E9 231: Digital Array Signal Processing

Scribe: Manasa
Dept. of ECE
Indian Institute of Science
Bangalore 560 012, India
manasamadala@ee.iisc.ernet.in

Class held on: 17 Nov 2008

1 Announcements

1. Last HW: 7.2.2, 7.2.9 and 7.4.24
2. Matlab: 7.3.1, 7.4.1, 7.7.1

2 Topics

- Recursive Least Squares approach
- Recursive Least Squares error approach

3 Class Notes

3.1 Sample Matrix Inversion

$$\begin{aligned}
\hat{\mathbf{S}} &= \tfrac{1}{\mathrm{K}} \sum_{k=1}^{K} \mathbf{X}_k \mathbf{X}_k^H \\
\hat{\mathbf{w}}_{mpdr,smi} &= \frac{\hat{\mathbf{S}}_x^{-1}\mathbf{v}_s}{\mathbf{v}_s^H \hat{S}_x^{-1}\mathbf{v}_s} \\
\rho(k) &\triangleq \frac{\mathrm{SINR}_{smi}(K)}{\mathrm{SINR}_{mvdr}} \\
E(mathbfSINR_{smi}) &= \alpha \mathrm{SINR}_{mvdr} \\
K &= \frac{N}{1-\alpha}
\end{aligned} \tag{1}$$

4 Recursive Implementation of SMI

$$\begin{aligned}
\mathbf{X}_k &= \mathbf{v}_s F_k + N_k \\
Y_k &= \mathbf{w}_k^H \mathbf{X}_k \\
&= \mathbf{w}_k^H \mathbf{v}_s F_k + \mathbf{w}_k^H N_k \quad \text{(using } \mathbf{w}_k^H \text{ i.e adaptive beamforming)} \\
Y_k &= F_k + \tilde{N}_k \\
\tilde{N}_k &\triangleq \mathbf{w}_k^H N_k
\end{aligned} \tag{2}$$

4.1 Least Squares Approach

$$\xi_N(K) = \sum_{k=1}^{K} \mu^{K-k} |\tilde{N}_k|^2 \tag{3}$$

- $0 < \mu < 1$ typically close to 1
- Exponential weighting

Since we have assumed distortionless condition and F_k is deterministic Least squares approach reduces to minimisation of,

$$\begin{aligned}
\xi_Y(K) &= \sum_{k=1}^{K} \mu^{K-k} |Y_k|^2 \\
\text{st } \mathbf{w}_k^H \mathbf{v}_s &= 1
\end{aligned}$$

Legranges multiplier,

$$L = \sum_{k=1}^{K} \mu^{K-k} \mathbf{w}_K^H X_k X_k^H \mathbf{w}_K + \lambda(\mathbf{w}_K^H \mathbf{v}_s - 1) + \lambda^\star(\mathbf{v}_s^H \mathbf{w}_K - 1)$$

Differentiating wrt $\mathbf{w}_K^H$

$$\begin{aligned}
\hat{\mathbf{w}}_{mpdr}(k) &= \frac{\Phi^{-1}(K)\mathbf{v}_s}{\mathbf{v}_s^H \Phi^{-1}(K) \mathbf{v}_s} \\
\mathbf{\Phi}(K) &= \sum_{k=1}^{K} \mu^{(K-k)} \mathbf{X}_k \mathbf{X}_k^H
\end{aligned} \tag{4}$$

Note:
$\hat{\mathbf{w}}_{mpdr}(k)$ is the MPDR beamformer with $\mathbf{S}_X$ replaced by $\mathbf{\Phi}(k)$

$$\begin{aligned}
\xi_X(K) &= (\mathbf{v}_s^H \mathbf{\Phi}^{-1}(K) \mathbf{v}_s)^{-1} \\
&\triangleq \Lambda(K) \\
E(\mathbf{\Phi}(K)) &= \sum_{k=1}^{K} \mu^{K-k} \mathbf{S}_x \\
&= \frac{1-\mu^K}{(1-\mu)} \mathbf{S}_x \qquad \forall k
\end{aligned} \tag{5}$$

Hence $\mathbf{\Phi}(K)$ is a biased estimator of $\mathbf{S}_x$

4.2 MMSE approach

- This is similar to Least Squares BF solution
- Assume F_k is known (desired signal). Hence we donot have to insist on distortionless constraint.So we want to minimise,

$$\begin{aligned}
\xi_\mu(K) &= \sum_{k=1}^{K} \mu^{K-k}|\rho_k|^2 \qquad 0 < \mu \leqslant 1 \\
\rho_k &\triangleq F_k - \mathbf{w}_k^H X_k, \qquad k = 1, \ldots, K \\
\xi_\mu(K) &= \sum_{k=1}^{K} (F_k - \mathbf{w}_k^H X_k)^H (F_k - \mathbf{w}_k^H X_k) \\
&\text{Differentiating wrt } \mathbf{w}_k^H \\
\hat{\mathbf{w}}_{LSE}(K) &= \mathbf{\Phi}^{-1}(K)\mathbf{\Phi}_{XF^H}(K) \\
\text{where} \quad \mathbf{\Phi}_{XF^H}(K) &= \sum_{k=1}^{K} \mu^{(K-k)} X_k F_k^H \\
Y_{LSE}(K) &= \hat{\mathbf{w}}_{LSE}(K)^H X_K
\end{aligned} \tag{6}$$

