)\frac{d\sin(\theta)}{\lambda}} = s[k]\cdot a_i(\theta) \text{ with } a_i(\theta) = \mathrm{e}^{-\mathrm{j}2\pi(i-1)\frac{d\sin(\theta}{\lambda}}$$

Note: In fact $a_i(\theta)$ also depends on ω

Scenario: Array signal model

$$\begin{aligned} \text{Array sensor vector} &: \quad \underline{\mathrm{x}}[k] = (x_1[k], x_2[k], \cdots x_J[k])^t \\ \text{Noise vector} &: \quad \underline{\mathrm{n}}[k] = (n_1[k], n_2[k], \cdots n_J[k])^t \\ \text{Steering vector} &: \quad \underline{\mathrm{a}}[k] = (a_1(\theta), a_2(\theta), \cdots a_J(\theta))^t \\ & \quad\ \text{with } a_i(\theta) = \mathrm{e}^{-\mathrm{j}\omega\tau_i(\theta)} \end{aligned}$$

Scenario: Array signal model

$$
\begin{aligned}
\text{Array sensor vector} \quad &: \quad \underline{\mathrm{x}}[k] = (x_1[k], x_2[k], \cdots x_J[k])^t \\
\text{Noise vector} \quad &: \quad \underline{\mathrm{n}}[k] = (n_1[k], n_2[k], \cdots n_J[k])^t \\
\text{Steering vector} \quad &: \quad \underline{\mathrm{a}}[k] = (a_1(\theta), a_2(\theta), \cdots a_J(\theta))^t \\
& \quad\;\; \text{with } a_i(\theta) = \mathrm{e}^{-\mathrm{j}\omega\tau_i(\theta)}
\end{aligned}
$$

Case: Noise observation, P sources, J sensors

Scenario: Array signal model

$$
\begin{aligned}
\text{Array sensor vector} &: \quad \underline{\mathrm{x}}[k] = (x_1[k], x_2[k], \cdots x_J[k])^t \\
\text{Noise vector} &: \quad \underline{\mathrm{n}}[k] = (n_1[k], n_2[k], \cdots n_J[k])^t \\
\text{Steering vector} &: \quad \underline{\mathrm{a}}[k] = (a_1(\theta), a_2(\theta), \cdots a_J(\theta))^t \\
& \quad\ \text{with } a_i(\theta) = \mathrm{e}^{-\mathrm{j}\omega\tau_i(\theta)}
\end{aligned}
$$

Case: Noise observation, P sources, J sensors

$$x_i[k] = \sum_{p=1}^{P} a_i(\theta_p) s_p[k] + n_i[k]$$

Scenario: Array signal model

$$
\begin{aligned}
\text{Array sensor vector} \quad &: \quad \underline{x}[k] = (x_1[k], x_2[k], \cdots x_J[k])^t \\
\text{Noise vector} \quad &: \quad \underline{n}[k] = (n_1[k], n_2[k], \cdots n_J[k])^t \\
\text{Steering vector} \quad &: \quad \underline{a}[k] = (a_1(\theta), a_2(\theta), \cdots a_J(\theta))^t \\
& \quad\ \text{with } a_i(\theta) = \mathrm{e}^{-\mathrm{j}\omega\tau_i(\theta)}
\end{aligned}
$$

Case: Noise observation, P sources, J sensors

$$
x_i[k] = \sum_{p=1}^{P} a_i(\theta_p) s_p[k] + n_i[k] \quad \Leftrightarrow \quad \boxed{\underline{x}[k] = \mathrm{A} \cdot \underline{s}[k] + \underline{n}[k]}
$$

Scenario: Array signal model

$$\begin{aligned}
\text{Array sensor vector} \quad &: \quad \underline{x}[k] = (x_1[k], x_2[k], \cdots x_J[k])^t \\
\text{Noise vector} \quad &: \quad \underline{n}[k] = (n_1[k], n_2[k], \cdots n_J[k])^t \\
\text{Steering vector} \quad &: \quad \underline{a}[k] = (a_1(\theta), a_2(\theta), \cdots a_J(\theta))^t \\
& \quad \text{with } a_i(\theta) = \mathrm{e}^{-\mathrm{j}\omega\tau_i(\theta)}
\end{aligned}$$

Case: Noise observation, P sources, J sensors

$$x_i[k] = \sum_{p=1}^{P} a_i(\theta_p) s_p[k] + n_i[k] \quad \Leftrightarrow \quad \boxed{\underline{x}[k] = \mathrm{A} \cdot \underline{s}[k] + \underline{n}[k]}$$

$$\begin{aligned}
J \times P \text{ steering matrix A} \quad &= \quad (\underline{a}(\theta_1), \underline{a}(\theta_2), \cdots, \underline{a}(\theta_P)) \\
P \times 1 \text{ signal vector } \underline{s}[k] \quad &= \quad (s_1[k], s_2[k], \cdots, s_P[k])^t
\end{aligned}$$

Scenario: Array signal model

$$\begin{aligned}
\text{Array sensor vector} &: \quad \underline{x}[k] = (x_1[k], x_2[k], \cdots x_J[k])^t \\
\text{Noise vector} &: \quad \underline{n}[k] = (n_1[k], n_2[k], \cdots n_J[k])^t \\
\text{Steering vector} &: \quad \underline{a}[k] = (a_1(\theta), a_2(\theta), \cdots a_J(\theta))^t \\
& \quad\ \text{with } a_i(\theta) = \mathrm{e}^{-\mathrm{j}\omega\tau_i(\theta)}
\end{aligned}$$

Case: Noise observation, P sources, J sensors

$$x_i[k] = \sum_{p=1}^{P} a_i(\theta_p) s_p[k] + n_i[k] \quad \Leftrightarrow \quad \boxed{\underline{x}[k] = \mathrm{A} \cdot \underline{s}[k] + \underline{n}[k]}$$

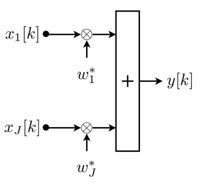
$$\begin{aligned}
J \times P \text{ steering matrix A} &= (\underline{a}(\theta_1), \underline{a}(\theta_2), \cdots, \underline{a}(\theta_P)) \\
P \times 1 \text{ signal vector } \underline{s}[k] &= (s_1[k], s_2[k], \cdots, s_P[k])^t
\end{aligned}$$

Covariance structure: $\boxed{\mathrm{R}_x = E\{\underline{x} \cdot \underline{x}^h\} = \mathrm{A}\mathrm{R}_s\mathrm{A}^h + \mathrm{R}_n}$

with $\mathrm{R}_s = E\{\underline{s} \cdot \underline{s}^h\}$ and $\mathrm{R}_n = E\{\underline{n} \cdot \underline{n}^h\} = \sigma_n^2 \mathbf{I}$

Scenario: ASP unit

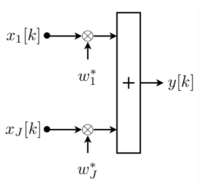
Case: Single complex weight for each sensor



$$
\begin{aligned}
y[k] &= \sum_{i=1}^{J} w_i^* x_i[k] = \underline{w}^h \cdot \underline{x}[k] \\
\underline{x}[k] &= (x_1[k], \cdots, x_J[k])t \\
\underline{w} &= (w_1, \cdots, w_J)^t
\end{aligned}
$$

Scenario: ASP unit

Case: Single complex weight for each sensor

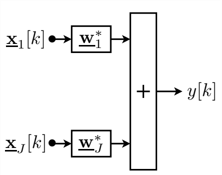


$$y[k] = \sum_{i=1}^{J} w_i^* x_i[k] = \underline{w}^h \cdot \underline{x}[k]$$

$$\underline{x}[k] = (x_1[k], \cdots, x_J[k])t$$

$$\underline{w} = (w_1, \cdots, w_J)^t$$

Case: FIR filter for each sensor



$\underline{\mathbf{w}}_i$: FIR filter with N weights

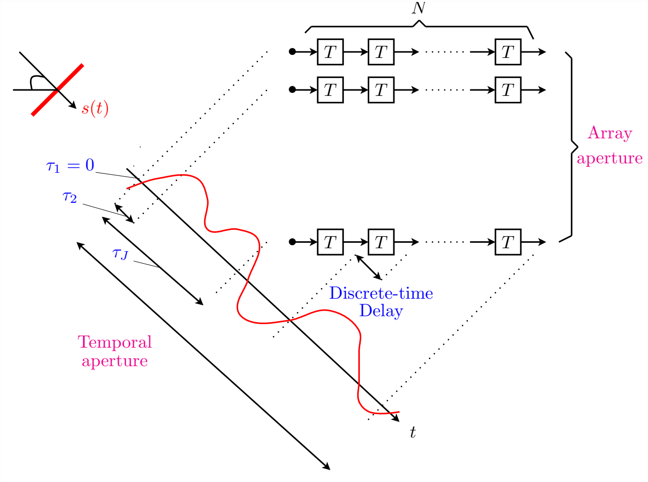
$$y[k] = \sum_{i=1}^{J} \underline{w}_i^h \cdot \underline{x}_i[k] = \underline{w}^h \cdot \underline{x}[k]$$

$$\underline{x}[k] = (\underline{x}_1[k], \cdots, \underline{x}_J[k])t$$

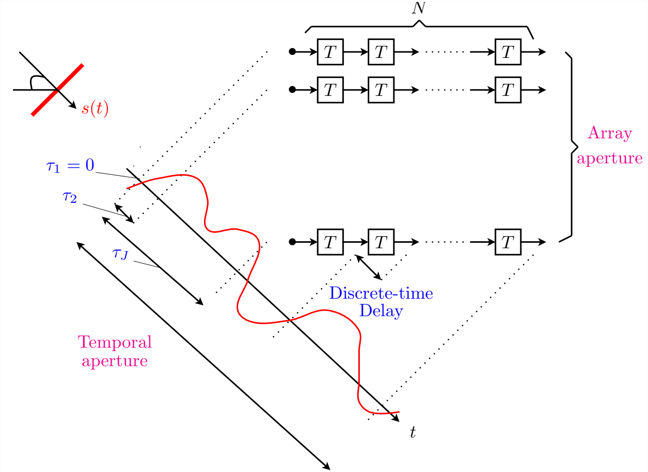
$$\underline{w} = (\underline{w}_1, \cdots, \underline{w}_J)^t$$

$$\underline{w}_i = (w_{i,1}, \cdots, w_{i,N})^t$$

Scenario: Spatial/ temporal filtering

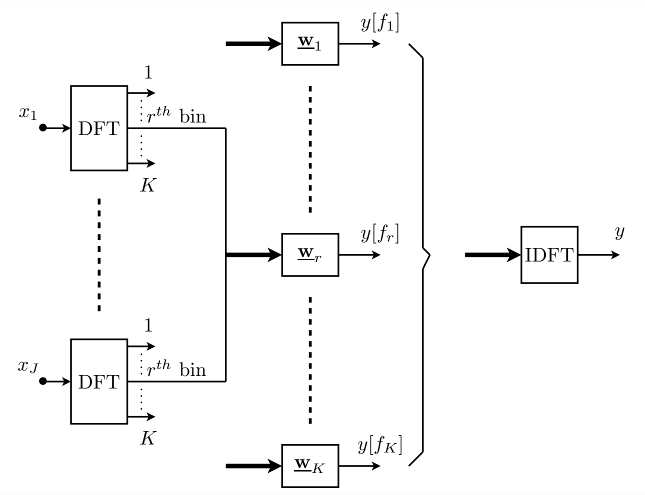


Scenario: Spatial/ temporal filtering



Note: FIR filtering effect both temporal and spatial response

Scenario: Broadband signals



ULA Beampattern

Main properties

ULA Beampattern

Assumptions:

ULA Beampattern

Assumptions:

- Single source $s(t) = e^{j\omega t}$
- Frequency relations: $\omega = 2\pi f = 2\pi \frac{c}{\lambda}$, with wavenumber λ and speed of propagation c (≈ 343 [m/sec])
- Direction Of Arrival (DOA): θ
- Far field, thus plane wavefront
- ULA with distance d [m] between sensors
- J omnidirectional sensors $\Rightarrow$ array aperture $= J \cdot d$ [m]

ULA Beampattern

Assumptions:

- Single source $s(t) = \mathrm{e}^{\mathrm{j}\omega t}$
- Frequency relations: $\omega = 2\pi f = 2\pi \frac{c}{\lambda}$, with wavenumber λ and speed of propagation c (≈ 343 [m/sec])
- Direction Of Arrival (DOA): θ
- Far field, thus plane wavefront
- ULA with distance d [m] between sensors
- J omnidirectional sensors $\Rightarrow$ array aperture $= J \cdot d$ [m]
- Model:$\underline{x}[k] = \underline{a}(\theta) \cdot s[k]$ (No noise, no interference)
- ASP unit: Single complex weight fw_i or each sensor

$$\Rightarrow \quad y[k] = \sum_{i=1}^{J} w_i^* x_i[k] = \underline{w}^h \cdot \underline{x}[k] = \underline{w}^h \cdot \underline{a}(\theta) \cdot s[k]$$

with $(\underline{a}(\theta))_i = \mathrm{e}^{-\mathrm{j}2\pi(i-1)\frac{d\sin(\theta)}{\lambda}}$

ULA Beampattern

Array response : $r(\theta) = \underline{w}^h \cdot \underline{a}(\theta)$

Other names: Angular response or directivity pattern

ULA Beampattern

Array response : $r(\theta) = \underline{w}^h \cdot \underline{a}(\theta)$

Other names: Angular response or directivity pattern

Beampattern : $B(\theta) = \frac{1}{J^2}|r(\theta)|^2 = \frac{1}{J^2}|\underline{w}^h \cdot \underline{a}(\theta)|^2$

ULA Beampattern

Array response : $r(\theta) = \underline{w}^h \cdot \underline{a}(\theta)$

Other names: Angular response or directivity pattern

Beampattern : $B(\theta) = \frac{1}{J^2}|r(\theta)|^2 = \frac{1}{J^2}|\underline{w}^h \cdot \underline{a}(\theta)|^2$

Notes:

- Array response: Response to unit-amplitude plane wave front from direction θ
- Non ideal sensor characteristics can be incorporated
- Weights effect both temporal and spatial response
- Vector space interpretation: Angle between $\underline{w}$ and $\underline{a}$
- To evaluate beampattern: Choose all weight equal to one, thus $\underline{w} = (1, 1, \cdots, 1)^t$

ULA Beampattern: The equation

$$B(\theta) \quad = \quad \frac{1}{J^2}|\underline{1}^t \cdot \underline{a}(\theta)|^2$$

ULA Beampattern: The equation

$$B(\theta) \;=\; \frac{1}{J^2}\left|\underline{1}^t \cdot \underline{a}(\theta)\right|^2 = \frac{1}{J^2}\left|\sum_{i=1}^{J} e^{-j2\pi(i-1)\frac{d}{\lambda}\sin(\theta)}\right|^2$$

ULA Beampattern: The equation

$$
\begin{aligned}
B(\theta) &= \frac{1}{J^2}\left|\underline{1}^t \cdot \underline{a}(\theta)\right|^2 = \frac{1}{J^2}\left|\sum_{i=1}^{J} \mathrm{e}^{-\mathrm{j}2\pi(i-1)\frac{d}{\lambda}\sin(\theta)}\right|^2 \\
&= \frac{1}{J^2}\left|\frac{1-\mathrm{e}^{-\mathrm{j}J2\pi\frac{d}{\lambda}\sin(\theta)}}{1-\mathrm{e}^{-\mathrm{j}2\pi\frac{d}{\lambda}\sin(\theta)}}\right|^2
\end{aligned}
$$

ULA Beampattern: The equation

$$\begin{aligned}
B(\theta) &= \frac{1}{J^2}\left|\underline{1}^t \cdot \underline{a}(\theta)\right|^2 = \frac{1}{J^2}\left|\sum_{i=1}^{J} e^{-j2\pi(i-1)\frac{d}{\lambda}\sin(\theta)}\right|^2 \\
&= \frac{1}{J^2}\left|\frac{1-e^{-jJ2\pi\frac{d}{\lambda}\sin(\theta)}}{1-e^{-j2\pi\frac{d}{\lambda}\sin(\theta)}}\right|^2 = \boxed{\frac{1}{J^2}\left|\frac{\sin(J\pi\frac{d}{\lambda}\sin(\theta))}{\sin(\pi\frac{d}{\lambda}\sin(\theta))}\right|^2}
\end{aligned}$$

ULA Beampattern: The equation

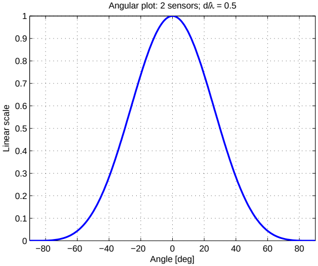
$$
\begin{aligned}
B(\theta) &= \frac{1}{J^2}\left|\underline{1}^t \cdot \underline{a}(\theta)\right|^2 = \frac{1}{J^2}\left|\sum_{i=1}^{J} e^{-\mathrm{j}2\pi(i-1)\frac{d}{\lambda}\sin(\theta)}\right|^2 \\
&= \frac{1}{J^2}\left|\frac{1-e^{-\mathrm{j}J2\pi\frac{d}{\lambda}\sin(\theta)}}{1-e^{-\mathrm{j}2\pi\frac{d}{\lambda}\sin(\theta)}}\right|^2 = \boxed{\frac{1}{J^2}\left|\frac{\sin(J\pi\frac{d}{\lambda}\sin(\theta))}{\sin(\pi\frac{d}{\lambda}\sin(\theta))}\right|^2}
\end{aligned}
$$

Main parameters:

- DOA θ
- Ratio $\frac{d}{\lambda}$ (everything scales with wavelength)
- Number of sensors J
- Element spacing d
- Array aperture $L = J \cdot d$

ULA Beampattern: Example $J = 2, \frac{d}{\lambda} = \frac{1}{2}$

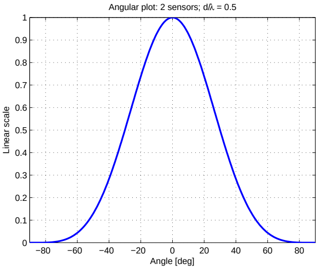
Linear plot:



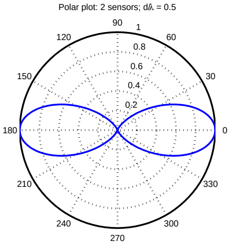
Polar plot:

ULA Beampattern: Example $J = 2, \frac{d}{\lambda} = \frac{1}{2}$

Linear plot:

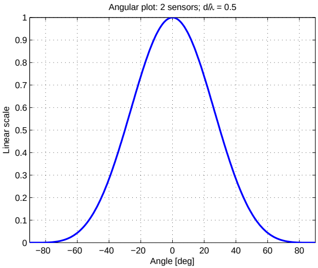


Polar plot:

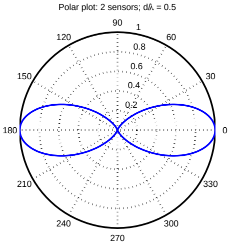


ULA Beampattern: Example $J = 2, \frac{d}{\lambda} = \frac{1}{2}$

Linear plot:



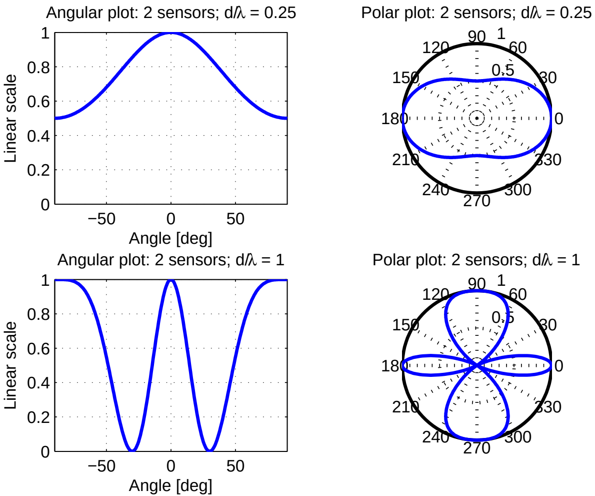
Polar plot:



Some preliminary conclusions:

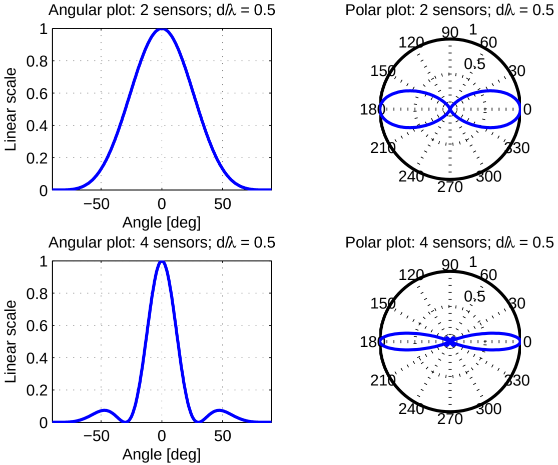
- Ambiguity between 'front' (line of sight) and 'back': $B(\theta) = B(\pi - \theta)$
- Zeroes if numerator of $B(\theta) = 0 \Rightarrow \theta = \arcsin(i \cdot \frac{1}{J} \cdot \frac{\lambda}{d})$
- For $\frac{d}{\lambda} = \frac{1}{2} \Rightarrow$ Zeroes at $\theta = \pm\frac{\pi}{2}$ and mainlobe (3dB) beamwidth: 60^o

