

---

---

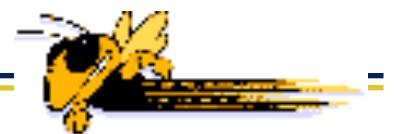
# ***Robust Constrained Optimization***

**ECE 6279: Spatial Array Processing  
Fall 2013  
Lecture 19**

**Prof. Aaron D. Lanterman**

**School of Electrical & Computer Engineering  
Georgia Institute of Technology**

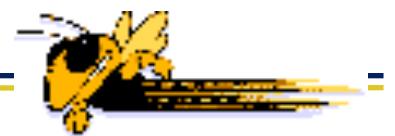
AL: 404-385-2548  
[<lanterma@ece.gatech.edu>](mailto:lanterma@ece.gatech.edu)



# Where We Are in J&D

---

- **Section 7.4.3**
- **Note: these slides will use R instead of  $R_y$**



# Adaptive Techniques So Far

---

- Recall MVDR formulation:

$$\min_{\mathbf{w}} \mathbf{w}^H \mathbf{R} \mathbf{w} \text{ s.t. } \mathbf{e}^H(\vec{k}) \mathbf{w} = 1$$

- Then derived EV, MUSIC, etc.



# Some Potential Problems

---

- Problem: If we don't know  $e(\vec{k})$  exactly, these techniques may break down
- Sensitivity to model mismatches increases with SNR!
- Overall sensitive to quality of estimate of  $R$

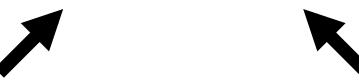


# Problems with Few Snapshots

---

- All methods use empirical correlation matrix:

$$\hat{\mathbf{R}} = \frac{1}{L} \sum_{l=0}^{L-1} \mathbf{y}(l) \mathbf{y}^H(l)$$

- All methods need a “full rank”  $\hat{\mathbf{R}}$
  - Hence, need  $L \geq M$
- No. of snapshots      No. of sensors
- 



# Try Building Errors into Formulation

---

$$\min_w \mathbf{w}^H \mathbf{R} \mathbf{w} \text{ s.t. } (\mathbf{e} + \boldsymbol{\delta})^H \mathbf{w} = 1$$

$$\text{and } \|\boldsymbol{\delta}\|^2 = \boldsymbol{\delta}^H \boldsymbol{\delta} \leq \varepsilon^2$$

- Note will also need to “find”  $\boldsymbol{\delta}$
- Use Lagrange multipliers

$$\begin{aligned} L &= \mathbf{w}^H \mathbf{R} \mathbf{w} \\ &+ \lambda_1^* [(\mathbf{e} + \boldsymbol{\delta})^H \mathbf{w} - 1] + \lambda_1 [(\mathbf{e} + \boldsymbol{\delta})^T \mathbf{w}^* - 1] \\ &+ \lambda_2 [\boldsymbol{\delta}^H \boldsymbol{\delta} - \varepsilon^2] \end{aligned}$$



# Taking Gradients

---

$$\begin{aligned} L &= \mathbf{w}^H \mathbf{R} \mathbf{w} \\ &+ \lambda_1^* [(\mathbf{e} + \boldsymbol{\delta})^H \mathbf{w} - 1] + \lambda_1 [\mathbf{w}^H (\mathbf{e} + \boldsymbol{\delta}) - 1] \\ &+ \lambda_2 [\boldsymbol{\delta}^H \boldsymbol{\delta} - \varepsilon^2] \end{aligned}$$

$$\nabla_{\mathbf{w}^*} L = \mathbf{R} \mathbf{w} + \lambda_1 (\mathbf{e} + \boldsymbol{\delta}) = 0$$

$$\nabla_{\boldsymbol{\delta}^*} L = \lambda_1^* \mathbf{w} + \lambda_2 \boldsymbol{\delta} = 0$$



# Constraint on Errors is Operative

---

$$\lambda_1^* \mathbf{w} + \lambda_2 \boldsymbol{\delta} = 0$$

- From Appendix C, either

$$\lambda_2 = 0 \text{ or } \|\boldsymbol{\delta}\|^2 = \varepsilon^2$$

- Taking  $\lambda_2 = 0 \rightarrow \mathbf{w} = 0$
- So instead take  $\lambda_2 \neq 0$

$$\rightarrow \boldsymbol{\delta} = -\frac{\lambda_1^*}{\lambda_2} \mathbf{w}$$



# Eliminating the Explicit Errors

---

$$\delta = -\frac{\lambda_1^*}{\lambda_2} \mathbf{w}$$

$$\mathbf{R}\mathbf{w} = -\lambda_1(\mathbf{e} + \delta)$$

$$\mathbf{R}\mathbf{w} = -\lambda_1 \left( \mathbf{e} - \frac{\lambda_1^*}{\lambda_2} \mathbf{w} \right)$$

$$\mathbf{R}\mathbf{w} = -\lambda_1 \mathbf{e} + \frac{|\lambda_1|^2}{\lambda_2} \mathbf{w}$$