Note:
Need to compute both $\Phi(K)$ and $\Phi_{XF^H}(K)$. But we donot need $\mathbf{v}_s$

5 Recursive Implementation of MPDR BF

Just need to find out $\mathbf{\Phi}^{-1}(k)$ interms of $\mathbf{\Phi}^{-1}(k-1)$

$$\begin{aligned}
\mathbf{\Phi}(k) &= \mathbf{\Phi}(k-1) + X_k X_k^H \\
&\text{Using matrix inversion lemma i.e:} \\
(A + BCD)^{-1} &= A^{-1} + A^{-1}B(DA^{-1}B + C^{-1})DA^{-1} \\
\therefore \qquad \mathbf{\Phi}^{-1}(k) &= \mu^{-1}\mathbf{\Phi}^{-1}(k-1) - \frac{\mu^{-2}\mathbf{\Phi}^{-1}(k-1)X_k X k^H \mathbf{\Phi}^{-1}(k-1)}{(1 + X_k^H \mu^{-1}\mathbf{\Phi}^{-1}(k-1)X_k)} \\
\mathbf{P}(k) &\triangleq \mathbf{\Phi}^{-1}(k) \\
\mathbf{g}(k) &\triangleq \frac{\mu^{-1}P(k-1)X_k}{(1 + \mu^{-1}X_k^H P(K-1)X_K)} \quad \text{Kalman gain factor} \\
\mathbf{P}(k) &= \mu^{-1}\mathbf{P}(k-1) - \mu^{-1}\mathbf{g}(k)X_k^H\mathbf{P}(k-1) \quad \text{RICCATTI eqn} \\
\hat{\mathbf{w}}_{mpdr}(k) &= \Lambda(k)\mathbf{P}(k)\mathbf{v}_s \\
\Lambda(k) &= (\mathbf{v}_s^H\mathbf{P}(k)\mathbf{v}_s)^{-1} \\
\hat{\mathbf{w}}_{mpdr}(k) &= \frac{\Lambda(k)}{(\mu\Lambda(k-1))}(I - \mathbf{g}(k)X_k^H)\hat{\mathbf{w}}_{mpdr}(k-1)
\end{aligned} \tag{7}$$

Intialisation:

$$\begin{aligned}
\mathbf{\Phi}(0) &= \sigma_0^2 I \\
\mathbf{w}_0 &= \frac{\mathbf{v}_s}{N}
\end{aligned} \tag{8}$$

(or any vector that satisfies the distortionless constraint)
Note:

$$\begin{aligned} \mathbf{\Phi}(K) &= \sum_{k=1}^{K} \mu^{K-k} X_k X_k^H + \mu^K \sigma_0^2 I \\ E(\mathbf{\Phi}(K)) &= \tfrac{(1-\mu^K)}{(1-\mu)} \mathbf{S}_x + \mu^K \sigma_0^2 I \\ \tilde{\mathbf{\Phi}}(K) &\triangleq \tfrac{(1-\mu)}{(1-\mu^K)} \mathbf{\Phi}(k) \\ \text{Then} \qquad E(\tilde{\mathbf{\Phi}}(K)) &= \mathbf{S}_x + \tfrac{(1-\mu)}{(1-\mu^K)} \mu^K \sigma_0^2 I \end{aligned} \tag{9}$$

- As K increases , the effect of diagonal loading decreases rapidly. So we may require an additional diagonal loading to ensure stability.

- $\mathbf{\Phi}_a(k)$ solves the diagonal loading problem
- But it cannot have a computationally simple recursive algorithm.
- So the solution is to use a moving window

5.1 Summary

1. Initialisation:

$$\begin{aligned} \hat{\mathbf{w}}_0 &= \frac{\mathbf{v}_s}{N} \\ \mathbf{P}_0 &= \tfrac{1}{\sigma_s^2} \mathbf{I} \end{aligned} \tag{10}$$