ULA Beampattern: $J = 2$, variable $\frac{d}{\lambda}$



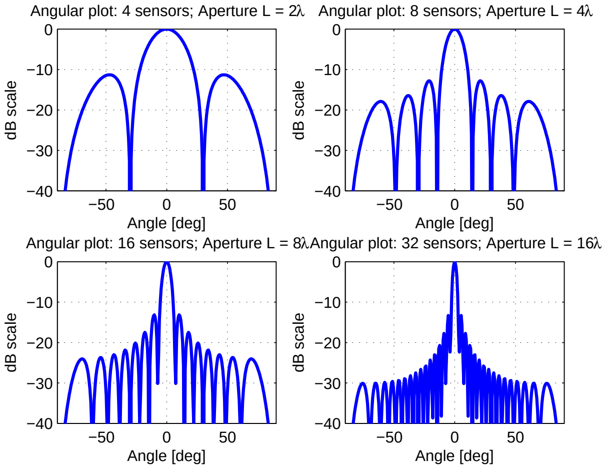
ULA Beampattern: Variable # sensors J

$$L = J \cdot d \text{ with } J \uparrow \text{ and fixed } \frac{d}{\lambda} = \frac{1}{2}$$



ULA Beampattern: dB scale, variable L, $\frac{d}{\lambda} = \frac{1}{2}$

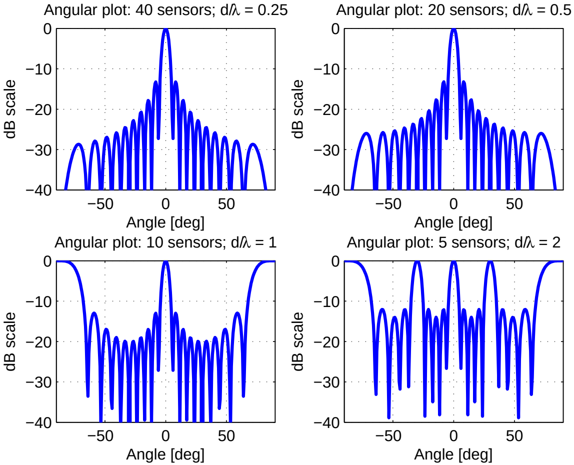
Aperture size $L = J \cdot d$



Note for assignment: Usually maximum at 0 [dB]

ULA Beampattern: dB scale, variable d, fixed L

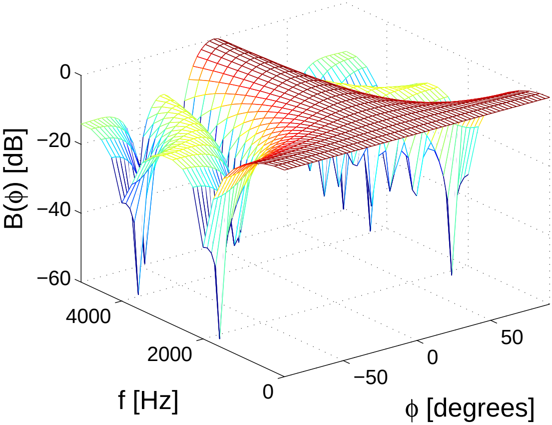
Fixed $L = J \cdot d = 10\lambda$



ULA Beampattern: Frequency dependency

Example: $$d = \frac{\lambda_{min}}{2} = \frac{c}{2 \cdot f_{max}} \approx 3.5[\text{cm}]$$

ULA Far field: J=5, d=0.035, f=0 to 5000 Hz



ULA Beampattern: DFT view

With $u = \frac{d\sin(\theta)}{\lambda} \Rightarrow$ ULA beampattern becomes:

$$B(u) = \frac{1}{J^2}|\underline{w}^h \cdot \underline{a}(u)|^2 = \frac{1}{J^2}|\sum_{p=0}^{J-1} w_p^* e^{-j2\pi p u}|^2$$

ULA Beampattern: DFT view

With $u = \frac{d\sin(\theta)}{\lambda} \Rightarrow$ ULA beampattern becomes:

$$B(u) = \frac{1}{J^2}|\underline{\mathrm{w}}^h \cdot \underline{\mathrm{a}}(u)|^2 = \frac{1}{J^2}|\sum_{p=0}^{J-1} w_p^* e^{-j2\pi p u}|^2$$

Note: Since unambiguous angles $-\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2}$, to avoid aliasing

$$-\frac{1}{2} \leq u \leq \frac{1}{2} \quad \Leftrightarrow \quad d \leq \frac{\lambda}{2}$$

ULA Beampattern: DFT view

With $u = \frac{d\sin(\theta)}{\lambda} \Rightarrow$ ULA beampattern becomes:

$$B(u) = \frac{1}{J^2}|\underline{w}^h \cdot \underline{a}(u)|^2 = \frac{1}{J^2}|\sum_{p=0}^{J-1} w_p^* e^{-j2\pi p u}|^2$$

Note: Since unambiguous angles $-\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2}$, to avoid aliasing

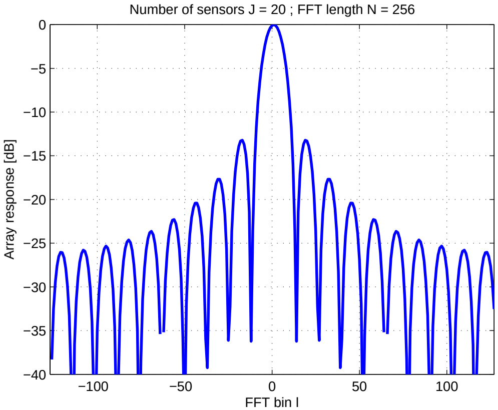
$$-\frac{1}{2} \leq u \leq \frac{1}{2} \quad \Leftrightarrow \quad d \leq \frac{\lambda}{2}$$

Zero padded DFT: With $N \geq J$

$$B_l = \frac{1}{J^2}|\sum_{p=0}^{J-1} w_p^* e^{-j\frac{2\pi}{N}pl}|^2$$

with $l = N \cdot u = N \cdot \frac{d\sin(\theta)}{\lambda}$

ULA Beampattern: DFT view



Compute corresponding angle via: $\theta = \arcsin\left(\frac{l}{N}\cdot\frac{\lambda}{d}\right)$

Data dependent beamforming

Data dependent beamforming

Conventional approaches:

- Beam steering
- Tapering
- Null steering
- Array response design

Beam steering

Purpose:

Compensate propagation path length differences of direct path from source to each sensor resulting is properly aligned direct path signals at the output

Beam steering

Purpose:

Compensate propagation path length differences of direct path from source to each sensor resulting is properly aligned direct path signals at the output

Design procedure:

Design weights such that delays in between sensor elements are compensated $\Rightarrow$ *Beampattern rotates*

Beam steering

Purpose:

Compensate propagation path length differences of direct path from source to each sensor resulting is properly aligned direct path signals at the output

Design procedure:

Design weights such that delays in between sensor elements are compensated $\Rightarrow$ *Beampattern rotates*

Other name: **Delay and Sum Beamformer (DSB)**

Beam steering

Desired source signal $s(t)$, of single frequency $f_d = \frac{c}{\lambda}$, at DOA θ_0:

$$\Rightarrow x_i[k] = s[k] \cdot a_i(\theta_0) = s[k] \cdot \mathrm{e}^{-\mathrm{j}2\pi f_d \tau_i}$$

Beam steering

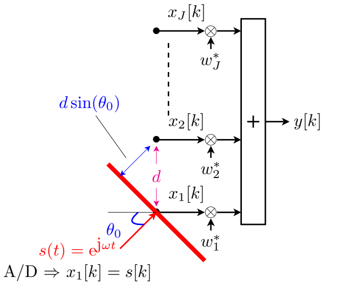
Desired source signal $s(t)$, of single frequency $f_d = \frac{c}{\lambda}$, at DOA θ_0:

$$\Rightarrow x_i[k] = s[k] \cdot a_i(\theta_0) = s[k] \cdot \mathrm{e}^{-\mathrm{j}2\pi f_d \tau_i}$$

ULA delay at sensor i:

$$\tau_i = (i-1) \cdot \frac{d \sin(\theta_0)}{c} \Rightarrow$$

$$x_i[k] = s[k] \cdot \mathrm{e}^{-\mathrm{j}2\pi(i-1)\frac{d\sin(\theta_0)}{\lambda}}$$



Beam steering

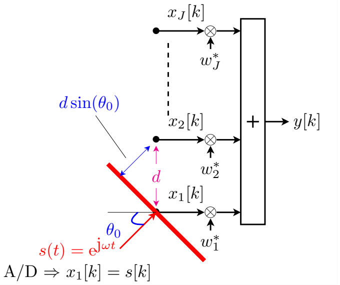
Desired source signal $s(t)$, of single frequency $f_d = \frac{c}{\lambda}$, at DOA θ_0:

$$\Rightarrow x_i[k] = s[k] \cdot a_i(\theta_0) = s[k] \cdot e^{-j2\pi f_d \tau_i}$$

ULA delay at sensor i:

$$\tau_i = (i-1) \cdot \frac{d \sin(\theta_0)}{c} \Rightarrow$$

$x_i[k] =$
$s[k] \cdot e^{-j2\pi(i-1)\frac{d \sin(\theta_0)}{\lambda}}$



In order to properly align desired source (DOA $= \theta_0$) at output:

$$\boxed{w_i^* = e^{+j2\pi(i-1)\frac{d \sin(\theta_0)}{\lambda}}} \quad \Rightarrow \quad y[k] = J \cdot s[k]$$

Beam steering

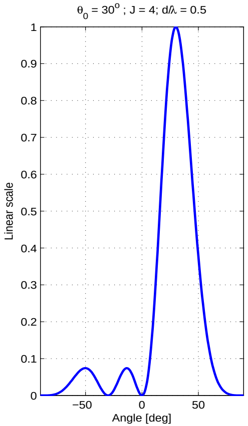
Resulting beampattern shifted/ rotated over θ_0:

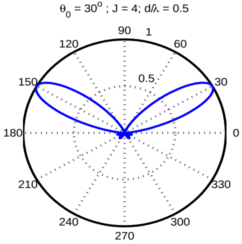
$$B(\theta) = \frac{1}{J^2} |\underline{\mathrm{w}}^h \cdot \underline{\mathrm{a}}(\theta)|^2 = \frac{1}{J^2} \left| \sum_{i=1}^{J} \mathrm{e}^{-\mathrm{j}2\pi f_d (i-1) \frac{d}{c} (\sin(\theta) - \sin(\theta_0))} \right|^2$$

Beam steering

Resulting beampattern shifted/ rotated over θ_0:

$$B(\theta) = \frac{1}{J^2} \left| \underline{w}^h \cdot \underline{a}(\theta) \right|^2 = \frac{1}{J^2} \left| \sum_{i=1}^{J} e^{-j 2\pi f_d (i-1) \frac{d}{c} (\sin(\theta) - \sin(\theta_0))} \right|^2$$

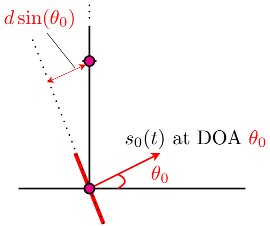




Beam steering

Electronic vs mechanical beamsteering

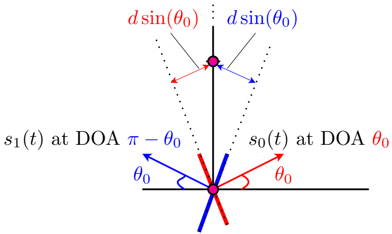
Electronic beamsteering



Beam steering

Electronic vs mechanical beamsteering

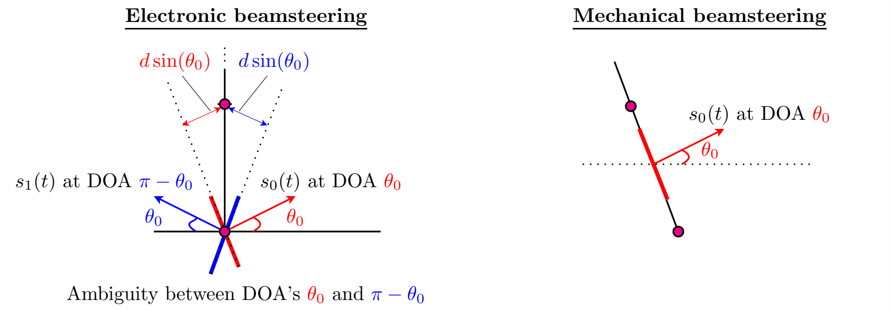
Electronic beamsteering



Ambiguity between DOA's θ_0 and $\pi - \theta_0$

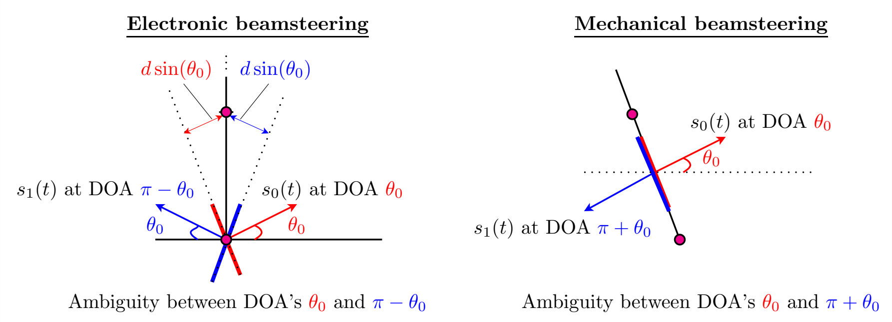
Beam steering

Electronic vs mechanical beamsteering



Beam steering

Electronic vs mechanical beamsteering



Beam steering: matched filter

Another view to beamsteering for spatially white noise

Beam steering: matched filter

Another view to beamsteering for spatially white noise

$$\begin{aligned} y[k] &= \underline{w}^h \cdot \underline{x}[k] = \underline{w}^h \cdot (\underline{a}(\theta)s[k] + \underline{n}[k]) \\ &= \underline{w}^h \cdot \underline{a}(\theta)s[k] + \underline{w}^h \cdot \underline{n}[k] = y_s[k] + y_n[k] \end{aligned}$$

Beam steering: matched filter

Another view to beamsteering for spatially white noise

$$\begin{aligned} y[k] &= \underline{w}^h \cdot \underline{x}[k] = \underline{w}^h \cdot (\underline{a}(\theta)s[k] + \underline{n}[k]) \\ &= \underline{w}^h \cdot \underline{a}(\theta)s[k] + \underline{w}^h \cdot \underline{n}[k] = y_s[k] + y_n[k] \end{aligned}$$

$$\Rightarrow \quad P_y = \sigma_s^2 \cdot \underline{w}^h \left(\underline{a}(\theta) \cdot \underline{a}^h(\theta) \right) \underline{w} + \sigma_n^2 \cdot \underline{w}^h \underline{w} = P_s + P_n$$

Beam steering: matched filter

Another view to beamsteering for spatially white noise

$$\begin{aligned} y[k] &= \underline{w}^h \cdot \underline{x}[k] = \underline{w}^h \cdot (\underline{a}(\theta)s[k] + \underline{n}[k]) \\ &= \underline{w}^h \cdot \underline{a}(\theta)s[k] + \underline{w}^h \cdot \underline{n}[k] = y_s[k] + y_n[k] \end{aligned}$$

$$\Rightarrow \quad P_y = \sigma_s^2 \cdot \underline{w}^h \left(\underline{a}(\theta) \cdot \underline{a}^h(\theta)\right) \underline{w} + \sigma_n^2 \cdot \underline{w}^h \underline{w} = P_s + P_n$$

Define in- and output SNR as: $\mathrm{SNR}_{in} = \frac{\sigma_s^2}{\sigma_n^2} \qquad \mathrm{SNR}_o = \frac{P_s}{P_n}$

Beam steering: matched filter

Another view to beamsteering for spatially white noise

$$\begin{aligned} y[k] &= \underline{w}^h \cdot \underline{x}[k] = \underline{w}^h \cdot (\underline{a}(\theta)s[k] + \underline{n}[k]) \\ &= \underline{w}^h \cdot \underline{a}(\theta)s[k] + \underline{w}^h \cdot \underline{n}[k] = y_s[k] + y_n[k] \end{aligned}$$

$$\Rightarrow \quad P_y = \sigma_s^2 \cdot \underline{w}^h \left(\underline{a}(\theta) \cdot \underline{a}^h(\theta) \right) \underline{w} + \sigma_n^2 \cdot \underline{w}^h \underline{w} = P_s + P_n$$

Define in- and output SNR as: $\mathrm{SNR}_{in} = \frac{\sigma_s^2}{\sigma_n^2}$ $\qquad \mathrm{SNR}_o = \frac{P_s}{P_n}$

$$\Rightarrow \quad \mathrm{SNR}_o = \frac{\underline{w}^h \left(\underline{a}(\theta) \cdot \underline{a}^h(\theta) \right) \underline{w}}{\underline{w}^h \underline{w}} \cdot \frac{\sigma_s^2}{\sigma_n^2} = G(\theta) \cdot \mathrm{SNR}_{in}$$

Beam steering: matched filter

Another view to beamsteering for spatially white noise

$$\begin{aligned} y[k] &= \underline{w}^h \cdot \underline{x}[k] = \underline{w}^h \cdot (\underline{a}(\theta)s[k] + \underline{n}[k]) \\ &= \underline{w}^h \cdot \underline{a}(\theta)s[k] + \underline{w}^h \cdot \underline{n}[k] = y_s[k] + y_n[k] \end{aligned}$$

$$\Rightarrow \quad P_y = \sigma_s^2 \cdot \underline{w}^h \left(\underline{a}(\theta) \cdot \underline{a}^h(\theta) \right) \underline{w} + \sigma_n^2 \cdot \underline{w}^h \underline{w} = P_s + P_n$$

Define in- and output SNR as: $\text{SNR}_{in} = \frac{\sigma_s^2}{\sigma_n^2}$ $\quad \text{SNR}_o = \frac{P_s}{P_n}$

$$\Rightarrow \quad \text{SNR}_o = \frac{\underline{w}^h \left(\underline{a}(\theta) \cdot \underline{a}^h(\theta) \right) \underline{w}}{\underline{w}^h \underline{w}} \cdot \frac{\sigma_s^2}{\sigma_n^2} = G(\theta) \cdot \text{SNR}_{in}$$

Thus SNR_o maximized by choosing $\underline{w} = \beta \cdot \underline{a}(\theta)$, with β some constant

Beam steering: matched filter

<u>**Another view to beamsteering for spatially white noise**</u>

$$\begin{aligned} y[k] &= \underline{w}^h \cdot \underline{x}[k] = \underline{w}^h \cdot (\underline{a}(\theta)s[k] + \underline{n}[k]) \\ &= \underline{w}^h \cdot \underline{a}(\theta)s[k] + \underline{w}^h \cdot \underline{n}[k] = y_s[k] + y_n[k] \end{aligned}$$

$$\Rightarrow \quad P_y = \sigma_s^2 \cdot \underline{w}^h \left(\underline{a}(\theta) \cdot \underline{a}^h(\theta) \right) \underline{w} + \sigma_n^2 \cdot \underline{w}^h \underline{w} = P_s + P_n$$

Define in- and output SNR as: $\mathrm{SNR}_{in} = \frac{\sigma_s^2}{\sigma_n^2}$ $\qquad \mathrm{SNR}_o = \frac{P_s}{P_n}$

$$\Rightarrow \quad \mathrm{SNR}_o = \frac{\underline{w}^h \left(\underline{a}(\theta) \cdot \underline{a}^h(\theta) \right) \underline{w}}{\underline{w}^h \underline{w}} \cdot \frac{\sigma_s^2}{\sigma_n^2} = G(\theta) \cdot \mathrm{SNR}_{in}$$

Thus SNR_o maximized by choosing $\underline{w} = \beta \cdot \underline{a}(\theta)$, with β some constant

Conclusion:

Spatial filter that maximizes DOA $\equiv$ <u>**Matched filter**</u> ($=$ max SNR_o)

Beam steering

Notes on beamsteering (delay and sum beamforming (DSB):

- Source location (or DOA) required
- Position sensors must be known
- DSB aligns only direct path
- Difference between electronic vs mechanical beamsteering
- In sense of max SNR_o, spatial matched filter optimum for spatially white noise

Beam steering: Discrete-time domain

Steering delay $\tau = \frac{d \sin(\theta)}{c}$; Sample rate $f_s = \frac{1}{T_s} \Rightarrow$

Beam steering: Discrete-time domain

Steering delay $\tau = \frac{d \sin(\theta)}{c}$; Sample rate $f_s = \frac{1}{T_s} \Rightarrow$

Steering possible for values $\tau = \alpha \cdot T_s$ with $|\alpha| = 0, 1, \cdots$

Beam steering: Discrete-time domain

Steering delay $\tau = \frac{d\sin(\theta)}{c}$; Sample rate $f_s = \frac{1}{T_s} \Rightarrow$

Steering possible for values $\tau = \alpha \cdot T_s$ with $|\alpha| = 0, 1, \cdots$

Possible steering to: $\theta_s = \arcsin\left(\frac{c \cdot \alpha \cdot T_s}{d}\right)$ with $|\alpha| = 0, 1, \cdots$

Beam steering: Discrete-time domain

Steering delay $\tau = \frac{d \sin(\theta)}{c}$; Sample rate $f_s = \frac{1}{T_s} \Rightarrow$

Steering possible for values $\tau = \alpha \cdot T_s$ with $|\alpha| = 0, 1, \cdots$

Possible steering to: $\theta_s = \arcsin\left(\frac{c \cdot \alpha \cdot T_s}{d}\right)$ with $|\alpha| = 0, 1, \cdots$

Example:

$d = \frac{\lambda}{2}$, $\lambda = \frac{c}{f_0}$ and $f_s = 2\gamma f_0 \Rightarrow \theta_s = \arcsin\left(\frac{\alpha}{\gamma}\right) \Leftrightarrow |\alpha| \leq \gamma$

Beam steering: Discrete-time domain

Steering delay $\tau = \frac{d \sin(\theta)}{c}$; Sample rate $f_s = \frac{1}{T_s} \Rightarrow$

Steering possible for values $\tau = \alpha \cdot T_s$ with $|\alpha| = 0, 1, \cdots$

Possible steering to: $\theta_s = \arcsin\left(\frac{c \cdot \alpha \cdot T_s}{d}\right)$ with $|\alpha| = 0, 1, \cdots$

Example:

$d = \frac{\lambda}{2}$, $\lambda = \frac{c}{f_0}$ and $f_s = 2\gamma f_0 \Rightarrow \theta_s = \arcsin\left(\frac{\alpha}{\gamma}\right) \Leftrightarrow |\alpha| \leq \gamma$

Conclusion: Beam can only be steered to $1 + 2\lfloor \gamma \rfloor$ different angles!