# Boring Manipulations

---

$$\mathbf{R}\mathbf{w} = -\lambda_1 \mathbf{e} + \frac{|\lambda_1|^2}{\lambda_2} \mathbf{w}$$

$$\left( \mathbf{R} - \frac{|\lambda_1|^2}{\lambda_2} \mathbf{I} \right) \mathbf{w} = -\lambda_1 \mathbf{e}$$

$$\mathbf{w} = -\lambda_1 \left( \mathbf{R} - \frac{|\lambda_1|^2}{\lambda_2} \mathbf{I} \right)^{-1} \mathbf{e}$$



# Yes, the Inverse is Well Defined

---

$$\mathbf{w}_\diamond = -\lambda_1 \left( \mathbf{R} - \frac{|\lambda_1|^2}{\lambda_2} \mathbf{I} \right)^{-1} \mathbf{e}(\vec{k})$$

- Can show  $\lambda_1 < 0, \lambda_2 < 0$ , so

$\mathbf{R} - \frac{|\lambda_1|^2}{\lambda_2} \mathbf{I}$  is invertible



# Running into Trouble

---

$$\mathbf{w}_{\diamond} = -\lambda_1 \left( \mathbf{R} - \frac{|\lambda_1|^2}{\lambda_2} \mathbf{I} \right)^{-1} \mathbf{e}(\vec{k})$$

- Computing  $\lambda_1, \lambda_2$  is difficult
- Several possible solutions for  $\lambda_1, \lambda_2$ ; must check to see which minimizes  $\mathbf{w}^H \mathbf{R} \mathbf{w}$
- $\lambda_1, \lambda_2$  change with  $\vec{k}$



# Punting

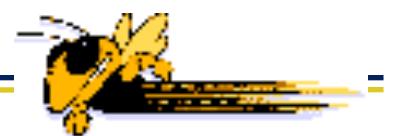
---

$$\mathbf{w}_{\diamond} = -\lambda_1 \left( \mathbf{R} - \frac{|\lambda_1|^2}{\lambda_2} \mathbf{I} \right)^{-1} \mathbf{e}(\vec{k})$$

- Let's try something ad-hoc:

$$\mathbf{w}_{\diamond} = (\mathbf{R} + \alpha \mathbf{I})^{-1} \mathbf{e}(\vec{k}), \alpha > 0$$

- “diagonal loading,” “ridge regression,” “Tikhonov regularization”

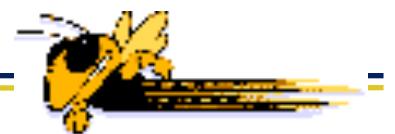


# Interpretation of Added Term

---

$$\mathbf{w}_\diamond = (\mathbf{R} + \alpha \mathbf{I})^{-1} \mathbf{e}(\vec{k}), \alpha > 0$$

- **Adding additional “white noise” to  $\mathbf{R}$  used in computing weights**
- **Remember sensitivity to errors increases with SNR**
- **Artificially decrease apparent SNR to decrease sensitivity**



# Alternative Problem Formulation

---

$$\min_{\mathbf{w}} \mathbf{w}^H \mathbf{R} \mathbf{w} \text{ s.t. } \mathbf{e}^H \mathbf{w} = 1$$

$$\text{and } \|\mathbf{w}\|^2 = \mathbf{w}^H \mathbf{w} \leq \beta$$

- Use Lagrange multipliers

$$\begin{aligned} L = & \mathbf{w}^H \mathbf{R} \mathbf{w} \\ & + \lambda_1^* [\mathbf{e}^H \mathbf{w} - 1] + \lambda_1 [\mathbf{e}^T \mathbf{w}^* - 1] \\ & + \lambda_2 [\mathbf{w}^H \mathbf{w} - \beta] \end{aligned}$$



# Taking Gradients

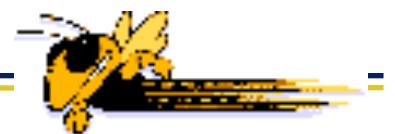
---

$$L = \mathbf{w}^H \mathbf{R} \mathbf{w}$$

$$\begin{aligned} &+ \lambda_1^* [\mathbf{e}^H \mathbf{w} - 1] + \lambda_1 [\mathbf{w}^H \mathbf{e} - 1] \\ &+ \lambda_2 [\mathbf{w}^H \mathbf{w} - \beta] \end{aligned}$$

$$\nabla_{\mathbf{w}^*} L = \mathbf{R} \mathbf{w} + \lambda_1 \mathbf{e} + \lambda_2 \mathbf{w} = 0$$

- If  $\lambda_2 = 0$ , get original MVDR  
(means  $\|\mathbf{w}\|^2 \leq \beta$  is inoperative)



# Results for Alternate Formulation

---

- If  $\lambda_2 \neq 0$ , we have

$$\mathbf{R}\mathbf{w} + \lambda_1\mathbf{e} + \lambda_2\mathbf{w} = 0$$

$$(\mathbf{R} + \lambda_2\mathbf{I})\mathbf{w} = -\lambda_1\mathbf{e}$$

$$\mathbf{w} = -\lambda_1(\mathbf{R} + \lambda_2\mathbf{I})^{-1}\mathbf{e}$$

- Same sort of structure as before!

