

## Constrained Optimization

ECE 6279: Spatial Array Processing  
Spring 2007  
Lecture 16  
2/19/07

Prof. Aaron D. Lanterman

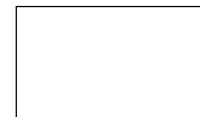
School of Electrical & Computer Engineering  
Georgia Institute of Technology  
AL: 404-385-2548  
<lanterma@ece.gatech.edu>

Copyright 2007 Aaron Lanterman



## Where We Are in J&D

- Section 7.2 and Appendix C



Copyright 2007 Aaron Lanterman



## Maximizing SNR for Known Colored Noise

- In Lecture 13, we showed that for

$$\text{Data: } \underline{\mathbf{y}} = \mathbf{e}(\vec{k}^0)\underline{\mathbf{s}} + \underline{\mathbf{n}}$$

$$E[|\underline{\mathbf{s}}|^2] = \sigma_s^2 \quad E[\underline{\mathbf{n}}\underline{\mathbf{n}}^H] = \mathbf{K}_n$$

we can maximize SNR by choosing

$$z = \mathbf{a}^H \underline{\mathbf{y}} = \mathbf{e}^H(\vec{k}^0)\mathbf{K}_n^{-1}\underline{\mathbf{y}}$$

$$\mathbf{a} = \mathbf{K}_n^{-1}\mathbf{e}(\vec{k}^0)$$



Copyright 2007 Aaron Lanterman



## Warning: Notation Change Ahead

- To match the book, we'll change  $\mathbf{a}$  to  $\mathbf{w}$
- This  $\mathbf{w}$  should not be confused with the “shading,” “tapering,” aka “windowing” weights we had along the diagonal of a matrix called  $\mathbf{W}$



Copyright 2007 Aaron Lanterman



## Unknown Colored Noise

- What if you don't know  $\mathbf{K}_n$ ?
- Be pessimistic: assume data is mostly noise, which we want to minimize, so minimize beamformer output

$$E[|z|^2] = E[|\mathbf{w}^H \mathbf{y}|^2]$$

$$= E[\mathbf{w}^H \mathbf{y} \mathbf{y}^H \mathbf{w}] = \mathbf{w}^H \mathbf{R}_y \mathbf{w}$$

Copyright 2007 Aaron Lanterman



## Problem Specification

$$\mathbf{w}_\diamond = \underset{\mathbf{w}}{\operatorname{arg\,min}} \mathbf{w}^H \mathbf{R}_y \mathbf{w}$$

- Has trivial solution of  $\mathbf{w}_\diamond = \mathbf{0}$
- To avoid nonsensical answer, add constraint  $\mathbf{C}\mathbf{w} = \mathbf{c}$
- For ex.,  $\mathbf{e}^H(\vec{k})\mathbf{w} = 1$  assures signal in look direction passes unharmed

Copyright 2007 Aaron Lanterman



## Real Vector-Parameter Optimization

$$\nabla_{\mathbf{x}} f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_M} \end{bmatrix} \stackrel{\text{set}}{=} \mathbf{0} \text{ to find stationary points}$$

Copyright 2007 Aaron Lanterman



## Gradients of Quadratic Forms (Real)

$$[\nabla_{\mathbf{x}} \{\mathbf{x}^T \mathbf{A} \mathbf{x}\}]_\ell = \left[ \nabla_{\mathbf{x}} \left\{ \sum_{ij} A_{ij} x_i x_j \right\} \right]_\ell$$

$$= \frac{d}{dx_l} \left\{ \sum_{ij} A_{ij} x_i x_j \right\} = \sum_i A_{il} x_i + \sum_j A_{lj} x_j$$

$$= \mathbf{A} \mathbf{x} + \mathbf{A}^T \mathbf{x} = 2\mathbf{A} \mathbf{x}$$

↑  
if  $\mathbf{A}$  is symmetric

Copyright 2007 Aaron Lanterman



### Complex Parameter Optimization (1)

- How can we minimize a real-valued function of a complex variable?  $f(\text{Re}\{z\}, \text{Im}\{z\})$
- Could take  $\frac{\partial f}{\partial \text{Re}\{z\}}, \frac{\partial f}{\partial \text{Im}\{z\}}$ , but that gets messy