Beam steering: Discrete-time domain

Steering delay $\tau = \frac{d \sin(\theta)}{c}$; Sample rate $f_s = \frac{1}{T_s} \Rightarrow$

Steering possible for values $\tau = \alpha \cdot T_s$ with $|\alpha| = 0, 1, \cdots$

Possible steering to: $\theta_s = \arcsin\left(\frac{c \cdot \alpha \cdot T_s}{d}\right)$ with $|\alpha| = 0, 1, \cdots$

Example:

$d = \frac{\lambda}{2}$, $\lambda = \frac{c}{f_0}$ and $f_s = 2\gamma f_0 \Rightarrow \theta_s = \arcsin\left(\frac{\alpha}{\gamma}\right) \Leftrightarrow |\alpha| \leq \gamma$

Conclusion: Beam can only be steered to $1 + 2\lfloor\gamma\rfloor$ different angles!

Example:

$f_s = 4 \cdot f_0 \Rightarrow$ Beam can be steered to 0^o, $\pm 30^o$, $\pm 90^o$

Beam steering: Discrete-time domain

Steering delay $\tau = \frac{d\sin(\theta)}{c}$; Sample rate $f_s = \frac{1}{T_s} \Rightarrow$

Steering possible for values $\tau = \alpha \cdot T_s$ with $|\alpha| = 0, 1, \cdots$

Possible steering to: $\theta_s = \arcsin\left(\frac{c \cdot \alpha \cdot T_s}{d}\right)$ with $|\alpha| = 0, 1, \cdots$

Example:

$d = \frac{\lambda}{2}$, $\lambda = \frac{c}{f_0}$ and $f_s = 2\gamma f_0 \Rightarrow \theta_s = \arcsin\left(\frac{\alpha}{\gamma}\right) \Leftrightarrow |\alpha| \leq \gamma$

Conclusion: Beam can only be steered to $1 + 2\lfloor\gamma\rfloor$ different angles!

Example:

$f_s = 4 \cdot f_0 \Rightarrow$ Beam can be steered to 0^o, $\pm 30^o$, $\pm 90^o$

If more directions needed: Use interpolation and/or fractional delays

Beam steering: Tapering

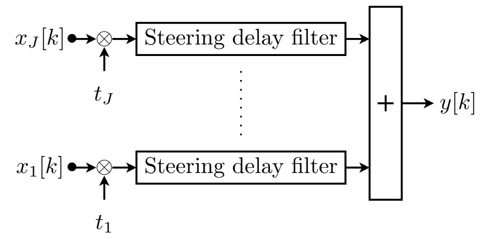
Goal:

Control shape of response i.e. to form beam. Thus window (weighted) sensor signals to compromise between resolution (main lobe width) and leakage (side lobe level) $\Rightarrow \underline{\mathrm{w}}_t = \underline{\mathrm{t}} \odot \underline{\mathrm{w}}$ *with* $\underline{\mathrm{t}}$ *taper window and* $\odot$ *element by element multiplication*

Beam steering: Tapering

Goal:

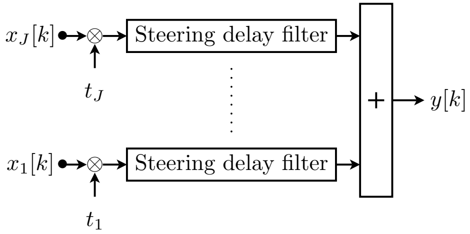
Control shape of response i.e. to form beam. Thus window (weighted) sensor signals to compromise between resolution (main lobe width) and leakage (side lobe level) $\Rightarrow \underline{w}_t = \underline{t} \odot \underline{w}$ *with* $\underline{t}$ *taper window and* $\odot$ *element by element multiplication*



Beam steering: Tapering

Goal:

Control shape of response i.e. to form beam. Thus window (weighted) sensor signals to compromise between resolution (main lobe width) and leakage (side lobe level) $\Rightarrow \underline{\mathrm{w}}_t = \underline{\mathrm{t}} \odot \underline{\mathrm{w}}$ *with* $\underline{\mathrm{t}}$ *taper window and* $\odot$ *element by element multiplication*



Notes:

- Taper weights used to shape beampattern
- Filters approximate delays (linear phase over frequency band of interest)

Null-steering

Goal: *Calculate J weights to meet M constraints, e.g. to amplify the desired, and to attenuate the undesired sources*

Null-steering

Goal: *Calculate J weights to meet M constraints, e.g. to amplify the desired, and to attenuate the undesired sources*

With $J \times 1$ weight vector $\underline{w} = (w_1, \cdots, w_J)^t$ set up M constraints:

$$\begin{array}{c} \underline{a}^h(\omega_1, \theta_1) \cdot \underline{w} = r_d(\omega_1, \theta_1) \\ \vdots \\ \underline{a}^h(\omega_M, \theta_M) \cdot \underline{w} = r_d(\omega_M, \theta_M) \end{array} \quad \Leftrightarrow \quad \boxed{A^h \cdot \underline{w} = \underline{r}_d}$$

with

$$\begin{aligned} A \equiv A(\omega, \theta) &= (\underline{a}(\omega_1, \theta_1), \cdots, \underline{a}(\omega_M, \theta_M)) \\ \underline{r}_d \equiv \underline{r}_d(\omega, \theta) &= (r_d(\omega_1, \theta_1), \cdots, r_d(\omega_M, \theta_M))^h \end{aligned}$$

Null-steering

Goal: *Calculate J weights to meet M constraints, e.g. to amplify the desired, and to attenuate the undesired sources*

With $J \times 1$ weight vector $\underline{w} = (w_1, \cdots, w_J)^t$ set up M constraints:

$$\left.\begin{array}{r} \underline{a}^h(\omega_1, \theta_1) \cdot \underline{w} = r_d(\omega_1, \theta_1) \\ \vdots \\ \underline{a}^h(\omega_M, \theta_M) \cdot \underline{w} = r_d(\omega_M, \theta_M) \end{array}\right. \quad \Leftrightarrow \quad \boxed{\mathrm{A}^h \cdot \underline{w} = \underline{r}_d}$$

with

$$\begin{aligned} \mathrm{A} \equiv \mathrm{A}(\omega, \theta) &= (\underline{a}(\omega_1, \theta_1), \cdots, \underline{a}(\omega_M, \theta_M)) \\ \underline{r}_d \equiv \underline{r}_d(\omega, \theta) &= (r_d(\omega_1, \theta_1), \cdots, r_d(\omega_M, \theta_M))^h \end{aligned}$$

Case: $M < J$ (*Less constraints then weights*)

$$\boxed{\underline{w} = (\mathrm{A}^h)^\dagger \cdot \underline{r}_d = \mathrm{A} \cdot \left(\mathrm{A}^h \cdot \mathrm{A}\right)^{-1} \cdot \underline{r}_d}$$

Null-steering

Example: *Null signal at 90° with 2 sensor ULA at distance half wavelength. Thus:* $J = 2$, $M = 1$, $\theta_u = \pi/2$, $r(\theta_u) = 0$, $d/\lambda = 1/2$

Null-steering

Example: *Null signal at 90° with 2 sensor ULA at distance half wavelength. Thus:* $J = 2$, $M = 1$, $\theta_u = \pi/2$, $r(\theta_u) = 0$, $d/\lambda = 1/2$

$$\mathrm{A}^h = \left(1, e^{j\pi \sin(\pi/2)}\right) = (1, -1) \quad \text{and} \quad \underline{r}_d = (0)$$

Null-steering

Example: *Null signal at 90° with 2 sensor ULA at distance half wavelength. Thus:* $J = 2$, $M = 1$, $\theta_u = \pi/2$, $r(\theta_u) = 0$, $d/\lambda = 1/2$

$$\mathrm{A}^h = \left(1, e^{j\pi \sin(\pi/2)}\right) = (1, -1) \quad \text{and} \quad \underline{\mathrm{r}}_d = (0)$$

$$\Rightarrow \ \underline{\mathrm{w}} = \mathrm{A} \cdot \left(\mathrm{A}^h \cdot \mathrm{A}\right)^{-1} \cdot \underline{\mathrm{r}}_d = \cdots = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Null-steering

Example: *Null signal at 90° with 2 sensor ULA at distance half wavelength. Thus:* $J = 2$, $M = 1$, $\theta_u = \pi/2$, $r(\theta_u) = 0$, $d/\lambda = 1/2$

$$\mathrm{A}^h = \left(1, e^{j\pi \sin(\pi/2)}\right) = (1, -1) \quad \text{and} \quad \underline{\mathrm{r}}_d = (0)$$

$$\Rightarrow \underline{\mathrm{w}} = \mathrm{A} \cdot \left(\mathrm{A}^h \cdot \mathrm{A}\right)^{-1} \cdot \underline{\mathrm{r}}_d = \cdots = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Indeed nulls at 90^0 $\cdots$ however also all others!

Null-steering

Example: *Null signal at 90° with 2 sensor ULA at distance half wavelength. Thus:* $J = 2$, $M = 1$, $\theta_u = \pi/2$, $r(\theta_u) = 0$, $d/\lambda = 1/2$

$$\mathrm{A}^h = \left(1, e^{j\pi \sin(\pi/2)}\right) = (1, -1) \quad \text{and} \quad \underline{\mathrm{r}}_d = (0)$$

$$\Rightarrow \underline{\mathrm{w}} = \mathrm{A} \cdot \left(\mathrm{A}^h \cdot \mathrm{A}\right)^{-1} \cdot \underline{\mathrm{r}}_d = \cdots = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Indeed nulls at 90^0 $\cdots$ however also all others!

Note: For this solution we obtain a "line through origin"

$$\mathrm{A}^h \cdot \underline{\mathrm{w}} = \underline{\mathrm{r}}_d \quad \Leftrightarrow \quad (1, -1) \cdot \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = 0 \quad \Rightarrow \quad w_1 = w_2$$

Since $J = 2$ and $M = 1$: One degree of freedom left!
This can be used e.g. to overcome the solution $w_1 = w_2 = 0$

Null-steering

Example: *Two (complex) plane waves. One desired at 0º, other undesired at 30º, ULA with 3 sensors at distance half wavelength*
$\Rightarrow J = 3,\ M = 2, \theta_d = 0,\ \theta_u = \pi/6,\ r(\theta_d) = 1,\ r(\theta_u) = 0,$
$d/\lambda = 1/2$

Null-steering

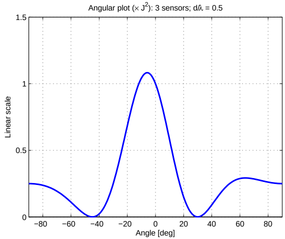
Example: *Two (complex) plane waves. One desired at 0°, other undesired at 30°, ULA with 3 sensors at distance half wavelength* $\Rightarrow J = 3,\ M = 2, \theta_d = 0,\ \theta_u = \pi/6,\ r(\theta_d) = 1,\ r(\theta_u) = 0,$ $d/\lambda = 1/2 \Rightarrow$

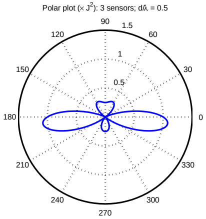
$$\mathrm{A}^h = \begin{pmatrix} 1 & 1 & 1 \\ 1 & \mathrm{e}^{\mathrm{j}\pi\sin(\pi/6)} & \mathrm{e}^{\mathrm{j}2\pi\sin(\pi/6)} \end{pmatrix} ;\ \underline{\mathrm{r}}_d = (1,0)^t \ \Rightarrow\ \underline{\mathrm{w}} = \frac{1}{8}\begin{pmatrix} 3-\mathrm{j} \\ 2 \\ 3+\mathrm{j} \end{pmatrix}$$

Null-steering

Example: *Two (complex) plane waves. One desired at 0º, other undesired at 30º, ULA with 3 sensors at distance half wavelength* $\Rightarrow J = 3,\ M = 2, \theta_d = 0,\ \theta_u = \pi/6,\ r(\theta_d) = 1,\ r(\theta_u) = 0,$ $d/\lambda = 1/2 \Rightarrow$

$$A^h = \begin{pmatrix} 1 & 1 & 1 \\ 1 & e^{j\pi \sin(\pi/6)} & e^{j2\pi \sin(\pi/6)} \end{pmatrix} ; \underline{r}_d = (1,0)^t \ \Rightarrow\ \underline{w} = \frac{1}{8} \begin{pmatrix} 3-j \\ 2 \\ 3+j \end{pmatrix}$$





Null-steering: Conclusions

Null steering used to cancel plane waves arriving from **known** *directions*

Null-steering: Conclusions

Null steering used to cancel plane waves arriving from **known** *directions*

Required knowledge:

DOA of desired/undesired signals, position of sensors

Null-steering: Conclusions

Null steering used to cancel plane waves arriving from **known** *directions*

Required knowledge:
DOA of desired/undesired signals, position of sensors

Performance:
SNR **not** maximized, but nulls can be put in DOA's of interferences

Null-steering: Conclusions

Null steering used to cancel plane waves arriving from **known** *directions*

Required knowledge:
DOA of desired/undesired signals, position of sensors

Performance:
SNR **not** maximized, but nulls can be put in DOA's of interferences

Properties:

- Result not robust to frequency jammer
- J weights can set maximum J predefined conditions
- Needs much a priori information
- $M < J$: Add extra constraints (e.g. minimize output power)
- Use FIR filters for broadband

Array response design: $M > J$

Procedure: $\underline{\mathrm{w}} = \arg\min_{\underline{\mathrm{w}}}\{E\}$

Array response design: $M > J$

$$\text{Procedure:} \quad \underline{w} = \arg\min_{\underline{w}}\{E\}$$

$$\begin{aligned}
\text{with error } E &= |A^h \cdot \underline{w} - \underline{r}_d|^2 = (\underline{w}^h \cdot A - \underline{r}_d^h) \cdot (A^h \cdot \underline{w} - \underline{r}_d) \\
&= \underline{w}^h A A^h \underline{w} - \underline{w}^h A \underline{r}_d - \underline{r}_d^h A^h \underline{w} - \underline{r}_d^h \underline{r}_d
\end{aligned}$$

Array response design: $M > J$

$$\text{Procedure:} \quad \underline{w} = \arg\min_{\underline{w}} \{E\}$$

$$\begin{aligned} \text{with error } E &= |A^h \cdot \underline{w} - \underline{r}_d|^2 = (\underline{w}^h \cdot A - \underline{r}_d^h) \cdot (A^h \cdot \underline{w} - \underline{r}_d) \\ &= \underline{w}^h A A^h \underline{w} - \underline{w}^h A \underline{r}_d - \underline{r}_d^h A^h \underline{w} - \underline{r}_d^h \underline{r}_d \end{aligned}$$

$$\frac{dE}{d\underline{w}} = \underline{0} \quad \Rightarrow \quad \boxed{\underline{w} = \left(A \cdot A^h\right)^{-1} \cdot A \cdot \underline{r}_d}$$

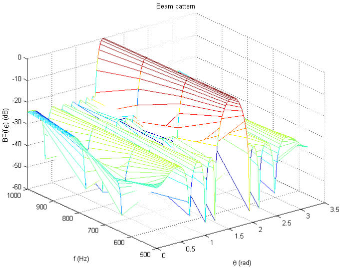
Array response design: $M > J$

$$\text{Procedure:} \quad \underline{w} = \arg\min_{\underline{w}}\{E\}$$

$$\begin{aligned} \text{with error } E &= |A^h \cdot \underline{w} - \underline{r}_d|^2 = (\underline{w}^h \cdot A - \underline{r}_d^h) \cdot (A^h \cdot \underline{w} - \underline{r}_d) \\ &= \underline{w}^h A A^h \underline{w} - \underline{w}^h A \underline{r}_d - \underline{r}_d^h A^h \underline{w} - \underline{r}_d^h \underline{r}_d \end{aligned}$$

$$\frac{dE}{d\underline{w}} = \underline{0} \quad \Rightarrow \quad \boxed{\underline{w} = \left(A \cdot A^h\right)^{-1} \cdot A \cdot \underline{r}_d}$$

Example: $B(\theta) = 1$ *at* $\theta = \pi/2$ *and* < -25 *[dB] outside this area*



Beamforming: overview

Data independent (conventional approach): **(Part IB)**

- **Beamsteering (DSB, phased array)**
- **Tapering**
- **Null steering/ Array response design**

Beamforming: overview

Data independent (conventional approach): **(Part IB)**

- **Beamsteering (DSB, phased array)**
- **Tapering**
- **Null steering/ Array response design**

Data dependent (statistical optimum): (Part IC)

- **Minimum Sidelobe Canceller**
- **Wiener**
- **Maximum SNR**
- **Linear Constraint Minimum Variance**
- **Generalized Sidelobe Canceller**

Summary part II

Far field, narrowband source, direction vector $\underline{v}$, at position $\underline{p}_i$:

$s[k]e^{-j\omega\tau_i}$ with $\tau_i = \frac{\underline{v}^t \cdot \underline{p}_i}{c}$, $\omega = 2\pi f$, $f = \frac{c}{\lambda}$

For ULA-case at sensor i: $s[k] \cdot a_i(\theta)$ with $a_i(\theta) = e^{-j2\pi(i-1)\frac{d\sin(\theta)}{\lambda}}$

Array response/ Angular response/ Directivity pattern:

$r(\theta) = \underline{w}^h \cdot \underline{a}(\theta)$

Beampattern: $B(\theta) = \frac{1}{J^2}|r(\theta)|^2$

For ULA with inter element distance d and $\underline{w} = \underline{1}$:

$$B(\theta) = \frac{1}{J^2}|\underline{1}^t \cdot \underline{a}(\theta)|^2 = \frac{1}{J^2}\left|\frac{1 - e^{-jJ2\pi\frac{d}{\lambda}\sin(\theta)}}{1 - e^{-j2\pi\frac{d}{\lambda}\sin(\theta)}}\right|^2$$

Summary part II

Beamsteering (shifted/ rotated over θ_0):

$$B(\theta) = \frac{1}{J^2}\left|\underline{\mathrm{w}}^h \cdot \underline{\mathrm{a}}(\theta)\right|^2 = \frac{1}{J^2}\left|\sum_{i=1}^{J} \mathrm{e}^{-\mathrm{j}2\pi f_d (i-1)\frac{d}{c}(\sin(\theta)-\sin(\theta_0))}\right|^2$$

Constrained beamforming:

With $J \times 1$ weight vector $\underline{\mathrm{w}} = (w_1, \cdots, w_J)^t$ set up M constraints:

$$\begin{aligned} \underline{\mathrm{a}}^h(\omega_1, \theta_1) \cdot \underline{\mathrm{w}} &= r_d(\omega_1, \theta_1) \\ &\vdots \\ \underline{\mathrm{a}}^h(\omega_M, \theta_M) \cdot \underline{\mathrm{w}} &= r_d(\omega_M, \theta_M) \end{aligned} \quad \Leftrightarrow \quad \boxed{\mathrm{A}^h \cdot \underline{\mathrm{w}} = \underline{\mathrm{r}}_d}$$