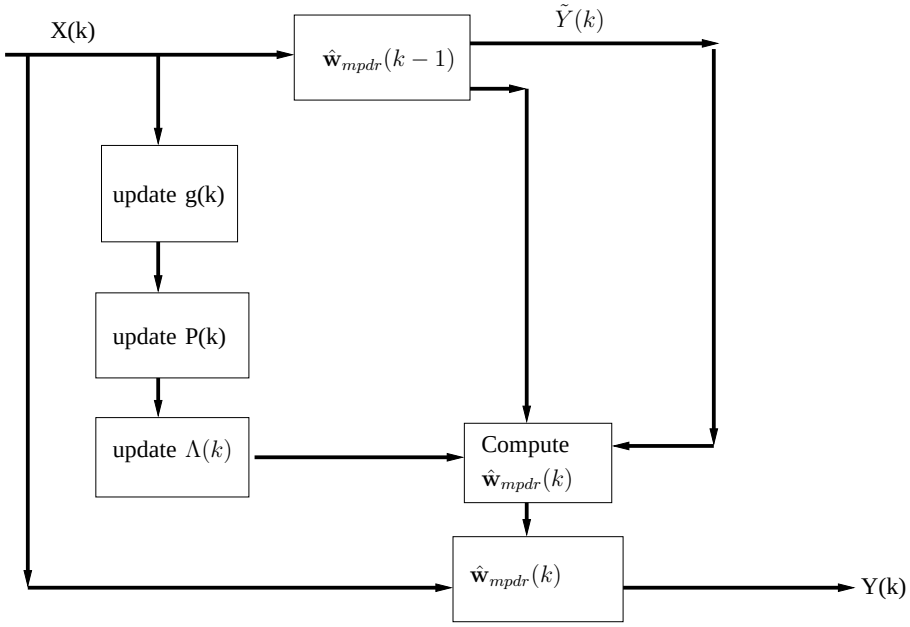
2. At each snapshot k = 1,2,3,... compute

$$\begin{aligned} \mathbf{g}(k) &\triangleq \frac{\mu^{-1} P(k-1) X_k}{(1 + \mu^{-1} X_k^H P(K-1) X_K)} \\ \mathbf{P}(k) &= \mu^{-1} \mathbf{P}(k-1) - \mu^{-1} \mathbf{g}(k) X_k^H \mathbf{P}(k-1) \\ \Lambda(k) &= (\mathbf{v}_s^H \mathbf{P}(k) \mathbf{v}_s)^{-1} \end{aligned} \tag{11}$$

3.

$$\begin{aligned} \hat{\mathbf{w}}_{mpdr}(k) &= \frac{\Lambda(k)}{(\mu \Lambda(k-1))} (I - \mathbf{g}(k) X_k^H) \hat{\mathbf{w}}_{mpdr}(k-1) \\ Y_k &= \hat{\mathbf{w}}_{mpdr}^H(k) X_k \\ \hat{\mathbf{w}}_{mpdr}(k) &= \frac{\Lambda(k)}{(\mu \Lambda(k-1))} (\hat{\mathbf{w}}_{mpdr}(k-1) - \mathbf{g}(k) \tilde{Y}^\star(k)) \\ \tilde{Y}_k &= \hat{\mathbf{w}}_{mpdr}^H(k-1) X_k \end{aligned} \tag{12}$$

Hence the name "gain factor" for $\mathbf{g}(k)$



5.2 Recursive Implementation of LSE BF

$$\begin{aligned}
\hat{\mathbf{w}}_{LSE}(k) &= \mathbf{\Phi}^{-1}(k)\mathbf{\Phi}_{XF^H}(k) \\
&= \mathbf{P}(K)\mathbf{\Phi}_{XF^H}(K) \\
\mathbf{\Phi}_{XF^H}(k) &= X_k F_k^H + \mu\mathbf{\Phi}_{XF^H}(k-1) \\
\hat{\mathbf{w}}_{LSE}(k) &= \hat{\mathbf{w}}_{LSE}(k-1) + \mathbf{g}(k)\mathbf{e}_p^H(k) \\
\mathbf{e}_p(k) &= F_k - \hat{\mathbf{w}}_{LSE}(k-1)X_k
\end{aligned} \tag{13}$$

5.3 Summary for LSE BF

1. Initialisation:

$$\begin{aligned}
\hat{\mathbf{w}}_0 &= \frac{\mathbf{v}_s}{N} \\
\mathbf{P}_0 &= \frac{1}{\sigma_s^2}\mathbf{I}
\end{aligned} \tag{14}$$

2. At each snapshot k = 1,2,3,... compute

$$\begin{aligned}
\mathbf{g}(k) &\triangleq \frac{\mu^{-1}\mathbf{P}(k-1)X_k}{(1+\mu^{-1}X_k^H)\mathbf{P}(k-1)X_K} \\
\mathbf{P}(k) &= \mu^{-1}\mathbf{P}(k-1) - \mu^{-1}\mathbf{g}(k)X_k^H\mathbf{P}(k-1)
\end{aligned} \tag{15}$$

3. Compute $\mathbf{e}_p(k) = F_k - \hat{\mathbf{w}}_{LSE}^H(k-1)X_k$
4. $\hat{\mathbf{w}}_{LSE}(k) = \hat{\mathbf{w}}_{LSE}(k-1) + \mathbf{g}(k)\mathbf{e}_p^H(k)$
5. $Y_k = \hat{\mathbf{w}}_{LSE}^H(k)X_k$

Note

Direct implematation needs knowledge of F_k. Typically we start with a known training sequence and later switch to a decision directed mode.

In contrast to MVDR and MPDR BFs, the LSE BF doesnot need knowledge of $\mathbf{v}_s$.