Copyright 2007 Aaron Lanterman

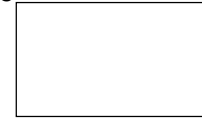


### Complex Parameter Optimization (2)

$$\text{Re}\{z\} = \frac{1}{2}[z + z^*], \quad \text{Im}\{z\} = \frac{1}{2j}[z - z^*]$$

$$\begin{bmatrix} 2\text{Re}\{z\} \\ 2j\text{Im}\{z\} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}}_{\text{invertible}} \begin{bmatrix} z \\ z^* \end{bmatrix}$$

- Can treat  $z$  and  $z^*$  as independent variables



Copyright 2007 Aaron Lanterman



### Complex Parameter Optimization (3)

- For scalars, stationary points can be found via

$$\frac{\partial f(z, z^*)}{\partial z} = 0 \quad \text{or} \quad \underbrace{\frac{\partial f(z, z^*)}{\partial z^*}}_{\text{usually easier}} = 0$$

- For vectors, must use  $\nabla_{z^*} f(z, z^*) = 0$  to find stationary points



Copyright 2007 Aaron Lanterman



### The Lagrangian

$$L(\mathbf{w}, \boldsymbol{\lambda}) = \mathbf{w}^H \mathbf{R}_y \mathbf{w} + \underbrace{\boldsymbol{\lambda}^H (\mathbf{C}\mathbf{w} - \mathbf{c}) + \boldsymbol{\lambda}^T (\mathbf{C}^* \mathbf{w}^* - \mathbf{c}^*)}_{\text{real-valued}}$$

↑  
treat  $\boldsymbol{\lambda}$  and  $\boldsymbol{\lambda}^*$  independently

- Caveat: for Lagrange trick to work, columns of  $\mathbf{C}$  must be linearly independent



Copyright 2007 Aaron Lanterman



### Gradients of Quadratic Forms (Complex)

$$\left[ \nabla_{\mathbf{w}^*} \{ \mathbf{w}^H \mathbf{R} \mathbf{w} \} \right]_{\ell} = \left[ \nabla_{\mathbf{w}^*} \left\{ \sum_{ij} R_{ij} w_i^* w_j \right\} \right]_{\ell}$$

$$= \frac{\partial}{\partial w_l^*} \left\{ \sum_{ij} R_{ij} w_i^* w_j \right\} = \sum_j R_{lj} w_j$$

$$\nabla_{\mathbf{w}^*} \{ \mathbf{w}^H \mathbf{R} \mathbf{w} \} = \mathbf{R} \mathbf{w}$$



Copyright 2007 Aaron Lanterman



### Gradients of Inner Products (Complex)

$$\left[ \nabla_{\mathbf{w}^*} \{ \mathbf{w}^H \mathbf{v} \} \right]_{\ell} = \left[ \nabla_{\mathbf{w}^*} \left\{ \sum_i w_i^* v_i \right\} \right]_{\ell} = v_l$$

$$\nabla_{\mathbf{w}^*} \{ \mathbf{w}^H \mathbf{v} \} = \mathbf{v}$$



Copyright 2007 Aaron Lanterman



### Dissecting the Lagrangian

$$L(\mathbf{w}, \boldsymbol{\lambda}) = \mathbf{w}^H \mathbf{R}_y \mathbf{w} + \boldsymbol{\lambda}^H (\mathbf{C} \mathbf{w} - \mathbf{c}) + \underbrace{\boldsymbol{\lambda}^T (\mathbf{C}^* \mathbf{w}^*)}_{\hat{\boldsymbol{\lambda}}^T \mathbf{C}^* \mathbf{w}^*} - \mathbf{c}^*$$