Null steering ($M < J$): $\underline{\mathrm{w}} = (\mathrm{A}^h)^\dagger \cdot \underline{\mathrm{r}}_d = \mathrm{A} \cdot \left(\mathrm{A}^h \cdot \mathrm{A}\right)^{-1} \cdot \underline{\mathrm{r}}_d$

Array response design ($M > J$): $\underline{\mathrm{w}} = \left(\mathrm{A} \cdot \mathrm{A}^h\right)^{-1} \cdot \mathrm{A} \cdot \underline{\mathrm{r}}_d$

DOA + Optimum and Adaptive Array Signal Processing

(Part IC)

DOA estimation

Goal:

Estimate Direction Of Arrival (DOA) of sources (and interferences) from noisy observations, in order to locate and/or track these sources

DOA estimation

Goal:

Estimate Direction Of Arrival (DOA) of sources (and interferences) from noisy observations, in order to locate and/or track these sources

Main techniques based on:

1. Maximizing power of steered beamformer
2. High-resolution spectral estimation concepts
3. Employing time-difference of arrival (*not in this course*)

Conventional DOA

1. Maximizing power of steered beamformer:

E.g. noisy observation, one source, J sensors:

$$\underline{x}[k] = \underline{a}(\theta_p) \cdot s[k] + \underline{n}[k] \quad \Rightarrow \quad R_x = \sigma_s^2 \underline{a}(\theta_p) \underline{a}^h(\theta_p) + \sigma_n^2 \mathbf{I}$$

$$y[k] = \underline{w}^h \cdot \underline{x}[k] \quad \Rightarrow \quad P_y = E\{|y[k]|^2\} = \underline{w}^h \cdot R_x \cdot \underline{w}$$

Conventional DOA

1. Maximizing power of steered beamformer:

E.g. noisy observation, one source, J sensors:

$$\underline{x}[k] = \underline{a}(\theta_p) \cdot s[k] + \underline{n}[k] \quad \Rightarrow \quad \mathrm{R}_x = \sigma_s^2 \underline{a}(\theta_p)\underline{a}^h(\theta_p) + \sigma_n^2 \mathbf{I}$$

$$y[k] = \underline{w}^h \cdot \underline{x}[k] \quad \Rightarrow \quad P_y = E\{|y[k]|^2\} = \underline{w}^h \cdot \mathrm{R}_x \cdot \underline{w}$$

Spatial spectrum: $\quad P(\theta) = \dfrac{P_y(\theta)}{||\underline{w}||^2}$

Conventional DOA

1. Maximizing power of steered beamformer:

E.g. noisy observation, one source, J sensors:

$$\underline{x}[k] = \underline{a}(\theta_p) \cdot s[k] + \underline{n}[k] \quad \Rightarrow \quad R_x = \sigma_s^2 \underline{a}(\theta_p)\underline{a}^h(\theta_p) + \sigma_n^2 \mathbf{I}$$

$$y[k] = \underline{w}^h \cdot \underline{x}[k] \quad \Rightarrow \quad P_y = E\{|y[k]|^2\} = \underline{w}^h \cdot R_x \cdot \underline{w}$$

Spatial spectrum: $$P(\theta) = \frac{P_y(\theta)}{||\underline{w}||^2}$$

Beamsteering: Max.$\{P_s/P_n\} \Leftrightarrow \underline{w} \equiv \underline{a}(\theta)$

Conventional DOA

1. Maximizing power of steered beamformer:

E.g. noisy observation, one source, J sensors:

$$\underline{x}[k] = \underline{a}(\theta_p) \cdot s[k] + \underline{n}[k] \quad \Rightarrow \quad R_x = \sigma_s^2 \underline{a}(\theta_p) \underline{a}^h(\theta_p) + \sigma_n^2 I$$

$$y[k] = \underline{w}^h \cdot \underline{x}[k] \quad \Rightarrow \quad P_y = E\{|y[k]|^2\} = \underline{w}^h \cdot R_x \cdot \underline{w}$$

Spatial spectrum: $P(\theta) = \dfrac{P_y(\theta)}{||\underline{w}||^2}$

Beamsteering: $\text{Max.}\{P_s/P_n\} \Leftrightarrow \underline{w} \equiv \underline{a}(\theta)$

$$\boxed{P(\theta) = \frac{\underline{a}^h(\theta) R_x \underline{a}(\theta)}{J}}$$

Conventional DOA

1. Maximizing power of steered beamformer:

E.g. noisy observation, one source, J sensors:

$$\underline{x}[k] = \underline{a}(\theta_p) \cdot s[k] + \underline{n}[k] \quad \Rightarrow \quad R_x = \sigma_s^2 \underline{a}(\theta_p)\underline{a}^h(\theta_p) + \sigma_n^2 I$$

$$y[k] = \underline{w}^h \cdot \underline{x}[k] \quad \Rightarrow \quad P_y = E\{|y[k]|^2\} = \underline{w}^h \cdot R_x \cdot \underline{w}$$

Spatial spectrum: $P(\theta) = \dfrac{P_y(\theta)}{||\underline{w}||^2}$

Beamsteering: Max.$\{P_s/P_n\} \Leftrightarrow \underline{w} \equiv \underline{a}(\theta)$

$$\boxed{P(\theta) = \frac{\underline{a}^h(\theta) R_x \underline{a}(\theta)}{J}}$$

Thus peak in $P(\theta)$ is DOA location p!

Conventional DOA

1. Maximizing power of steered beamformer:

E.g. noisy observation, one source, J sensors:

$$\underline{x}[k] = \underline{a}(\theta_p) \cdot s[k] + \underline{n}[k] \quad \Rightarrow \quad R_x = \sigma_s^2 \underline{a}(\theta_p)\underline{a}^h(\theta_p) + \sigma_n^2 I$$

$$y[k] = \underline{w}^h \cdot \underline{x}[k] \quad \Rightarrow \quad P_y = E\{|y[k]|^2\} = \underline{w}^h \cdot R_x \cdot \underline{w}$$

Spatial spectrum: $P(\theta) = \dfrac{P_y(\theta)}{||\underline{w}||^2}$

Beamsteering: $\text{Max.}\{P_s/P_n\} \Leftrightarrow \underline{w} \equiv \underline{a}(\theta)$

$$\boxed{P(\theta) = \frac{\underline{a}^h(\theta) R_x \underline{a}(\theta)}{J}}$$

Thus peak in $P(\theta)$ is DOA location p!

Note: In practice estimate $\hat{R}_x = \dfrac{1}{T}\sum_{k=1}^{T} \underline{x}[k] \cdot \underline{x}^h[k]$

Conventional DOA

Example: ULA

$d/\lambda = 0.5$; $P = 4$ sources (at -20,0,40 and 70 degrees); $J = 8$ sensors

High-resolution DOA

High-resolution DOA technique based on signal subspace (see Appendix) method:

Spectral MUSIC = MUltiple Signal Classification

High-resolution DOA

High-resolution DOA technique based on signal subspace (see Appendix) method:

Spectral MUSIC = MUltiple Signal Classification

Signal model: J sensors, P sources, $P < J$

$$x_i = \sum_{p=1}^{P} a_i(\theta_p) \cdot s_p[k] + n_i[k] \text{for } i = 1, \cdots, J \quad \Leftrightarrow \quad \underline{\mathrm{x}}[k] = \mathrm{A} \cdot \underline{\mathrm{s}}[k] + \underline{\mathrm{n}}[k]$$

High-resolution DOA

High-resolution DOA technique based on signal subspace (see Appendix) method:

Spectral MUSIC = MUltiple Signal Classification

Signal model: J sensors, P sources, $P < J$

$$x_i = \sum_{p=1}^{P} a_i(\theta_p) \cdot s_p[k] + n_i[k] \text{for } i = 1, \cdots, J \quad \Leftrightarrow \quad \underline{\mathrm{x}}[k] = \mathrm{A} \cdot \underline{\mathrm{s}}[k] + \underline{\mathrm{n}}[k]$$

Covariance structure:

$$\mathrm{R}_x = E\{\underline{\mathrm{x}} \cdot \underline{\mathrm{x}}^h\} = \mathrm{A}\mathrm{R}_s\mathrm{A} + \mathrm{R}_n$$

$$\text{with} \quad \mathrm{R}_s = E\{\underline{\mathrm{s}} \cdot \underline{\mathrm{s}}^h\} = \mathrm{diag}\{\sigma_{s_1}^2, \cdots, \sigma_{s_P}^2\} \quad \text{and} \quad \mathrm{R}_n = \sigma_n^2 \mathrm{I}$$

High-resolution DOA

Recall from appendix:

$$\mathsf{R}_x = \mathsf{U}_x \Lambda_x \mathsf{U}_x^h = \mathsf{U}_s \Lambda_{s,n} \mathsf{U}_s^h + \mathsf{U}_n \Lambda_n \mathsf{U}_x^n$$

$$\mathsf{U}_x = (\underline{\mathsf{u}}_1, \cdots, \underline{\mathsf{u}}_J) \; ; \; \Lambda_x = \mathrm{diag}\{\lambda_{x_1}, \cdots, \lambda_{x_J}\}$$

High-resolution DOA

Recall from appendix:

$$\mathsf{R}_x = \mathsf{U}_x \Lambda_x \mathsf{U}_x^h = \mathsf{U}_s \Lambda_{s,n} \mathsf{U}_s^h + \mathsf{U}_n \Lambda_n \mathsf{U}_x^n$$
$$\mathsf{U}_x = (\underline{\mathsf{u}}_1, \cdots, \underline{\mathsf{u}}_J) \ ; \ \Lambda_x = \mathrm{diag}\{\lambda_{x_1}, \cdots, \lambda_{x_J}\}$$

Signal subspace:

$$\mathsf{U}_s = (\underline{\mathsf{u}}_1, \cdots, \underline{\mathsf{u}}_P) \ ; \ \Lambda_{s,n} = \mathrm{diag}\{\lambda_{s_1} + \sigma_n^2, \cdots, \lambda_{s_P} + \sigma_n^2\}$$

High-resolution DOA

Recall from appendix:

$$R_x = U_x \Lambda_x U_x^h = U_s \Lambda_{s,n} U_s^h + U_n \Lambda_n U_x^n$$
$$U_x = (\underline{u}_1, \cdots, \underline{u}_J) \; ; \; \Lambda_x = \text{diag}\{\lambda_{x_1}, \cdots, \lambda_{x_J}\}$$

Signal subspace:

$$U_s = (\underline{u}_1, \cdots, \underline{u}_P) \; ; \; \Lambda_{s,n} = \text{diag}\{\lambda_{s_1} + \sigma_n^2, \cdots, \lambda_{s_P} + \sigma_n^2\}$$

Noise subspace:

$$U_n = (\underline{u}_{P+1}, \cdots, \underline{u}_J) \; ; \; \Lambda_n = \text{diag}\{\sigma_n^2, \cdots, \sigma_n^2\}$$

High-resolution DOA

transpose
conjugate
orthogonal

Recall from appendix:

$$\mathrm{R}_x = \mathrm{U}_x \Lambda_x \mathrm{U}_x^h = \mathrm{U}_s \Lambda_{s,n} \mathrm{U}_s^h + \mathrm{U}_n \Lambda_n \mathrm{U}_x^n$$

$$\mathrm{U}_x = (\underline{\mathrm{u}}_1, \cdots, \underline{\mathrm{u}}_J) \; ; \; \Lambda_x = \mathrm{diag}\{\lambda_{x_1}, \cdots, \lambda_{x_J}\}$$

Signal subspace:

$$\mathrm{U}_s = (\underline{\mathrm{u}}_1, \cdots, \underline{\mathrm{u}}_P) \; ; \; \Lambda_{s,n} = \mathrm{diag}\{\lambda_{s_1} + \sigma_n^2, \cdots, \lambda_{s_P} + \sigma_n^2\}$$

Noise subspace:

$$\mathrm{U}_n = (\underline{\mathrm{u}}_{P+1}, \cdots, \underline{\mathrm{u}}_J) \; ; \; \Lambda_n = \mathrm{diag}\{\sigma_n^2, \cdots, \sigma_n^2\}$$

Properties:

$$\mathrm{U}_s \perp \mathrm{U}_n \quad \Leftrightarrow \quad \mathrm{U}_s^h \cdot \mathrm{U}_n = 0 \quad \Leftrightarrow \quad \mathrm{U}_n^h \cdot \mathrm{U}_s = 0$$

$$U_s^h \cdot U_s = I$$

High-resolution DOA

How obtain DOA's from this?

High-resolution DOA

How obtain DOA's from this?

$$
\begin{aligned}
\mathsf{R}_x &= \mathsf{U}_s \Lambda_{s,n} \mathsf{U}_s^h + \mathsf{U}_n \Lambda_n \mathsf{U}_n^h \\
\mathsf{R}_x &= \mathsf{A}\mathsf{R}_s \mathsf{A}^h + \sigma_n^2 \mathsf{I}
\end{aligned}
$$

High-resolution DOA

How obtain DOA's from this?

$$\begin{aligned} \mathsf{R}_x &= \mathsf{U}_s \Lambda_{s,n} \mathsf{U}_s^h + \mathsf{U}_n \Lambda_n \mathsf{U}_n^h \quad \Rightarrow \quad \mathsf{R}_x \mathsf{U}_n = \sigma_n^2 \mathsf{U}_n \\ \mathsf{R}_x &= \mathsf{A} \mathsf{R}_s \mathsf{A}^h + \sigma_n^2 \mathsf{I} \end{aligned}$$

High-resolution DOA

How obtain DOA's from this?

$$
\begin{aligned}
\mathrm{R}_x &= \mathrm{U}_s \Lambda_{s,n} \mathrm{U}_s^h + \mathrm{U}_n \Lambda_n \mathrm{U}_n^h \quad \Rightarrow \quad \mathrm{R}_x \mathrm{U}_n = \sigma_n^2 \mathrm{U}_n \\
\mathrm{R}_x &= \mathrm{A}\mathrm{R}_s\mathrm{A}^h + \sigma_n^2 \mathrm{I} \quad \Rightarrow \quad \mathrm{R}_x \mathrm{U}_n = \mathrm{A}\mathrm{R}_s\mathrm{A}^h \mathrm{U}_n + \sigma_n^2 \mathrm{U}_n
\end{aligned}
$$

High-resolution DOA

How obtain DOA's from this?

$$\begin{aligned} \mathrm{R}_x &= \mathrm{U}_s \Lambda_{s,n} \mathrm{U}_s^h + \mathrm{U}_n \Lambda_n \mathrm{U}_n^h \quad \Rightarrow \quad \mathrm{R}_x \mathrm{U}_n = \sigma_n^2 \mathrm{U}_n \\ \mathrm{R}_x &= \mathrm{A} \mathrm{R}_s \mathrm{A}^h + \sigma_n^2 \mathrm{I} \quad \Rightarrow \quad \mathrm{R}_x \mathrm{U}_n = \mathrm{A} \mathrm{R}_s \mathrm{A}^h \mathrm{U}_n + \sigma_n^2 \mathrm{U}_n \end{aligned}$$

$$\Rightarrow \mathrm{A} \mathrm{R}_s \mathrm{A}^h \mathrm{U}_n = 0.$$

High-resolution DOA

How obtain DOA's from this?

$$\begin{aligned} R_x &= U_s \Lambda_{s,n} U_s^h + U_n \Lambda_n U_n^h \quad \Rightarrow \quad R_x U_n = \sigma_n^2 U_n \\ R_x &= A R_s A^h + \sigma_n^2 I \quad \Rightarrow \quad R_x U_n = A R_s A^h U_n + \sigma_n^2 U_n \end{aligned}$$

$\Rightarrow A R_s A^h U_n = 0.$ Together with $A R_s$ full rank

$$\Rightarrow \quad A^h U_n = 0 \quad \Leftrightarrow \quad U_n^h A = 0$$

High-resolution DOA

How obtain DOA's from this?

$$\begin{aligned} R_x &= U_s \Lambda_{s,n} U_s^h + U_n \Lambda_n U_n^h \quad \Rightarrow \quad R_x U_n = \sigma_n^2 U_n \\ R_x &= A R_s A^h + \sigma_n^2 I \quad \Rightarrow \quad R_x U_n = A R_s A^h U_n + \sigma_n^2 U_n \end{aligned}$$

$\Rightarrow A R_s A^h U_n = 0$. Together with AR_s full rank

$$\Rightarrow \quad A^h U_n = 0 \quad \Leftrightarrow \quad U_n^h A = 0$$

Result: Obtain desired DOA's by solving θ from:

$$\boxed{\begin{aligned} \underline{u}_i^h \cdot \underline{a}(\theta_p) = 0 \quad \text{for} \quad & \underline{u}_i \in U_n = \{u_{P+1}, \cdots, \underline{u}_J\} \\ & \underline{a}(\theta_p) \in A = \{\underline{a}(\theta_1), \cdots, \underline{a}(\theta_P)\} \end{aligned}}$$

High-resolution DOA

Use this result in "Spectral MUSIC" cost function:

$$C_{SM}(\theta) = \sum_{\underline{u}_i \in U_n} \left| \underline{u}_i^h \underline{a}(\theta) \right|^2 = \underline{a}^h(\theta) \left(\sum_{\underline{u}_i \in U_n} \underline{u}_i \underline{u}_i^h \right) \underline{a}(\theta)$$

High-resolution DOA

Use this result in "Spectral MUSIC" cost function:

$$
\begin{aligned}
C_{SM}(\theta) &= \sum_{\underline{u}_i \in \mathrm{U}_n} \left| \underline{u}_i^h \underline{a}(\theta) \right|^2 = \underline{a}^h(\theta) \left(\sum_{\underline{u}_i \in \mathrm{U}_n} \underline{u}_i \underline{u}_i^h \right) \underline{a}(\theta) \\
&= \underline{a}^h(\theta) \left(\mathrm{U}_n \mathrm{U}_n^h \right) \underline{a}(\theta) = \underline{a}^h(\theta) \left(\mathrm{P}_n \right) \underline{a}(\theta)
\end{aligned}
$$

High-resolution DOA

Use this result in "Spectral MUSIC" cost function:

$$C_{SM}(\theta) = \sum_{\underline{u}_i \in U_n} \left| \underline{u}_i^h \underline{a}(\theta) \right|^2 = \underline{a}^h(\theta) \left(\sum_{\underline{u}_i \in U_n} \underline{u}_i \underline{u}_i^h \right) \underline{a}(\theta)$$

$$= \underline{a}^h(\theta) \left(U_n U_n^h \right) \underline{a}(\theta) = \underline{a}^h(\theta) \left(P_n \right) \underline{a}(\theta)$$

with projection matrix $P_n = U_n U_n^h$

High-resolution DOA

Use this result in "Spectral MUSIC" cost function:

$$\begin{aligned}
C_{SM}(\theta) &= \sum_{\underline{u}_i \in U_n} \left| \underline{u}_i^h \underline{a}(\theta) \right|^2 = \underline{a}^h(\theta) \left(\sum_{\underline{u}_i \in U_n} \underline{u}_i \underline{u}_i^h \right) \underline{a}(\theta) \\
&= \underline{a}^h(\theta) \left(U_n U_n^h \right) \underline{a}(\theta) = \underline{a}^h(\theta) \left(P_n \right) \underline{a}(\theta)
\end{aligned}$$

with projection matrix $P_n = U_n U_n^h$

$P_n \underline{a}(\theta)$ is projection of $\underline{a}(\theta)$ on noise subspace U_n.

High-resolution DOA

Use this result in "Spectral MUSIC" cost function:

$$
\begin{aligned}
C_{SM}(\theta) &= \sum_{\underline{u}_i \in \mathrm{U}_n} \left| \underline{u}_i^h \underline{a}(\theta) \right|^2 = \underline{a}^h(\theta) \left(\sum_{\underline{u}_i \in \mathrm{U}_n} \underline{u}_i \underline{u}_i^h \right) \underline{a}(\theta) \\
&= \underline{a}^h(\theta) \left(\mathrm{U}_n \mathrm{U}_n^h \right) \underline{a}(\theta) = \underline{a}^h(\theta) \left(\mathrm{P}_n \right) \underline{a}(\theta)
\end{aligned}
$$

with projection matrix $\mathrm{P}_n = \mathrm{U}_n \mathrm{U}_n^h$

$\mathrm{P}_n \underline{a}(\theta)$ is projection of $\underline{a}(\theta)$ on noise subspace U_n.

Conclusion: C_{SM} is innerproduct of $\underline{a}(\theta)$ and projection of $\underline{a}(\theta)$ on U_n

$\boxed{C_{SM} = 0 \;\; \text{only true for DOA's } \theta_p \text{ with } p = 1, \cdots, P}$

High-resolution DOA

Define pseudo-spectrum:

$$P_{SM}(\theta) = \frac{||\underline{\mathsf{a}}(\theta)||^2}{C_{SM}} = \frac{||\underline{\mathsf{a}}(\theta)||^2}{\underline{\mathsf{a}}^h(\theta)\mathsf{U}_n\mathsf{U}_n^h\underline{\mathsf{a}}(\theta)} = \frac{J}{\underline{\mathsf{a}}^h(\theta)\mathsf{P}_n\underline{\mathsf{a}}(\theta)}$$

High-resolution DOA

Define pseudo-spectrum:

$$P_{SM}(\theta) = \frac{||\underline{\mathrm{a}}(\theta)||^2}{C_{SM}} = \frac{||\underline{\mathrm{a}}(\theta)||^2}{\underline{\mathrm{a}}^h(\theta)\mathrm{U}_n\mathrm{U}_n^h\underline{\mathrm{a}}(\theta)} = \frac{J}{\underline{\mathrm{a}}^h(\theta)\mathrm{P}_n\underline{\mathrm{a}}(\theta)}$$

Notes:

- Minimizing $C_{SM}(\theta) \quad \Leftrightarrow \quad$ Maximizing $P_{SM}(\theta)$
- Sharp peaks (high resolution) in vicinity of source DOA's θ_p, $p = 1, \cdots, P$
- In practice: $\hat{\mathrm{R}}_x \Rightarrow \hat{\mathrm{U}}_s$ not completely orthogonal to $\hat{\mathrm{U}}_n$
- P_{SM} averages $J - P$ pseudo spectra of individual noise vectors
- 'Pseudo' in name since no info about real power

High-resolution DOA

Spectral-MUSIC algorithm

High-resolution DOA

Spectral-MUSIC algorithm

1. Compute/ estimate R_x

High-resolution DOA

Spectral-MUSIC algorithm

1. Compute/ estimate R_x
2. Compute EVD of R_x and split signal- noise subspace:

$$R_x = U_x \Lambda_x U_x^h = U_s \Lambda_{s,n} U_s^h + U_n \Lambda_n U_n^h$$

High-resolution DOA

Spectral-MUSIC algorithm

1. Compute/ estimate R_x
2. Compute EVD of R_x and split signal- noise subspace:

$$R_x = U_x \Lambda_x U_x^h = U_s \Lambda_{s,n} U_s^h + U_n \Lambda_n U_n^h$$

3. Compute projection matrix: $P_n = U_n U_n^h$

High-resolution DOA

Spectral-MUSIC algorithm

1. Compute/ estimate R_x
2. Compute EVD of R_x and split signal- noise subspace:

$$\mathrm{R}_x = \mathrm{U}_x \Lambda_x \mathrm{U}_x^h = \mathrm{U}_s \Lambda_{s,n} \mathrm{U}_s^h + \mathrm{U}_n \Lambda_n \mathrm{U}_n^h$$

3. Compute projection matrix: $\mathrm{P}_n = \mathrm{U}_n \mathrm{U}_n^h$
4. Evaluate pseudo spectrum:

$$P_{SM}(\theta) = \frac{J}{\underline{\mathrm{a}}^h(\theta) \mathrm{P}_n \underline{\mathrm{a}}(\theta)}$$

High-resolution DOA

Spectral-MUSIC algorithm

1. Compute/ estimate R_x
2. Compute EVD of R_x and split signal- noise subspace:

$$\mathrm{R}_x = \mathrm{U}_x \Lambda_x \mathrm{U}_x^h = \mathrm{U}_s \Lambda_{s,n} \mathrm{U}_s^h + \mathrm{U}_n \Lambda_n \mathrm{U}_n^h$$

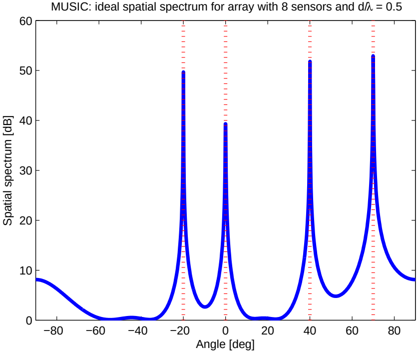
3. Compute projection matrix: $\mathrm{P}_n = \mathrm{U}_n \mathrm{U}_n^h$
4. Evaluate pseudo spectrum:

$$P_{SM}(\theta) = \frac{J}{\underline{\mathrm{a}}^h(\theta)\mathrm{P}_n \underline{\mathrm{a}}(\theta)}$$

5. Source DOA's θ_p for $p = 1, \cdots, P$:
   Locate P sharpest peaks in $P_{SM}(\theta)$

High-resolution DOA

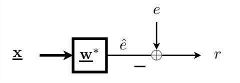
Example: ULA
$d/\lambda = 0.5$; $P = 4$ sources (at -20,0,40 and 70 degrees); $J = 8$ sensors



Note: This example exploits $J - P = 4$ *noise sources*

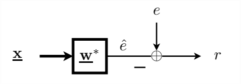
Minimum Mean Squared Error (Wiener)

Minimum Mean Squared Error (Wiener)



$$\underline{w}_{mse} = arg\,\min_{\underline{w}}\{J\}$$
$$J = E\{|r|^2\}$$

Minimum Mean Squared Error (Wiener)

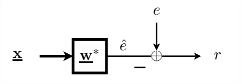


$$\underline{w}_{mse} = argmin_{\underline{w}}\{J\}$$
$$J = E\{|r|^2\}$$

$$\Rightarrow \quad \underline{w}_{mse} = R_x^{-1}\underline{r}_{e^*x} \quad \text{and} \quad J_{min} = E\{|e|^2\} - \underline{r}_{e^*x}^h R_x^{-1}\underline{r}_{e^*x}$$

$$\text{with} \quad R_x = E\{\underline{x}\cdot\underline{x}^h\} \qquad \underline{r}_{e^*x} = E\{\underline{x}\cdot e^*\}$$

Minimum Mean Squared Error (Wiener)



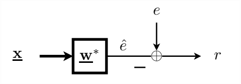
$$\begin{aligned} \underline{w}_{mse} &= \arg\min_{\underline{w}}\{J\} \\ J &= E\{|r|^2\} \end{aligned}$$

$$\Rightarrow \quad \underline{w}_{mse} = R_x^{-1}\underline{r}_{e^*x} \quad \text{and} \quad J_{min} = E\{|e|^2\} - \underline{r}^h_{e^*x} R_x^{-1} \underline{r}_{e^*x}$$

$$\text{with} \quad R_x = E\{\underline{x} \cdot \underline{x}^h\} \qquad \underline{r}_{e^*x} = E\{\underline{x} \cdot e^*\}$$

Necessary knowledge: R_x and $\underline{r}_{e^*x}$ (both from measurements)

Minimum Mean Squared Error (Wiener)



$$\underline{w}_{mse} = \arg\min_{\underline{w}}\{J\}$$
$$J = E\{|r|^2\}$$

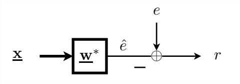
$$\Rightarrow \quad \underline{w}_{mse} = R_x^{-1}\underline{r}_{e^*x} \quad \text{and} \quad J_{min} = E\{|e|^2\} - \underline{r}_{e^*x}^h R_x^{-1}\underline{r}_{e^*x}$$

$$\text{with} \quad R_x = E\{\underline{x}\cdot\underline{x}^h\} \quad \underline{r}_{e^*x} = E\{\underline{x}\cdot e^*\}$$

Necessary knowledge: R_x and $\underline{r}_{e^*x}$ (both from measurements)

Example: ULA, 1 source, narrowband, farfield $\rightarrow \underline{x} = \underline{a}\cdot s + \underline{n}$ and $e = s$

Minimum Mean Squared Error (Wiener)



$$\underline{w}_{mse} = argmin_{\underline{w}}\{J\}$$
$$J = E\{|r|^2\}$$

$$\Rightarrow \quad \underline{w}_{mse} = \mathrm{R}_x^{-1}\underline{r}_{e^*x} \quad \text{and} \quad J_{min} = E\{|e|^2\} - \underline{r}_{e^*x}^h \mathrm{R}_x^{-1}\underline{r}_{e^*x}$$

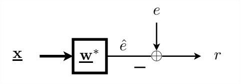
$$\text{with} \quad \mathrm{R}_x = E\{\underline{x}\cdot\underline{x}^h\} \quad \underline{r}_{e^*x} = E\{\underline{x}\cdot e^*\}$$

Necessary knowledge: R_x and $\underline{r}_{e^*x}$ (both from measurements)

Example: ULA, 1 source, narrowband, farfield $\rightarrow \underline{x} = \underline{a}\cdot s + \underline{n}$ and $e = s$

$$\mathrm{R}_x = \left(\underline{a}\cdot\underline{a}^h\right)\cdot\sigma_s^2 + \sigma_n^2\cdot \mathrm{I} \quad \text{and} \quad \underline{r}_{e^*x} = \underline{a}\cdot\sigma_s^2$$

Minimum Mean Squared Error (Wiener)



$$\underline{w}_{mse} = arg\min_{\underline{w}}\{J\}$$
$$J = E\{|r|^2\}$$

$$\Rightarrow \quad \underline{w}_{mse} = R_x^{-1}\underline{r}_{e^*x} \quad \text{and} \quad J_{min} = E\{|e|^2\} - \underline{r}_{e^*x}^h R_x^{-1}\underline{r}_{e^*x}$$

$$\text{with} \quad R_x = E\{\underline{x}\cdot\underline{x}^h\} \quad \underline{r}_{e^*x} = E\{\underline{x}\cdot e^*\}$$

Necessary knowledge: R_x and $\underline{r}_{e^*x}$ (both from measurements)

Example: ULA, 1 source, narrowband, farfield $\rightarrow \underline{x} = \underline{a}\cdot s + \underline{n}$ and $e = s$

$$R_x = \left(\underline{a}\cdot\underline{a}^h\right)\cdot\sigma_s^2 + \sigma_n^2\cdot I \quad \text{and} \quad \underline{r}_{e^*x} = \underline{a}\cdot\sigma_s^2$$

$$\Rightarrow \quad \underline{w}_{mse} = \beta\cdot\underline{a} \quad \text{and} \quad J_{min} = \beta\cdot\sigma_n^2 \quad \text{with} \quad \beta = \frac{(\sigma_s^2/\sigma_n^2)}{1 + J\cdot(\sigma_s^2/\sigma_n^2)}$$

Minimum Mean Squared Error (Wiener)

Thus for ULA, one source, narrowband, farfield:

- $\underline{w}_{mse} = \beta \cdot \underline{a}$, which is equivalent to matched filter result, which maximizes SNR
- $J_{min} \approx \frac{1}{J} \cdot \sigma_n^2 \Rightarrow$ SNR impoved approx. by factor J

Minimum Mean Squared Error (Wiener)

Thus for ULA, one source, narrowband, farfield:

- $\underline{w}_{mse} = \beta \cdot \underline{a}$, which is equivalent to matched filter result, which maximizes SNR
- $J_{min} \approx \frac{1}{J} \cdot \sigma_n^2 \Rightarrow$ SNR impoved approx. by factor J

Conclusion MMSE

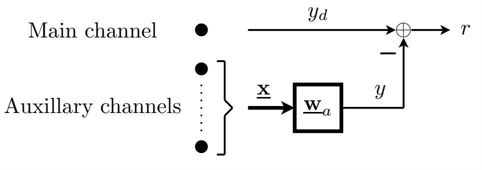
\+ Simple

\+ Direction of desired signal may be unknown

\- Must generate reference signal

Multiple Sidelobe Canceller (Applebaum)

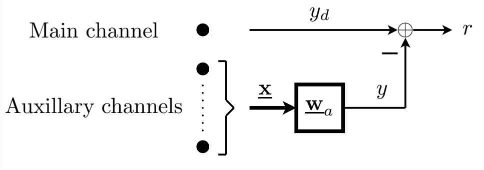
Use auxillary channel to cancel interference in main channel



$$\underline{w}_{opt} = arg \min_{\underline{w}_a} \{ |y_d - \underline{w}_a^h \cdot \underline{x}|^2 \}$$

Multiple Sidelobe Canceller (Applebaum)

Use auxillary channel to cancel interference in main channel

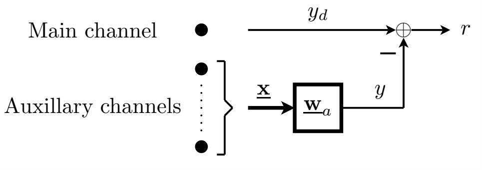


$$\underline{w}_{opt} = arg \min_{\underline{w}_a} \{ |y_d - \underline{w}_a^h \cdot \underline{x}|^2 \}$$

Main assumption MSC: Interference assumed to be present in both main and auxillary channels. Desired signal strongly present in main channel, but **below noise level** in auxillary channels

Multiple Sidelobe Canceller (Applebaum)

Use auxillary channel to cancel interference in main channel



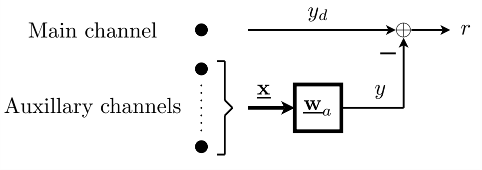
$$\underline{w}_{opt} = arg \min_{\underline{w}_a} \{ |y_d - \underline{w}_a^h \cdot \underline{x}|^2 \}$$

Main assumption MSC: Interference assumed to be present in both main and auxillary channels. Desired signal strongly present in main channel, but **below noise level** in auxillary channels

$$\Rightarrow \boxed{\underline{w}_{opt} = R_x^{-1} \cdot \underline{r}_{xy_d^*}} \text{ and } \boxed{P_{out} = \sigma_{y_d}^2 - \underline{r}_{xy_d}^h R_x^{-1} \underline{r}_{xy_d}}$$

Multiple Sidelobe Canceller (Applebaum)

Use auxillary channel to cancel interference in main channel



$$\underline{w}_{opt} = arg\min_{\underline{w}_a}\{|y_d - \underline{w}_a^h \cdot \underline{x}|^2\}$$

Main assumption MSC: Interference assumed to be present in both main and auxillary channels. Desired signal strongly present in main channel, but **below noise level** in auxillary channels

$$\Rightarrow \boxed{\underline{w}_{opt} = R_x^{-1} \cdot \underline{r}_{xy_d^*}} \text{ and } \boxed{P_{out} = \sigma_{y_d}^2 - \underline{r}_{xy_d}^h R_x^{-1} \underline{r}_{xy_d}}$$

Conclusion MSC: Simple, but requires desired signal below noise level in auxillay channels, otherwise ...

Linear Constrained Minimum Variance (Frost)

Previous methods may be unsatisfactory e.g. if desired signal is of unknown strength or is always present $\rightarrow$

- **MSC:** Signal cancelling
- **Max SINR:** Needs signal and noise covariance estimate
- **MMSE:** Lack of knowledge reference signal

Linear Constrained Minimum Variance (Frost)

Previous methods may be unsatisfactory e.g. if desired signal is of unknown strength or is always present $\rightarrow$

- **MSC:** Signal cancelling
- **Max SINR:** Needs signal and noise covariance estimate
- **MMSE:** Lack of knowledge reference signal

Design philosophy LCMV:

Design weight vector by minimizing average output power subject to M constraints that filter response remains constant at some specific frequencies of interest

Linear Constrained Minimum Variance

Average output power:

$$P_y = E\{|y|^2\} = \underline{\mathrm{w}}^h \cdot E\{\underline{\mathrm{x}}\underline{\mathrm{x}}^h\} \cdot \underline{\mathrm{w}} = \underline{\mathrm{w}}^h \cdot \mathrm{R}_x \cdot \underline{\mathrm{w}}$$

Linear Constrained Minimum Variance

Average output power:

$$P_y = E\{|y|^2\} = \underline{\mathrm{w}}^h \cdot E\{\underline{\mathrm{x}}\underline{\mathrm{x}}^h\} \cdot \underline{\mathrm{w}} = \underline{\mathrm{w}}^h \cdot \mathrm{R}_x \cdot \underline{\mathrm{w}}$$

$M < J$ linear independent constraints:

$\mathrm{C}^h\underline{\mathrm{w}} = \underline{\mathrm{r}}_d$ with $J \times M$ constraint matrix C

Linear Constrained Minimum Variance

Average output power:

$$P_y = E\{|y|^2\} = \underline{\mathrm{w}}^h \cdot E\{\underline{\mathrm{x}}\underline{\mathrm{x}}^h\} \cdot \underline{\mathrm{w}} = \underline{\mathrm{w}}^h \cdot \mathrm{R}_x \cdot \underline{\mathrm{w}}$$

$M < J$ linear independent constraints:

$\mathrm{C}^h \underline{\mathrm{w}} = \underline{\mathrm{r}}_d$ with $J \times M$ constraint matrix C

Error criterion:

$$\min_{\underline{\mathrm{w}}} \{P_y\} = \min_{\underline{\mathrm{w}}} \{\underline{\mathrm{w}}^h \cdot \mathrm{R}_x \cdot \underline{\mathrm{w}}\} \text{ subject to } \mathrm{C}^h \cdot \underline{\mathrm{w}} = \underline{\mathrm{r}}_d$$

Linear Constrained Minimum Variance

Average output power:

$$P_y = E\{|y|^2\} = \underline{\mathrm{w}}^h \cdot E\{\underline{\mathrm{x}}\underline{\mathrm{x}}^h\} \cdot \underline{\mathrm{w}} = \underline{\mathrm{w}}^h \cdot \mathrm{R}_x \cdot \underline{\mathrm{w}}$$

$M < J$ linear independent constraints:

$\mathrm{C}^h \underline{\mathrm{w}} = \underline{\mathrm{r}}_d$ with $J \times M$ constraint matrix C

Error criterion:

$$\min_{\underline{\mathrm{w}}}\{P_y\} = \min_{\underline{\mathrm{w}}}\{\underline{\mathrm{w}}^h \cdot \mathrm{R}_x \cdot \underline{\mathrm{w}}\} \text{ subject to } \mathrm{C}^h \cdot \underline{\mathrm{w}} = \underline{\mathrm{r}}_d$$

Solution via Lagrange multipliers:

$$J = \underline{\mathrm{w}}^h \cdot \mathrm{R}_x \cdot \underline{\mathrm{w}} + \underline{\lambda}\left(\mathrm{C}^h \underline{\mathrm{w}} - \underline{\mathrm{r}}_d\right)$$

Linear Constrained Minimum Variance

Average output power:

$$P_y = E\{|y|^2\} = \underline{w}^h \cdot E\{\underline{x}\underline{x}^h\} \cdot \underline{w} = \underline{w}^h \cdot \mathrm{R}_x \cdot \underline{w}$$

$M < J$ linear independent constraints:

$\mathrm{C}^h\underline{w} = \underline{r}_d$ with $J \times M$ constraint matrix C

Error criterion:

$$\min_{\underline{w}}\{P_y\} = \min_{\underline{w}}\{\underline{w}^h \cdot \mathrm{R}_x \cdot \underline{w}\} \text{ subject to } \mathrm{C}^h \cdot \underline{w} = \underline{r}_d$$

Solution via Lagrange multipliers:

$$J = \underline{w}^h \cdot \mathrm{R}_x \cdot \underline{w} + \underline{\lambda}\left(\mathrm{C}^h\underline{w} - \underline{r}_d\right)$$

Solution similar to results in part I:

$$\boxed{\underline{w} = \mathrm{R}_x^{-1}\mathrm{C}\left(\mathrm{C}^h\mathrm{R}_x^{-1}\mathrm{C}\right)^{-1}\underline{r}_d} \text{ and } \boxed{P_y = \underline{r}_d^h\left(\mathrm{C}^h\mathrm{R}_x^{-1}\mathrm{C}\right)^{-1}\underline{r}_d}$$

Linear Constrained Minimum Variance

Conclusions LCMV:

Linear Constrained Minimum Variance

Conclusions LCMV:

+ High resolution

Linear Constrained Minimum Variance

Conclusions LCMV:

+ High resolution
+ LCMV controls spectral leakage, in contrast to conventional (e.g. null-steering) methods

Linear Constrained Minimum Variance

Conclusions LCMV:

+ High resolution
+ LCMV controls spectral leakage, in contrast to conventional (e.g. null-steering) methods
+ If there are interferences present, LCMV tends to null them out

Linear Constrained Minimum Variance

Conclusions LCMV:

+ High resolution
+ LCMV controls spectral leakage, in contrast to conventional (e.g. null-steering) methods
+ If there are interferences present, LCMV tends to null them out

- Sensitive due to inverse correlation matrix

Minimum Variance Distortionless Response

MVDR:

$$\min_{\underline{w}}\{P_y\} = \min_{\underline{w}}\{E\{|y|^2\}\ \text{subject to}\ \underline{w}^h \cdot \underline{a} = 1$$

Minimum Variance Distortionless Response

MVDR:

$$\min_{\underline{w}}\{P_y\} = \min_{\underline{w}}\{E\{|y|^2\}\} \text{ subject to } \underline{w}^h \cdot \underline{a} = 1$$

$$\Rightarrow \boxed{\underline{w} = \frac{R^{-1}\underline{a}}{\underline{a}^h R^{-1}\underline{a}}} \text{ and } \boxed{P_y = \frac{1}{\underline{a}^h R^{-1}\underline{a}}}$$