$$\hat{\boldsymbol{\lambda}}^T \mathbf{C}^* \mathbf{w}^* = (\boldsymbol{\lambda}^T \mathbf{C}^* \mathbf{w}^*)^T = \mathbf{w}^H \mathbf{C}^H \boldsymbol{\lambda}$$

$$\begin{aligned} \nabla_{\mathbf{w}^*} L(\mathbf{w}, \boldsymbol{\lambda}) &= \nabla_{\mathbf{w}^*} \{ \mathbf{w}^H \mathbf{R}_y \mathbf{w} + \mathbf{w}^H \mathbf{C}^H \boldsymbol{\lambda} \} \\ &= \mathbf{R}_y \mathbf{w} + \mathbf{C}^H \boldsymbol{\lambda} \end{aligned}$$



Copyright 2007 Aaron Lanterman



### Eliminating the Lagrange Multiplier

$$\mathbf{R}_y \mathbf{w}_{\diamond} + \mathbf{C}^H \boldsymbol{\lambda}_{\diamond} = 0$$

$$\mathbf{w}_{\diamond} = -\mathbf{R}_y^{-1} \mathbf{C}^H \boldsymbol{\lambda}_{\diamond}$$

want to get rid of  
Lagrange multiplier

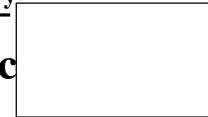
$$= -\mathbf{R}_y^{-1} \mathbf{C}^H [-(\mathbf{C} \mathbf{R}_y^{-1} \mathbf{C}^H)^{-1} \mathbf{c}]$$

$$= \mathbf{R}_y^{-1} \mathbf{C}^H (\mathbf{C} \mathbf{R}_y^{-1} \mathbf{C}^H)^{-1} \mathbf{c}$$

Use constraint:

$$\mathbf{C} \mathbf{w}_{\diamond} = \mathbf{c} \rightarrow -\mathbf{C} \mathbf{R}_y^{-1} \mathbf{C}^H \boldsymbol{\lambda}_{\diamond} = \mathbf{c}$$

$$\boldsymbol{\lambda}_{\diamond} = -(\mathbf{C} \mathbf{R}_y^{-1} \mathbf{C}^H)^{-1} \mathbf{c}$$



Copyright 2007 Aaron Lanterman



## Power of Beamformer Output

$$\mathbf{w} = \mathbf{R}_y^{-1} \mathbf{C}^H (\mathbf{C} \mathbf{R}_y^{-1} \mathbf{C}^H)^{-1} \mathbf{c}$$

Power of beamformer output:  $\mathbf{w}^H \mathbf{R}_y \mathbf{w} =$   
 $\mathbf{c}^H (\mathbf{C} \mathbf{R}_y^{-1} \mathbf{C}^H)^{-H} \mathbf{C} \mathbf{R}_y^{-H} \mathbf{R}_y^{-1} \mathbf{C}^H (\mathbf{C} \mathbf{R}_y^{-1} \mathbf{C}^H)^{-1} \mathbf{c}$

(Using  $\mathbf{A}^H = \mathbf{A} \rightarrow \mathbf{A}^{-H} \equiv (\mathbf{A}^{-1})^H = \mathbf{A}^{-1}$ )

$$= \mathbf{c}^H (\mathbf{C} \mathbf{R}_y^{-1} \mathbf{C}^H)^{-1} \mathbf{C} \mathbf{R}_y^{-1} \mathbf{C}^H (\mathbf{C} \mathbf{R}_y^{-1} \mathbf{C}^H)^{-1} \mathbf{c}$$

$$= \mathbf{c}^H (\mathbf{C} \mathbf{R}_y^{-1} \mathbf{C}^H)^{-1} \mathbf{c}$$