Minimum Variance Distortionless Response

MVDR:

$$\min_{\underline{w}}\{P_y\} = \min_{\underline{w}}\{E\{|y|^2\}\} \text{ subject to } \underline{w}^h \cdot \underline{a} = 1$$

$$\Rightarrow \boxed{\underline{w} = \frac{R^{-1}\underline{a}}{\underline{a}^h R^{-1}\underline{a}}} \text{ and } \boxed{P_y = \frac{1}{\underline{a}^h R^{-1}\underline{a}}}$$

Notes MVDR (=Capon):

- MVDR is special case of LCMV with $r_d = 1$ and $C = \underline{a}$
- MVDR is max SNR if $R = \sigma_s^2 \underline{a}\underline{a}^h + R_n$

Minimum Variance Distortionless Response

MVDR:

$$\min_{\underline{w}}\{P_y\} = \min_{\underline{w}}\{E\{|y|^2\}\} \text{ subject to } \underline{w}^h \cdot \underline{a} = 1$$

$$\Rightarrow \boxed{\underline{w} = \frac{R^{-1}\underline{a}}{\underline{a}^h R^{-1}\underline{a}}} \text{ and } \boxed{P_y = \frac{1}{\underline{a}^h R^{-1}\underline{a}}}$$

Notes MVDR (=Capon):

- MVDR is special case of LCMV with $r_d = 1$ and $C = \underline{a}$
- MVDR is max SNR if $R = \sigma_s^2 \underline{a}\underline{a}^h + R_n$

Conclusions MVDR/ LCMV:

\+ Flexible and general constraints possible

\- Computation constraint weight vector not trivial

\- As signal extractor: sensitive to errors in DOA

\- Problems with correlated signals

Generalized Sidelobe Canceller (Griffith-Jim)

GSC: Alternative formulation of LCMV, illustrates relationship between MSC and LCMV. Mechanism to change constrained minimization problem in unconstrained one

Generalized Sidelobe Canceller (Griffith-Jim)

GSC: Alternative formulation of LCMV, illustrates relationship between MSC and LCMV. Mechanism to change constrained minimization problem in unconstrained one

With M independent constraints and J $(< M)$ weights $\Rightarrow$
$C^h \underline{w} = \underline{r}_d$

Generalized Sidelobe Canceller (Griffith-Jim)

GSC: Alternative formulation of LCMV, illustrates relationship between MSC and LCMV. Mechanism to change constrained minimization problem in unconstrained one

With M independent constraints and J $(< M)$ weights $\Rightarrow$
$\mathrm{C}^h \underline{\mathrm{w}} = \underline{\mathrm{r}}_d$

Construct full rank $J \times J$ matrix $\mathrm{B} = (\mathrm{C}|\mathrm{T})$
B has J independent columns; Rank $J \times M$ matrix C is M and Rank $J \times (J - M)$ matrix T is $J - M$

Generalized Sidelobe Canceller (Griffith-Jim)

GSC: Alternative formulation of LCMV, illustrates relationship between MSC and LCMV. Mechanism to change constrained minimization problem in unconstrained one

With M independent constraints and J ($< M$) weights $\Rightarrow$
$\mathrm{C}^h \underline{\mathrm{w}} = \underline{\mathrm{r}}_d$

Construct full rank $J \times J$ matrix $\mathrm{B} = (\mathrm{C}|\mathrm{T})$
B has J independent columns; Rank $J \times M$ matrix C is M and Rank $J \times (J - M)$ matrix T is $J - M$

Any $\underline{\mathrm{w}} \in J$ dimensional space spanned by columns B:

$$\underline{\mathrm{w}} = \mathrm{C} \cdot \underline{\mathrm{v}} - \mathrm{T} \cdot \underline{\mathrm{w}}_a \;\; \text{with} \;\; \mathrm{C} \cdot \underline{\mathrm{v}} = \underline{\mathrm{w}}_c \;\; \text{weights belong to constraint}$$

$$\text{Constraint} \;\; \Rightarrow \;\; \mathrm{C}^h \underline{\mathrm{w}} = \mathrm{C}^h \mathrm{C} \underline{\mathrm{v}} - \mathrm{C}^h \mathrm{T} \underline{\mathrm{w}}_a = \underline{\mathrm{r}}_d$$

Generalized Sidelobe Canceller (Griffith-Jim)

GSC: Alternative formulation of LCMV, illustrates relationship between MSC and LCMV. Mechanism to change constrained minimization problem in unconstrained one

With M independent constraints and J ($< M$) weights $\Rightarrow$ $C^h\underline{w} = \underline{r}_d$

Construct full rank $J \times J$ matrix $B = (C|T)$
B has J independent columns; Rank $J \times M$ matrix C is M and Rank $J \times (J - M)$ matrix T is $J - M$

Any $\underline{w} \in J$ dimensional space spanned by columns B:

$$\underline{w} = C \cdot \underline{v} - T \cdot \underline{w}_a \;\text{ with }\; C \cdot \underline{v} = \underline{w}_c \;\text{ weights belong to constraint}$$

$$\text{Constraint} \;\Rightarrow\; C^h\underline{w} = C^hC\underline{v} - C^hT\underline{w}_a = \underline{r}_d$$

$$\text{Construct T such that } C^hT = 0 \Rightarrow \underline{v} = \left(C^hC\right)^{-1}\underline{r}_d$$

Generalized Sidelobe Canceller (Griffith-Jim)

GSC: Alternative formulation of LCMV, illustrates relationship between MSC and LCMV. Mechanism to change constrained minimization problem in unconstrained one

With M independent constraints and J ($< M$) weights $\Rightarrow$ $\mathrm{C}^h \underline{\mathrm{w}} = \underline{\mathrm{r}}_d$

Construct full rank $J \times J$ matrix $\mathrm{B} = (\mathrm{C}|\mathrm{T})$
B has J independent columns; Rank $J \times M$ matrix C is M and Rank $J \times (J - M)$ matrix T is $J - M$

Any $\underline{\mathrm{w}} \in J$ dimensional space spanned by columns B:

$$\underline{\mathrm{w}} = \mathrm{C} \cdot \underline{\mathrm{v}} - \mathrm{T} \cdot \underline{\mathrm{w}}_a \ \text{ with } \ \mathrm{C} \cdot \underline{\mathrm{v}} = \underline{\mathrm{w}}_c \ \text{ weights belong to constraint}$$

$$\text{Constraint} \ \Rightarrow \ \mathrm{C}^h \underline{\mathrm{w}} = \mathrm{C}^h \mathrm{C} \underline{\mathrm{v}} - \mathrm{C}^h \mathrm{T} \underline{\mathrm{w}}_a = \underline{\mathrm{r}}_d$$

$$\text{Construct T such that } \mathrm{C}^h \mathrm{T} = 0 \Rightarrow \underline{\mathrm{v}} = \left(\mathrm{C}^h \mathrm{C}\right)^{-1} \underline{\mathrm{r}}_d$$

$$\underline{\mathrm{w}}_c = \mathrm{C} \cdot \underline{\mathrm{v}} = \boxed{\mathrm{C} \cdot \left(\mathrm{C}^h \mathrm{C}\right)^{-1} \underline{\mathrm{r}}_d}$$

Generalized Sidelobe Canceller

Possible solutions for $\underline{w}_a$:

With $\underline{w} = \underline{w}_c - T \cdot \underline{w}_a$ we can write constrained optimization:

$$\min_{\underline{w}} \{P_y\} = \min_{\underline{w}} \{\underline{w}^h R \underline{w}\} \;\text{ s.t. }\; C^h \cdot \underline{w} = \underline{r}_d$$

Generalized Sidelobe Canceller

Possible solutions for $\underline{w}_a$:

With $\underline{w} = \underline{w}_c - T \cdot \underline{w}_a$ we can write constrained optimization:

$$\min_{\underline{w}}\{P_y\} = \min_{\underline{w}}\{\underline{w}^h R \underline{w}\} \text{ s.t. } C^h \cdot \underline{w} = \underline{r}_d$$

to following unconstrained optimization:

$$\min_{\underline{w}_a}\left\{(\underline{w}_c - T\underline{w}_a)^h R\,(\underline{w}_c - T\underline{w}_a)\right\}$$

Generalized Sidelobe Canceller

Possible solutions for $\underline{w}_a$:

With $\underline{w} = \underline{w}_c - T \cdot \underline{w}_a$ we can write constrained optimization:

$$\min_{\underline{w}}\{P_y\} = \min_{\underline{w}}\{\underline{w}^h R \underline{w}\} \;\text{ s.t. }\; C^h \cdot \underline{w} = \underline{r}_d$$

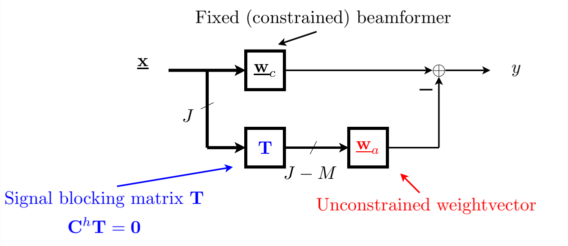
to following unconstrained optimization:

$$\min_{\underline{w}_a}\left\{(\underline{w}_c - T\underline{w}_a)^h R\,(\underline{w}_c - T\underline{w}_a)\right\}$$

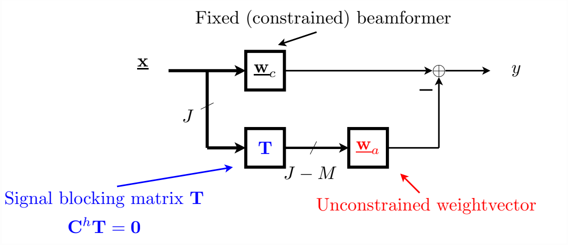
$$\frac{d}{d\underline{w}_a} = \underline{0} \;\Rightarrow\; -2T^h R \underline{w}_c + 2T^h R T \underline{w}_a = \underline{0} \;\Rightarrow$$

$$\boxed{\underline{w}_a = \left(T^h R T\right)^{-1} T^h R \underline{w}_c}$$

Generalized Sidelobe Canceller



Generalized Sidelobe Canceller



Note: Blocking matrix can be constructed by any orthogonalization procedure (e.g. Gramm- Schmidt or QR- decomposition)

Generalized Sidelobe Canceller

Example: $\underline{r}_d = 1$ in steering direction θ_0

Generalized Sidelobe Canceller

Example: $\underline{r}_d = 1$ in steering direction θ_0

$$C = \underline{a}(\omega_0, \theta_0) = (1, e^{-j\phi_0}, e^{-j2\phi_0}, \cdots, e^{-j(J-1)\phi_0})^t$$

with $\phi_0 = \omega_0 \frac{d \sin(\theta_0)}{c}$

Generalized Sidelobe Canceller

Example: $\underline{r}_d = 1$ in steering direction θ_0

$$C = \underline{a}(\omega_0, \theta_0) = (1, e^{-j\phi_0}, e^{-j2\phi_0}, \cdots, e^{-j(J-1)\phi_0})^t$$

with $\phi_0 = \omega_0 \frac{d \sin(\theta_0)}{c} \quad \Rightarrow$

$$\underline{w}_c = C \left(C^h C \right)^{-1} \underline{r}_d = \underline{a} \left(\underline{a}^h \underline{a} \right)^{-1} (1) = \frac{1}{J} \underline{a}(\omega_0, \theta_0)$$

Generalized Sidelobe Canceller

Example: $\underline{\mathrm{r}}_d = 1$ in steering direction θ_0

$$\mathrm{C} = \underline{\mathrm{a}}(\omega_0, \theta_0) = (1, \mathrm{e}^{-\mathrm{j}\phi_0}, \mathrm{e}^{-\mathrm{j}2\phi_0}, \cdots, \mathrm{e}^{-\mathrm{j}(J-1)\phi_0})^t$$

with $\phi_0 = \omega_0 \frac{d \sin(\theta_0)}{c} \quad \Rightarrow$

$$\underline{\mathrm{w}}_c = \mathrm{C}\left(\mathrm{C}^h \mathrm{C}\right)^{-1} \underline{\mathrm{r}}_d = \underline{\mathrm{a}}\left(\underline{\mathrm{a}}^h \underline{\mathrm{a}}\right)^{-1}(1) = \frac{1}{J}\underline{\mathrm{a}}(\omega_0, \theta_0)$$

Beampattern of this fixed filter part:

$$|r(\omega, \theta)| = |\underline{\mathrm{w}}_c^h \cdot \underline{\mathrm{a}}(\omega, \theta)| = \left| \frac{1}{J} \sum_{i=1}^{J} \mathrm{e}^{\mathrm{j}(i-1)\frac{d}{c}(\omega_0 \sin(\theta_0) - \omega \sin(\theta))} \right|$$

Generalized Sidelobe Canceller

Example: $\underline{r}_d = 1$ in steering direction θ_0

$$C = \underline{a}(\omega_0, \theta_0) = (1, e^{-j\phi_0}, e^{-j2\phi_0}, \cdots, e^{-j(J-1)\phi_0})^t$$

with $\phi_0 = \omega_0 \frac{d \sin(\theta_0)}{c} \quad \Rightarrow$

$$\underline{w}_c = C \left(C^h C \right)^{-1} \underline{r}_d = \underline{a} \left(\underline{a}^h \underline{a} \right)^{-1} (1) = \frac{1}{J} \underline{a}(\omega_0, \theta_0)$$

Beampattern of this fixed filter part:

$$|r(\omega, \theta)| = |\underline{w}_c^h \cdot \underline{a}(\omega, \theta)| = \left| \frac{1}{J} \sum_{i=1}^{J} e^{j(i-1)\frac{d}{c}(\omega_0 \sin(\theta_0) - \omega \sin(\theta))} \right|$$

For $\omega = \omega_0 = 2\pi f_0 = 2\pi \frac{c}{\lambda_0}$ beampattern becomes:

$$\boxed{|r(\omega_0, \theta)| \quad = \quad \frac{1}{J} \frac{\sin\left(J\pi \frac{d}{\lambda_0} [\sin(\theta_0) - \sin(\theta)] \right)}{\sin\left(\pi \frac{d}{\lambda_0} [\sin(\theta_0) - \sin(\theta)] \right)}}$$

Generalized Sidelobe Canceller

Furthermore construction of blocking matrix:

$$\mathsf{C}^h\mathsf{T} = 0 \quad \Rightarrow \quad \underline{\mathsf{a}}^h\mathsf{T} = 0 \ \text{ with } \ J \times (J-1) \text{ matrix } \mathsf{T}$$

Generalized Sidelobe Canceller

Furthermore construction of blocking matrix:

$$\mathrm{C}^h \mathrm{T} = 0 \quad \Rightarrow \quad \underline{\mathrm{a}}^h \mathrm{T} = 0 \;\text{ with }\; J \times (J-1) \text{ matrix } \mathrm{T}$$

$$\text{E.g. } \mathrm{T} = \begin{pmatrix} -1 & -1 & \cdots & -1 \\ e^{-j\phi_0} & 0 \cdots & 0 & \\ 0 & e^{-j2\cdot\phi_0} & \cdots & 0 \\ 0 & 0 & \cdots & e^{-j(J-1)\cdot\phi_0} \end{pmatrix}$$

Generalized Sidelobe Canceller

Furthermore construction of blocking matrix:

$$\mathsf{C}^h\mathsf{T} = 0 \quad \Rightarrow \quad \underline{\mathsf{a}}^h\mathsf{T} = 0 \;\text{ with }\; J \times (J-1) \text{ matrix } \mathsf{T}$$

$$\text{E.g. } \mathsf{T} = \begin{pmatrix} -1 & -1 & \cdots & -1 \\ e^{-j\phi_0} & 0\cdots & 0 & \\ 0 & e^{-j2\cdot\phi_0} & \cdots & 0 \\ 0 & 0 & \cdots & e^{-j(J-1)\cdot\phi_0} \end{pmatrix}$$

Thus for m^{th} column (with $m = 1, 2, \cdots, J-1$)

$$\underline{\mathsf{t}}_m = \left(-1, 0, \cdots, 0, e^{-jm\phi_0}, 0, \cdots, 0\right)^t$$

with beampattern $|r_m| = |\underline{\mathsf{t}}_m^h \cdot \underline{\mathsf{a}}(\omega_0, \theta)|$

Generalized Sidelobe Canceller

Furthermore construction of blocking matrix:

$$\mathrm{C}^h\mathrm{T} = 0 \;\;\Rightarrow\;\; \underline{\mathrm{a}}^h\mathrm{T} = 0 \;\text{ with }\; J \times (J-1) \text{ matrix T}$$

$$\text{E.g. } \mathrm{T} = \begin{pmatrix} -1 & -1 & \cdots & -1 \\ \mathrm{e}^{-\mathrm{j}\phi_0} & 0 \cdots & 0 & \\ 0 & \mathrm{e}^{-\mathrm{j}2\cdot\phi_0} & \cdots & 0 \\ 0 & 0 & \cdots & \mathrm{e}^{-\mathrm{j}(J-1)\cdot\phi_0} \end{pmatrix}$$

Thus for m^{th} column (with $m = 1, 2, \cdots, J-1$)

$$\underline{\mathrm{t}}_m = \left(-1, 0, \cdots, 0, \mathrm{e}^{-\mathrm{j}m\phi_0}, 0, \cdots, 0\right)^t$$

with beampattern $|r_m| = |\underline{\mathrm{t}}_m^h \cdot \underline{\mathrm{a}}(\omega_0, \theta)| \;\Rightarrow$ for $d/\lambda = 1/2$:

$$\boxed{|r_m(\omega, \theta)| \;=\; 2\left|\sin\left(\frac{1}{2} m\pi \left[\sin(\theta_0) - \sin(\theta)\right]\right)\right|}$$

Generalized Sidelobe Canceller

Furthermore construction of blocking matrix:

$$\mathrm{C}^h\mathrm{T} = 0 \quad \Rightarrow \quad \underline{\mathrm{a}}^h\mathrm{T} = 0 \;\text{ with }\; J \times (J-1) \text{ matrix T}$$

$$\text{E.g. } \mathrm{T} = \begin{pmatrix} -1 & -1 & \cdots & -1 \\ e^{-j\phi_0} & 0 \cdots & 0 & \\ 0 & e^{-j2\cdot\phi_0} & \cdots & 0 \\ 0 & 0 & \cdots & e^{-j(J-1)\cdot\phi_0} \end{pmatrix}$$

Thus for m^{th} column (with $m = 1, 2, \cdots, J-1$)

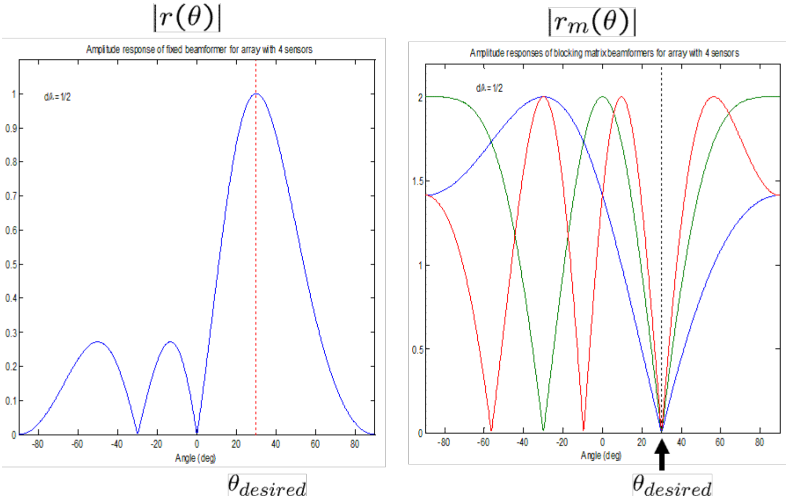
$$\underline{\mathrm{t}}_m = \left(-1, 0, \cdots, 0, e^{-jm\phi_0}, 0, \cdots, 0\right)^t$$

with beampattern $|r_m| = |\underline{\mathrm{t}}_m^h \cdot \underline{\mathrm{a}}(\omega_0, \theta)| \;\Rightarrow$ for $d/\lambda = 1/2$:

$$\boxed{|r_m(\omega, \theta)| \;=\; 2\left|\sin\left(\frac{1}{2} m\pi \left[\sin(\theta_0) - \sin(\theta)\right]\right)\right|}$$

(= amplitude response of m^{th} column blocking matrix)

Generalized Sidelobe Canceller



Adaptive Array Signal Processing

Why adaptive?

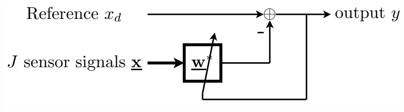
In most previous results knowledge of SOS needed.
These statistics are usually unknown and/or time-varying
With ergodic assumption $\rightarrow$ SOS can be estimated from available data samples $\rightarrow$ adaptive solutions!

Adaptive Array Signal Processing

Why adaptive?

In most previous results knowledge of SOS needed.
These statistics are usually unknown and/or time-varying
With ergodic assumption $\rightarrow$ SOS can be estimated from available data samples $\rightarrow$ adaptive solutions!

General adaptive array structure:



Wiener: $\underline{w}_{opt} = arg \min_{\underline{w}} \left\{ E\{|y|^2\} \right\}$

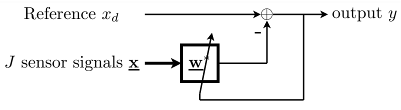
$$\Rightarrow \qquad \underline{w}_{opt} = R_x^{-1} \cdot \underline{r}_{xx_d^*}$$

Adaptive Array Signal Processing

Why adaptive?

In most previous results knowledge of SOS needed.
These statistics are usually unknown and/or time-varying
With ergodic assumption $\rightarrow$ SOS can be estimated from available data samples $\rightarrow$ adaptive solutions!

General adaptive array structure:



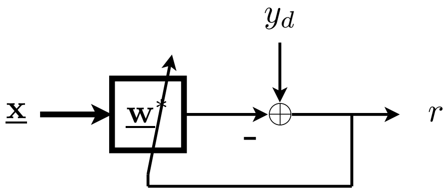
Wiener: $\underline{w}_{opt} = arg \min_{\underline{w}} \left\{ E\{|y|^2\} \right\}$

$$\Rightarrow \qquad \underline{w}_{opt} = R_x^{-1} \cdot \underline{r}_{xx_d^*}$$

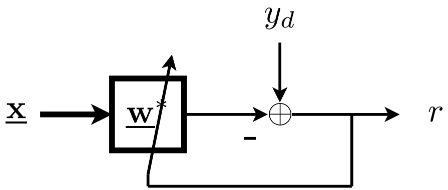
LMS update rule : $\underline{w}[k+1] = \underline{w}[k] + 2\alpha \underline{x}[k] y^*[k]$

Final value : $\lim_{k \to \infty} E\{\underline{w}[k]\} = \underline{w}_{opt}$

Adaptive MMSE (Wiener)



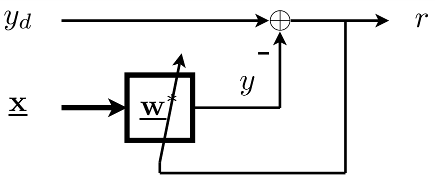
Adaptive MMSE (Wiener)



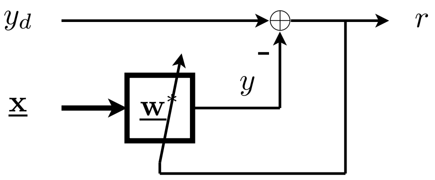
Optimal Wiener : $\underline{w}_{opt} = arg \min_{\underline{w}} \left\{ E\{|r|^2\} \right\} = R_x^{-1} \cdot \underline{r}_{xy_d^*}$

LMS update rule : $\underline{w}[k+1] = \underline{w}[k] + 2\alpha \underline{x}[k] r^*[k]$

Adaptive MSC



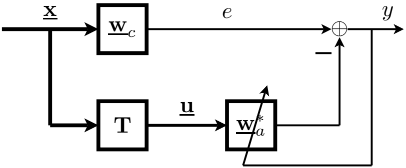
Adaptive MSC



Optimal Wiener : $\underline{w}_{opt} = arg \min_{\underline{w}} \left\{ E\{|r|^2\} \right\} = R_x^{-1} \cdot \underline{r}_{xy_d^*}$

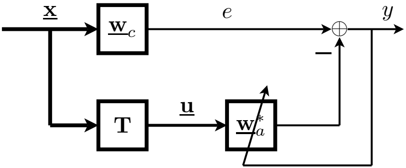
LMS update rule : $\underline{w}[k+1] = \underline{w}[k] + 2\alpha \underline{x}[k] r^*[k]$

Adaptive GSC



$$
\begin{aligned}
\underline{w}_c &= C \cdot \left(C^h C\right)^{-1} \underline{r}_d \\
C^h T &= 0 \\
\underline{u} &= T \cdot \underline{x}
\end{aligned}
$$

Adaptive GSC



$$\underline{w}_c = C \cdot \left(C^h C\right)^{-1} \underline{r}_d$$

$$C^h T = 0$$

$$\underline{u} = T \cdot \underline{x}$$

Optimal Wiener : $\underline{w}_{opt} = arg \min_{\underline{w}} \left\{ E\{|r|^2\} \right\} = R_u^{-1} \cdot \underline{r}_{ue^*}$

LMS update rule : $\underline{w}[k+1] = \underline{w}[k] + 2\alpha \underline{u}[k] y^*[k]$

Appendix AASP

Content appendix

- Eigenvalue problem
- Generalized inverse
- Projection matrix
- Matrix inversion lemma
- Signal subspace techniques

Eigenvalue problem

Procedure: With eigenvalues λ_i and eigenvectors $\underline{\mathrm{q}}_i$:

$$\mathrm{R} \cdot \underline{\mathrm{q}}_i = \lambda_i \cdot \underline{\mathrm{q}}_i \quad \Rightarrow \quad (\mathrm{R} - \lambda_i \mathrm{I}) \cdot \underline{\mathrm{q}}_i = \underline{0} \ \text{ for } i = 0, 1, \cdots, N-1$$

Eigenvalue problem

Procedure: With eigenvalues λ_i and eigenvectors $\underline{q}_i$:

$$\mathrm{R} \cdot \underline{q}_i = \lambda_i \cdot \underline{q}_i \quad \Rightarrow \quad (\mathrm{R} - \lambda_i \mathrm{I}) \cdot \underline{q}_i = \underline{0} \text{ for } i = 0, 1, \cdots, N-1$$

With $\mathrm{Q} = (\underline{q}_0, \cdots, \underline{q}_{N-1})$ and $\Lambda = diag\{\lambda_0, \cdots, \lambda_{N-1}\}$

$$\mathrm{R} \cdot \mathrm{Q} = \mathrm{Q} \cdot \Lambda \qquad \Rightarrow \qquad \mathrm{R} = \mathrm{Q} \Lambda \mathrm{Q}^{-1}$$

Eigenvalue problem

Procedure: With eigenvalues λ_i and eigenvectors $\underline{q}_i$:

$$\mathrm{R} \cdot \underline{q}_i = \lambda_i \cdot \underline{q}_i \quad \Rightarrow \quad (\mathrm{R} - \lambda_i \mathrm{I}) \cdot \underline{q}_i = \underline{0} \ \text{ for } i = 0, 1, \cdots, N-1$$

With $\mathrm{Q} = (\underline{q}_0, \cdots, \underline{q}_{N-1})$ and $\Lambda = diag\{\lambda_0, \cdots, \lambda_{N-1}\}$

$$\mathrm{R} \cdot \mathrm{Q} = \mathrm{Q} \cdot \Lambda \qquad \Rightarrow \qquad \mathrm{R} = \mathrm{Q} \Lambda \mathrm{Q}^{-1}$$

Property: *Eigenvectors* $\underline{q}_i$ *orthogonal* $\Rightarrow$

$$\mathrm{Q}^h \cdot \mathrm{Q} = \mathrm{Q} \cdot \mathrm{Q}^h = c \cdot \mathrm{I} \ \text{ with } \ c \ \text{ some constant}$$

Eigenvalue problem

Procedure: With eigenvalues λ_i and eigenvectors $\underline{q}_i$:

$$R \cdot \underline{q}_i = \lambda_i \cdot \underline{q}_i \quad \Rightarrow \quad (R - \lambda_i I) \cdot \underline{q}_i = \underline{0} \text{ for } i = 0, 1, \cdots, N-1$$

With $Q = (\underline{q}_0, \cdots, \underline{q}_{N-1})$ and $\Lambda = diag\{\lambda_0, \cdots, \lambda_{N-1}\}$

$$R \cdot Q = Q \cdot \Lambda \quad \Rightarrow \quad R = Q\Lambda Q^{-1}$$

Property: *Eigenvectors* $\underline{q}_i$ *orthogonal* $\Rightarrow$

$$Q^h \cdot Q = Q \cdot Q^h = c \cdot I \text{ with } c \text{ some constant}$$

Main result:

Diagonalization: $\boxed{Q^h R Q = \Lambda \quad \Leftrightarrow \quad R = Q\Lambda Q^h}$

Eigenvalue problem

Example MA(1):

$x[k] = i[k] + ai[k-1]$ with $E\{i[k]\} = 0$ and $E\{i^2[k]\} = \sigma_i^2 \Rightarrow$

Eigenvalue problem

Example MA(1):

$x[k] = i[k] + ai[k-1]$ with $E\{i[k]\} = 0$ and $E\{i^2[k]\} = \sigma_i^2 \Rightarrow$

$\rho[0] = (1+a^2)\sigma_i^2$; $\rho[1] = \rho[-1] = a\sigma_i^2$; $\rho[\tau] = 0$ for $|\tau| \geq 2$

Eigenvalue problem

Example MA(1):

$x[k] = i[k] + ai[k-1]$ with $E\{i[k]\} = 0$ and $E\{i^2[k]\} = \sigma_i^2 \Rightarrow$

$\rho[0] = (1+a^2)\sigma_i^2$; $\rho[1] = \rho[-1] = a\sigma_i^2$; $\rho[\tau] = 0$ for $|\tau| \geq 2$

Eigenvalues problem $\det(\mathrm{R} - \lambda \mathrm{I}) = 0$ for $N = 2$ (with $\gamma = \rho[1]/\rho[0]$):

$$\Lambda = \begin{pmatrix} \lambda_0 & 0 \\ 0 & \lambda_1 \end{pmatrix} = \begin{pmatrix} 1+\gamma & 0 \\ 0 & 1-\gamma \end{pmatrix} \; ; \; \mathrm{Q} = \left(\underline{\mathrm{q}}_0, \underline{\mathrm{q}}_1\right) = c \cdot \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Eigenvalue problem

Example MA(1):

$x[k] = i[k] + ai[k-1]$ with $E\{i[k]\} = 0$ and $E\{i^2[k]\} = \sigma_i^2 \Rightarrow$

$\rho[0] = (1+a^2)\sigma_i^2$; $\rho[1] = \rho[-1] = a\sigma_i^2$; $\rho[\tau] = 0$ for $|\tau| \geq 2$

Eigenvalues problem $\det(\mathrm{R} - \lambda \mathrm{I}) = 0$ for $N = 2$ (with $\gamma = \rho[1]/\rho[0]$):

$$\Lambda = \begin{pmatrix} \lambda_0 & 0 \\ 0 & \lambda_1 \end{pmatrix} = \begin{pmatrix} 1+\gamma & 0 \\ 0 & 1-\gamma \end{pmatrix} \;;\; \mathrm{Q} = \left(\underline{q}_0, \underline{q}_1\right) = c \cdot \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Notes:

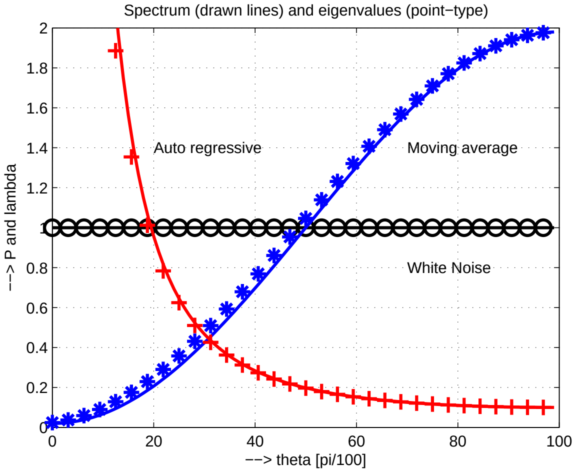
- Vector $\underline{q}_0$ orthogonal to $\underline{q}_1$ since $\underline{q}_0^t \cdot \underline{q}_1 = 0$
- For white noise ($a = 0$): $\Lambda = \mathrm{I}$
- For MA(1) with $N > 2$: R is tri-diagonal

Eigenvalue problem

Example: Eigenvalues and psd for white noise, MA(1) and AR(1)

Eigenvalue problem

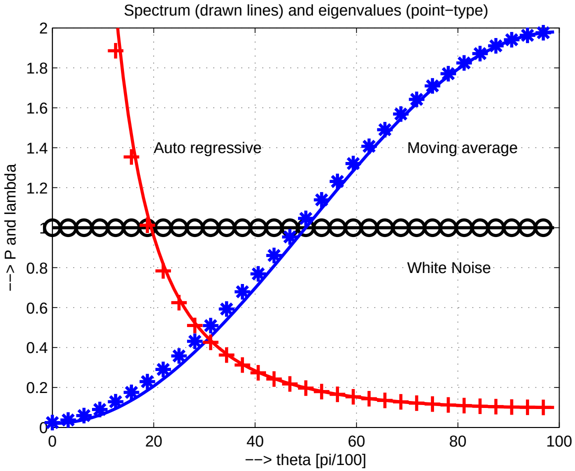
Example: Eigenvalues and psd for white noise, MA(1) and AR(1)



Eigenvalue problem

Back2Slides

Example: Eigenvalues and psd for white noise, MA(1) and AR(1)



Generalized inverse

Goal:
For known $M \times N$ matrix A and $M \times 1$ vector $\underline{\mathrm{b}}$, solve linear set of M equations and N unknowns:

$$\mathrm{A} \cdot \underline{\mathrm{w}} = \underline{\mathrm{b}}$$

Generalized inverse

Goal:

For known $M \times N$ matrix A *and $M \times 1$ vector* $\underline{\mathsf{b}}$, *solve linear set of M equations and N unknowns:*

$$\mathsf{A} \cdot \underline{\mathsf{w}} = \underline{\mathsf{b}}$$

General solution for $N \times 1$ vector $\underline{\mathsf{w}}$:

$$\boxed{\underline{\mathsf{w}} = \mathsf{A}^{\dagger} \cdot \underline{\mathsf{b}}}$$

with $\dagger$ the generalized (Moore-Penroose) pseudo inverse, defined as:

Generalized inverse

Goal:

For known $M \times N$ matrix A *and $M \times 1$ vector* $\underline{\mathsf{b}}$, *solve linear set of M equations and N unknowns:*

$$\mathsf{A} \cdot \underline{\mathsf{w}} = \underline{\mathsf{b}}$$

General solution for $N \times 1$ vector $\underline{\mathsf{w}}$:

$$\boxed{\underline{\mathsf{w}} = \mathsf{A}^{\dagger} \cdot \underline{\mathsf{b}}}$$

with $\dagger$ the generalized (Moore-Penroose) pseudo inverse, defined as:

$$\begin{aligned}
\mathsf{A}^{\dagger} &= \mathsf{A}^{h}\left(\mathsf{A}\mathsf{A}^{h}\right)^{-1} && \text{for} \quad M < N \\
\mathsf{A}^{\dagger} &= \mathsf{A}^{-1} && \text{for} \quad M = N \\
\mathsf{A}^{\dagger} &= \left(\mathsf{A}^{h}\mathsf{A}\right)^{-1}\mathsf{A}^{h} && \text{for} \quad M > N
\end{aligned}$$

Generalized inverse

Case $M < N$: $\underline{\mathrm{w}} = \mathrm{A}^{\dagger} \cdot \underline{\mathrm{b}} \Rightarrow$ Multiple solutions

Generalized inverse

Case $M < N$: $\underline{w} = A^{\dagger} \cdot \underline{b} \Rightarrow$ Multiple solutions

<u>Example $M = 1,\ N = 2$:</u>

$$w_1 + w_2 = 2 \;\;\Rightarrow\;\; A \cdot \underline{w} = \underline{b} \;\;\Leftrightarrow\;\; (1, 1) \cdot \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = (2)$$

Generalized inverse

Case $M < N$: $\underline{\mathrm{w}} = \mathrm{A}^{\dagger} \cdot \underline{\mathrm{b}} \Rightarrow$ Multiple solutions

<u>Example $M = 1,\ N = 2$:</u>

$$w_1 + w_2 = 2 \;\;\Rightarrow\;\; \mathrm{A} \cdot \underline{\mathrm{w}} = \underline{\mathrm{b}} \;\;\Leftrightarrow\;\; (1, 1) \cdot \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = (2)$$

$$\underline{\mathrm{w}} = \mathrm{A}^{\dagger} \cdot \underline{\mathrm{b}} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

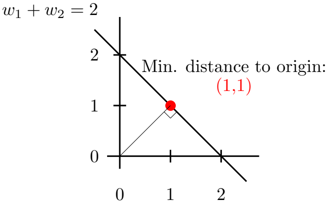
Generalized inverse

Case $M < N$: $\underline{\mathrm{w}} = \mathrm{A}^{\dagger} \cdot \underline{\mathrm{b}} \Rightarrow$ Multiple solutions

<u>Example $M = 1,\ N = 2$:</u>

$$w_1 + w_2 = 2 \quad \Rightarrow \quad \mathrm{A} \cdot \underline{\mathrm{w}} = \underline{\mathrm{b}} \quad \Leftrightarrow \quad (1,1) \cdot \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = (2)$$

$$\underline{\mathrm{w}} = \mathrm{A}^{\dagger} \cdot \underline{\mathrm{b}} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$



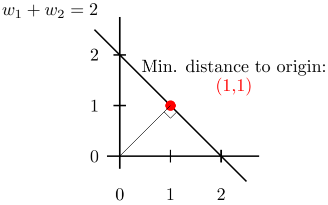
Generalized inverse

Case $M < N$: $\underline{w} = A^{\dagger} \cdot \underline{b} \Rightarrow$ Multiple solutions

Example $M = 1,\ N = 2$:

$$w_1 + w_2 = 2 \quad \Rightarrow \quad A \cdot \underline{w} = \underline{b} \quad \Leftrightarrow \quad (1,1) \cdot \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = (2)$$

$$\underline{w} = A^{\dagger} \cdot \underline{b} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$



Conclusion:

$\dagger$ results in solution with smallest Euclidian norm ("minimum distance to the origin (0,0)")

Generalized inverse

Case $M = N$: $\underline{w} = A^{-1} \cdot \underline{b}$ (A must be invertable)

Generalized inverse

Case $M = N$: $\underline{w} = A^{-1} \cdot \underline{b}$ (A must be invertable)

Example $M = 2,\ N = 2$:

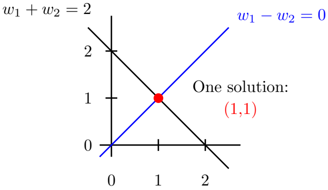
$$\begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \end{pmatrix} \quad \Rightarrow \quad \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

Generalized inverse

Case $M = N$: $\underline{\mathrm{w}} = \mathrm{A}^{-1} \cdot \underline{\mathrm{b}}$ (A must be invertable)

<u>Example $M = 2$, $N = 2$:</u>

$$\begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \end{pmatrix} \quad \Rightarrow \quad \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

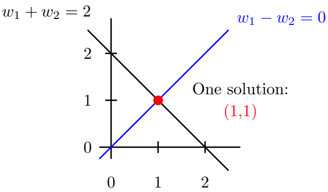


Generalized inverse

Case $M = N$: $\underline{w} = A^{-1} \cdot \underline{b}$ (A must be invertable)

Example $M = 2,\ N = 2$:

$$\begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \end{pmatrix} \quad \Rightarrow \quad \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$



Conclusion: † *results in unique solution*

Generalized inverse

Case $M > N$: Overdetermined set of equations

Generalized inverse

Case $M > N$: Overdetermined set of equations

Min. norm solution: $\underline{w} = \arg\min_{\underline{w}} ||A\underline{w} - \underline{b}||^2$

Generalized inverse

Case $M > N$: Overdetermined set of equations

Min. norm solution: $\underline{w} = \arg\min_{\underline{w}} ||A\underline{w} - \underline{b}||^2$

$$J = ||A\underline{w} - \underline{b}||^2 = \underline{w}^h A^h A\underline{w} - \underline{w}^h A^h \underline{b} - \underline{b}^h A\underline{w} + \underline{b}^h \underline{b}$$

Generalized inverse

Case $M > N$: Overdetermined set of equations

Min. norm solution: $\underline{w} = \arg\min_{\underline{w}} ||A\underline{w} - \underline{b}||^2$

$$J = ||A\underline{w} - \underline{b}||^2 = \underline{w}^h A^h A\underline{w} - \underline{w}^h A^h \underline{b} - \underline{b}^h A\underline{w} + \underline{b}^h \underline{b}$$

$$\Rightarrow \quad \frac{\mathrm{d}J}{\mathrm{d}\underline{w}} = 0 \quad \Rightarrow \quad \underline{w} = \left(A^h A\right)^{-1} A^h \cdot \underline{b} = A^{\dagger} \cdot \underline{b}$$

Generalized inverse

Case $M > N$: Overdetermined set of equations

Min. norm solution: $\underline{w} = \arg\min_{\underline{w}} ||A\underline{w} - \underline{b}||^2$

$$J = ||A\underline{w} - \underline{b}||^2 = \underline{w}^h A^h A \underline{w} - \underline{w}^h A^h \underline{b} - \underline{b}^h A \underline{w} + \underline{b}^h \underline{b}$$

$$\Rightarrow \quad \frac{dJ}{d\underline{w}} = 0 \quad \Rightarrow \quad \underline{w} = \left(A^h A\right)^{-1} A^h \cdot \underline{b} = A^\dagger \cdot \underline{b}$$

<u>Example $M = 3$, $N = 2$:</u>

$$\begin{pmatrix} 1 & -1 \\ 1 & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \\ 2 \end{pmatrix} \quad \Rightarrow \quad \underline{w} = A^\dagger \cdot \underline{b} = \begin{pmatrix} \frac{4}{3} \\ 1 \end{pmatrix}$$

Generalized inverse

Case $M > N$: Overdetermined set of equations

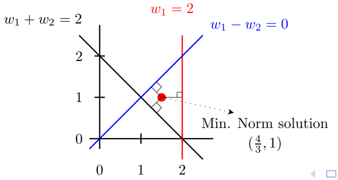
Min. norm solution: $\underline{w} = \arg\min_{\underline{w}} ||A\underline{w} - \underline{b}||^2$

$$J \;=\; ||A\underline{w} - \underline{b}||^2 = \underline{w}^h A^h A \underline{w} - \underline{w}^h A^h \underline{b} - \underline{b}^h A \underline{w} + \underline{b}^h \underline{b}$$

$$\Rightarrow \quad \frac{\mathrm{d}J}{\mathrm{d}\underline{w}} = 0 \quad \Rightarrow \quad \underline{w} = \left(A^h A\right)^{-1} A^h \cdot \underline{b} = A^\dagger \cdot \underline{b}$$

<u>Example $M = 3,\ N = 2$:</u>

$$\begin{pmatrix} 1 & -1 \\ 1 & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \\ 2 \end{pmatrix} \quad \Rightarrow \quad \underline{w} = A^\dagger \cdot \underline{b} = \begin{pmatrix} \frac{4}{3} \\ 1 \end{pmatrix}$$



Generalized inverse

Back2Slides

Case $M > N$: Overdetermined set of equations

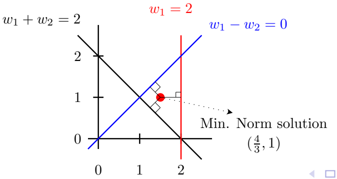
Min. norm solution: $\underline{w} = \arg\min_{\underline{w}} ||A\underline{w} - \underline{b}||^2$

$$J = ||A\underline{w} - \underline{b}||^2 = \underline{w}^h A^h A\underline{w} - \underline{w}^h A^h \underline{b} - \underline{b}^h A\underline{w} + \underline{b}^h \underline{b}$$

$$\Rightarrow \quad \frac{dJ}{d\underline{w}} = 0 \quad \Rightarrow \quad \underline{w} = \left(A^h A\right)^{-1} A^h \cdot \underline{b} = A^{\dagger} \cdot \underline{b}$$

Example $M = 3,\ N = 2$:

$$\begin{pmatrix} 1 & -1 \\ 1 & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \\ 2 \end{pmatrix} \quad \Rightarrow \quad \underline{w} = A^{\dagger} \cdot \underline{b} = \begin{pmatrix} \frac{4}{3} \\ 1 \end{pmatrix}$$



Projection matrix

Square matrix P is **Projection** matrix if: $P^2 = P$

Projection matrix

Square matrix P is **Projection** matrix if: $\mathrm{P}^2 = \mathrm{P}$

Orthogonal projection matrix: $\mathrm{P}^h = \mathrm{P}$ and $\mathrm{P}^2 = \mathrm{P}$

Projection matrix

Square matrix P is **Projection** matrix if: $\mathrm{P}^2 = \mathrm{P}$

Orthogonal projection matrix: $\mathrm{P}^h = \mathrm{P}$ and $\mathrm{P}^2 = \mathrm{P}$

General: $N \times M$ matrix V, with linearly independent columns

Projection $(N \times 1)$ $\underline{\mathrm{b}}$ onto M-dim subspace spanned by columns V:

$$\hat{\underline{\mathrm{b}}} = \mathrm{P}_V \cdot \underline{\mathrm{b}} \quad \text{and} \quad \underline{\mathrm{b}}^{\perp} = (\mathbf{I} - \mathrm{P}_V) \cdot \underline{\mathrm{b}}$$

$$\text{with projection matrix : } \boxed{\mathrm{P}_V = \mathrm{V}\left(\mathrm{V}^h\mathrm{V}\right)^{-1}\mathrm{V}^h}$$

Projection matrix

Square matrix P is **Projection** matrix if: $\mathrm{P}^2 = \mathrm{P}$

Orthogonal projection matrix: $\mathrm{P}^h = \mathrm{P}$ and $\mathrm{P}^2 = \mathrm{P}$

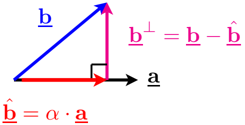
General: $N \times M$ matrix V, with linearly independent columns

Projection $(N \times 1)$ $\underline{\mathrm{b}}$ onto M-dim subspace spanned by columns V:

$$\hat{\underline{\mathrm{b}}} = \mathrm{P}_V \cdot \underline{\mathrm{b}} \quad \text{and} \quad \underline{\mathrm{b}}^\perp = (\mathrm{I} - \mathrm{P}_V) \cdot \underline{\mathrm{b}}$$

$$\text{with projection matrix : } \boxed{\mathrm{P}_V = \mathrm{V}\left(\mathrm{V}^h\mathrm{V}\right)^{-1}\mathrm{V}^h}$$

Simple example:



$$\hat{\underline{\mathrm{b}}}^h \cdot \underline{\mathrm{b}}^\perp = 0 \;\Rightarrow\; \alpha = (\underline{\mathrm{a}}^h\underline{\mathrm{a}})^{-1}\underline{\mathrm{a}}^h \cdot \underline{\mathrm{b}}$$

$$\Rightarrow\; \hat{\underline{\mathrm{b}}} = \mathrm{P}_a \cdot \underline{\mathrm{b}} \quad \text{and} \quad \underline{\mathrm{b}}^\perp = (\mathrm{I} - \mathrm{P}_a) \cdot \underline{\mathrm{b}}$$

with $\mathrm{P}_a = \underline{\mathrm{a}}(\underline{\mathrm{a}}^h\underline{\mathrm{a}})^{-1}\underline{\mathrm{a}}^h$

Matrix inversion lemma

Matrix dimensions: A: $N \times N$; B: $N \times M$; C: $M \times M$; D: $M \times N$

$$\boxed{(\mathsf{A} + \mathsf{BCD})^{-1} = \mathsf{A}^{-1} - \mathsf{A}^{-1}\mathsf{B}\left(\mathsf{D}\mathsf{A}^{-1}\mathsf{B} + \mathsf{C}^{-1}\right)^{-1}\mathsf{D}\mathsf{A}^{-1}}$$

Matrix inversion lemma

Matrix dimensions: A: $N \times N$; B: $N \times M$; C: $M \times M$; D: $M \times N$

$$\boxed{(\mathrm{A} + \mathrm{BCD})^{-1} = \mathrm{A}^{-1} - \mathrm{A}^{-1}\mathrm{B}\left(\mathrm{DA}^{-1}\mathrm{B} + \mathrm{C}^{-1}\right)^{-1}\mathrm{DA}^{-1}}$$

<u>Simple example:</u> (scalar case)

$$(a + x \cdot y)^{-1} = \frac{1}{a + x \cdot y} = \cdots = a^{-1} - \frac{a^{-1}xya^{-1}}{1 + ya^{-1}x}$$

Matrix inversion lemma

Matrix dimensions: A: $N \times N$; B: $N \times M$; C: $M \times M$; D: $M \times N$

$$\boxed{(\mathrm{A} + \mathrm{BCD})^{-1} = \mathrm{A}^{-1} - \mathrm{A}^{-1}\mathrm{B}\left(\mathrm{DA}^{-1}\mathrm{B} + \mathrm{C}^{-1}\right)^{-1}\mathrm{DA}^{-1}}$$

Simple example: (scalar case)

$$(a + x \cdot y)^{-1} = \frac{1}{a + x \cdot y} = \cdots = a^{-1} - \frac{a^{-1}xya^{-1}}{1 + ya^{-1}x}$$

Special case: (RLS-like)

$\mathrm{B} = \underline{\mathrm{x}}$: $N \times 1$; $\mathrm{C} = 1$: 1×1 and $\mathrm{D} = \underline{\mathrm{x}}^h$

$$\left(\mathrm{A} + \underline{\mathrm{x}}\underline{\mathrm{x}}^h\right)^{-1} = \mathrm{A}^{-1} - \frac{\mathrm{A}^{-1}\underline{\mathrm{x}}\underline{\mathrm{x}}^h\mathrm{A}^{-1}}{1 + \underline{\mathrm{x}}^h\mathrm{A}^{-1}\underline{\mathrm{x}}}$$

Signal subspace techniques

Goal: Determine spectral peaks in noisy measurements

Signal subspace techniques

Goal: Determine spectral peaks in noisy measurements

Signal model: J sensors, P sources $(P < J)$. For $i = 1, \cdots, J$:

$$x_i = \sum_{p=1}^{P} a_i(\theta_p) \cdot s_p[k] + n_i[k] \quad \Leftrightarrow \quad \underline{\mathrm{x}}[k] = \mathrm{A} \cdot \underline{\mathrm{s}}[k] + \underline{\mathrm{n}}[k]$$

Signal subspace techniques

Goal: Determine spectral peaks in noisy measurements

Signal model: J sensors, P sources $(P < J)$. For $i = 1, \cdots, J$:

$$x_i = \sum_{p=1}^{P} a_i(\theta_p) \cdot s_p[k] + n_i[k] \quad \Leftrightarrow \quad \underline{x}[k] = \mathrm{A} \cdot \underline{s}[k] + \underline{n}[k]$$

Covariance structure:

$$\mathrm{R}_x = E\{\underline{x} \cdot \underline{x}^h\} = \mathrm{A}\mathrm{R}_s\mathrm{A}^h + \mathrm{R}_n$$

with $\mathrm{R}_s = E\{\underline{ss}^h\} = \mathrm{diag}\{\sigma_{s_1}^2, \cdots, \sigma_{s_P}^2\}$; $\mathrm{R}_n = \sigma_n^2 \mathrm{I}$

and $J \times P$ steering matrix $\mathrm{A} = (\underline{a}(\theta_1), \cdots, \underline{a}(\theta_P))$

Signal subspace techniques

What about rank of these matrices?

Signal subspace techniques

<u>**What about rank of these matrices?**</u>

$\text{Rank}\{\text{A}\} = P$; $\text{Rank}\{\text{R}_s\} = P \Rightarrow \text{Rank}\{\text{A}\text{R}_s\text{A}^h\} = P$

Signal subspace techniques

What about rank of these matrices?

$\text{Rank}\{\mathrm{A}\} = P$; $\text{Rank}\{\mathrm{R}_s\} = P \Rightarrow \text{Rank}\{\mathrm{A}\mathrm{R}_s\mathrm{A}^h\} = P$

Furthermore since $\text{Rank}\{\mathrm{R}_n\} = J \Rightarrow \text{Rank}\{\mathrm{R}_x\} = J$

Signal subspace techniques

<u>What about rank of these matrices?</u>

$\text{Rank}\{\text{A}\} = P$; $\text{Rank}\{\text{R}_s\} = P \Rightarrow \text{Rank}\{\text{AR}_s\text{A}^h\} = P$

Furthermore since $\text{Rank}\{\text{R}_n\} = J \Rightarrow \text{Rank}\{\text{R}_x\} = J$

Eigenvalue decomposition source signal part:

$$\left(\text{AR}_s\text{A}^h\right) \cdot \underline{\text{u}}_i = \lambda_{s_i} \cdot \underline{\text{u}}_i \quad i = 1, \cdots, J$$

Signal subspace techniques

What about rank of these matrices?

$\text{Rank}\{\text{A}\} = P$; $\text{Rank}\{\text{R}_s\} = P \Rightarrow \text{Rank}\{\text{AR}_s\text{A}^h\} = P$

Furthermore since $\text{Rank}\{\text{R}_n\} = J \Rightarrow \text{Rank}\{\text{R}_x\} = J$

Eigenvalue decomposition source signal part:

$$\left(\text{AR}_s\text{A}^h\right) \cdot \underline{\text{u}}_i = \lambda_{s_i} \cdot \underline{\text{u}}_i \quad i = 1, \cdots, J$$

Since $\text{Rank}\{\text{AR}_s\text{A}^h\} = P \Rightarrow \lambda_{s_{P+1}} = \cdots = \lambda_{s_J} = 0$

Signal subspace techniques

What about rank of these matrices?

$\text{Rank}\{\mathrm{A}\} = P$; $\text{Rank}\{\mathrm{R}_s\} = P \Rightarrow \text{Rank}\{\mathrm{A}\mathrm{R}_s\mathrm{A}^h\} = P$

Furthermore since $\text{Rank}\{\mathrm{R}_n\} = J \Rightarrow \text{Rank}\{\mathrm{R}_x\} = J$

Eigenvalue decomposition source signal part:

$$\left(\mathrm{A}\mathrm{R}_s\mathrm{A}^h\right) \cdot \underline{\mathrm{u}}_i = \lambda_{s_i} \cdot \underline{\mathrm{u}}_i \quad i = 1, \cdots, J$$

Since $\text{Rank}\{\mathrm{A}\mathrm{R}_s\mathrm{A}^h\} = P \Rightarrow \lambda_{s_{P+1}} = \cdots = \lambda_{s_J} = 0$

Order eigenvalues: $\lambda_{s_1} \geq \cdots \geq \lambda_{s_P} > 0$

Signal subspace techniques

What about rank of these matrices?

$\text{Rank}\{\mathrm{A}\} = P$; $\text{Rank}\{\mathrm{R}_s\} = P \Rightarrow \text{Rank}\{\mathrm{A}\mathrm{R}_s\mathrm{A}^h\} = P$

Furthermore since $\text{Rank}\{\mathrm{R}_n\} = J \Rightarrow \text{Rank}\{\mathrm{R}_x\} = J$

Eigenvalue decomposition source signal part:

$$\left(\mathrm{A}\mathrm{R}_s\mathrm{A}^h\right) \cdot \underline{\mathrm{u}}_i = \lambda_{s_i} \cdot \underline{\mathrm{u}}_i \quad i = 1, \cdots, J$$

Since $\text{Rank}\{\mathrm{A}\mathrm{R}_s\mathrm{A}^h\} = P \Rightarrow \lambda_{s_{P+1}} = \cdots = \lambda_{s_J} = 0$

Order eigenvalues: $\lambda_{s_1} \geq \cdots \geq \lambda_{s_P} > 0$

$$\Rightarrow \quad \left(\mathrm{A}\mathrm{R}_s\mathrm{A}^h\right) \cdot \mathrm{U}_s = \mathrm{U}_s \cdot \Lambda_s$$

with $\mathrm{U}_s = (\underline{\mathrm{u}}_1, \cdots, \underline{\mathrm{u}}_P)$; $\Lambda_s = \text{diag}\{\lambda_{s_1}, \cdots, \lambda_{s_P}\}$

Signal subspace techniques

Eigenvalue decomposition input signal (use $\underline{\mathrm{u}}_i$ for $i = 1, \cdots, J$):

$$\mathrm{R}_x \cdot \underline{\mathrm{u}}_i = \left(\mathrm{A}\mathrm{R}_s\mathrm{A}^h\right) \cdot \underline{\mathrm{u}}_i + \sigma_n^2\mathrm{I} \cdot \underline{\mathrm{u}}_i = (\lambda_{s_i} + \sigma_n^2) \cdot \underline{\mathrm{u}}_i$$

Signal subspace techniques

Eigenvalue decomposition input signal (use $\underline{u}_i$ for $i = 1, \cdots, J$):

$$R_x \cdot \underline{u}_i = \left(A R_s A^h \right) \cdot \underline{u}_i + \sigma_n^2 \mathbf{I} \cdot \underline{u}_i = (\lambda_{s_i} + \sigma_n^2) \cdot \underline{u}_i$$

Thus eigenvalues can be divided into two groups:

$$\lambda_{x_i} = \begin{cases} \lambda_{s_i} + \sigma_n^2 & \text{for } i = 1, \cdots, P \\ \sigma_n^2 & \text{for } i = P+1, \cdots, J \end{cases}$$

Signal subspace techniques

Eigenvalue decomposition input signal (use $\underline{u}_i$ for $i = 1, \cdots, J$):

$$\mathrm{R}_x \cdot \underline{u}_i = \left(\mathrm{A}\mathrm{R}_s\mathrm{A}^h\right) \cdot \underline{u}_i + \sigma_n^2 \mathrm{I} \cdot \underline{u}_i = (\lambda_{s_i} + \sigma_n^2) \cdot \underline{u}_i$$

Thus eigenvalues can be divided into two groups:

$$\lambda_{x_i} = \begin{cases} \lambda_{s_i} + \sigma_n^2 & \text{for } i = 1, \cdots, P \\ \sigma_n^2 & \text{for } i = P+1, \cdots, J \end{cases}$$

With $\mathrm{U}_x = (\underline{u}_1, \cdots, \underline{u}_J)$ and $\Lambda_x = \mathrm{diag}\{\lambda_{x_1}, \cdots, \lambda_{x_J}\}$ we can write:

$$\begin{aligned} \mathrm{R}_x &= \mathrm{U}_x \Lambda_x \mathrm{U}_x^h = \sum_{i=1}^{J} \lambda_{x_i} \underline{u}_i \underline{u}_i^h = \sum_{i=1}^{P} (\lambda_{s_i} + \sigma_n^2) \underline{u}_i \underline{u}_i^h + \sum_{i=P+1}^{J} \sigma_n^2 \underline{u}_i \underline{u}_i^h \\ &= \mathrm{U}_s \Lambda_{s,n} \mathrm{U}_s^h + \mathrm{U}_n \Lambda_n \mathrm{U}_n^h \end{aligned}$$

Signal subspace techniques

$$
\begin{aligned}
\textbf{Signal subspace} \quad &: \quad \mathrm{U}_s = (\underline{\mathrm{u}}_1, \cdots, \underline{\mathrm{u}}_P) \\
& \quad \Lambda_{s,n} = \mathrm{diag}\{\lambda_{s_1} + \sigma_n^2, \cdots, \lambda_{s_P} + \sigma_n^2\} \\
\textbf{Noise subspace} \quad &: \quad \mathrm{U}_n = (\underline{\mathrm{u}}_{P+1}, \cdots, \underline{\mathrm{u}}_J) \\
& \quad \Lambda_n = \mathrm{diag}\{\sigma_n^2, \cdots, \sigma_n^2\}
\end{aligned}
$$

Signal subspace techniques

$$
\begin{aligned}
\textbf{Signal subspace} \quad &: \quad U_s = (\underline{u}_1, \cdots, \underline{u}_P) \\
&\quad\;\; \Lambda_{s,n} = \mathrm{diag}\{\lambda_{s_1} + \sigma_n^2, \cdots, \lambda_{s_P} + \sigma_n^2\} \\
\textbf{Noise subspace} \quad &: \quad U_n = (\underline{u}_{P+1}, \cdots, \underline{u}_J) \\
&\quad\;\; \Lambda_n = \mathrm{diag}\{\sigma_n^2, \cdots, \sigma_n^2\}
\end{aligned}
$$

$$
\begin{aligned}
\text{Since} \qquad & U_x = (\underline{u}_1, \cdots, \underline{u}_J) = (U_s, U_n) \\
\text{and} \qquad & \text{all eigenvectors orthogonal} \quad \underline{u}_i \perp \underline{u}_j
\end{aligned}
$$

$$
\Rightarrow \quad \boxed{U_s \perp U_n \quad \Leftrightarrow \quad U_s^h \cdot U_n = 0 \Leftrightarrow \quad U_n^h \cdot U_s = 0}
$$

Signal subspace techniques

$$
\begin{aligned}
\textbf{Signal subspace} \quad : \quad & \mathrm{U}_s = (\underline{\mathrm{u}}_1, \cdots, \underline{\mathrm{u}}_P) \\
& \Lambda_{s,n} = \mathrm{diag}\{\lambda_{s_1} + \sigma_n^2, \cdots, \lambda_{s_P} + \sigma_n^2\} \\
\textbf{Noise subspace} \quad : \quad & \mathrm{U}_n = (\underline{\mathrm{u}}_{P+1}, \cdots, \underline{\mathrm{u}}_J) \\
& \Lambda_n = \mathrm{diag}\{\sigma_n^2, \cdots, \sigma_n^2\}
\end{aligned}
$$

Since $\quad \mathrm{U}_x = (\underline{\mathrm{u}}_1, \cdots, \underline{\mathrm{u}}_J) = (\mathrm{U}_s, \mathrm{U}_n)$

and $\quad$ all eigenvectors orthogonal $\underline{\mathrm{u}}_i \perp \underline{\mathrm{u}}_j$

$$\Rightarrow \quad \boxed{\mathrm{U}_s \perp \mathrm{U}_n \quad \Leftrightarrow \quad \mathrm{U}_s^h \cdot \mathrm{U}_n = 0 \Leftrightarrow \quad \mathrm{U}_n^h \cdot \mathrm{U}_s = 0}$$

Conclusion:

Any vector from signal subspace is orthogonal to noise subspace

End

Appendix