ROSP/DCOM1

Digital Information Processing

Part on ”Blind Equalization”

Philippe Ciblat

Telecom Paris, Institut Polytechnique de Paris, Palaiseau, France

General model

Goal: decode the data from $n = 0, \cdots, N$

$$y(n) = \sum_{\ell=0}^{L} h_\ell s_{n-\ell} + w(n)$$

with $w(n)$ white Gaussian

$$\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{w}$$

with $\mathbf{y} = [y(0), \ldots, y(N)]^{\mathrm{T}}$
and $\mathbf{s} = [\underbrace{s_{-L}, \cdots, s_{-1}}_{known}, s_0, \ldots, s_N]^{\mathrm{T}}$

Knowledge for the receiver to work:

- The data $\mathbf{y}$
- The channel matrix $\mathbf{H}$

Context

If H not known at the receiver side

- estimate it thanks to training sequence
- or blindly

Blind equalization: two approaches

- estimate **H** blindly
- directly estimate the equalizer **P** blindly

Here, find out P blindly

- SISO model: $y(n) = \sum_{\ell=0}^{L} h_\ell s_{n-\ell} + w(n)$
- Extension to MIMO model (source separation): straightforward

Second-order statistics

$$S_{yy}(e^{2i\pi f}) = \sum_{\tau} r_{yy}(\tau) e^{-2i\pi f \tau} = E_s |h(e^{2i\pi f})|^2$$

with $r_{yy}(\tau) = \mathbb{E}[y(t+\tau)\overline{y(t)}]$

Remarks

- No phase information
- **High order statistics are required!**

Let $\mathbf{x} = [x(1), \dots, x(N)]$ be a stationary random vector

$$M_X : \boldsymbol{f} \mapsto \mathbb{E}[e^{2i\pi \boldsymbol{f}^{\mathrm{T}} \mathbf{x}}] \quad \left(= \int p_X(\mathbf{x}) e^{2i\pi \boldsymbol{f}^{\mathrm{T}} \mathbf{x}} d\mathbf{x} \right)$$

s-order moments $\propto$ s-order coefficient of series expansion of M_X

$$C_X : \boldsymbol{f} \mapsto \log(M_X(\boldsymbol{f}))$$

s-order cumulants $\propto$ s-order coefficient of series expansion of C_X

High-order statistics

$$\underbrace{S_{yyyy}(e^{2i\pi f_1}, e^{2i\pi f_2}, e^{2i\pi f_3})}_{\text{trispectrum}} = \sum_{\tau_1,\tau_2,\tau_3} \text{cum}(\tau_1, \tau_2, \tau_3) e^{2i\pi(f_1\tau_1 + f_2\tau_2 + f_3\tau_3)}$$

$$\propto \quad h(e^{2i\pi f_1})\overline{h(e^{2i\pi f_2})h(e^{2i\pi f_3})}h(e^{2i\pi(-f_1+f_2+f_3)})$$

with $\text{cum}(\tau_1, \tau_2, \tau_3) = \text{cum}_4(y(n), y(n+\tau_1), \overline{y(n-\tau_2)}, \overline{y(n-\tau_3)})$

$\Rightarrow$ **Trispectrum contains phase information**

Problem

- How extracting phase information?
- This is the goal of high-order blind equalization algorithms

High-order algorithms

$$\min_{p} \mathbb{E}\left[f(z(n))\right]$$

with

- $z(n) = p \star y(n)$
- f a nonlinear and nonquadratic cost function

Sato Algorithm [Sato1975]

$$J = \mathbb{E}\left[(z(n) - \text{sign}(z(n)))^2\right]$$

Constant Modulus Algorithm (CMA) [Godard1980]

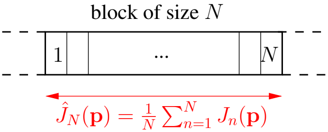
$$J = \mathbb{E}\left[\left(|z(n)|^2 - \text{Const}\right)^2\right]$$

with $\text{Const} = \mathbb{E}[|s_n|^4]/\mathbb{E}[|s_n|^2]$

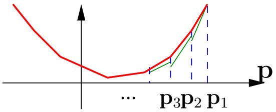
Implementation issue

How finding the minimum of $J(\mathbf{p}) = \mathbb{E}[J_n(\mathbf{p})]$?

Blockwise processing



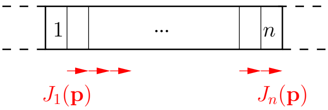
We replace $J(\mathbf{p})$ with $\hat{J}_N(\mathbf{p})$



Gradient algo.

$$\mathbf{p}_{i+1} = \mathbf{p}_i - \mu \frac{\partial \hat{J}_N(\mathbf{p})}{\partial \overline{\mathbf{p}}}\Big|_{\mathbf{p}_i}$$

Adaptive processing



We replace $J(\mathbf{p})$ with $J_n(\mathbf{p})$ at time/iteration n

- LMS
- Newton

(Stochastic) Gradient algo.

$$\mathbf{p}_{n+1} = \mathbf{p}_n - \mu \frac{\partial J_n(\mathbf{p})}{\partial \overline{\mathbf{p}}}\Big|_{\mathbf{p}_n}$$

Application to CMA

Adaptive implementation

$$\begin{aligned}\mathbf{p}_{n+1} &= \mathbf{p}_n - \mu\overline{\mathbf{y}_{L_p}(n)}z(n)(|z(n)|^2 - \text{Const}) \\ &= \mathbf{p}_n - \mu\overline{\mathbf{y}_{L_p}(n)}(z(n) - \text{fct}(z(n)))\end{aligned}$$

with

- $\mathbf{y}_{L_p}(n) = [y(n), \cdots, y(n - L_p)]^{\mathrm{T}}$
- $\text{fct}(z(n)) = z(n)(1 + \text{Const} - |z(n)|^2)$

Special case: training sequence (known s_n)

$$J = \mathbb{E}[|z(n) - s_n|^2]$$

Adaptive implementation

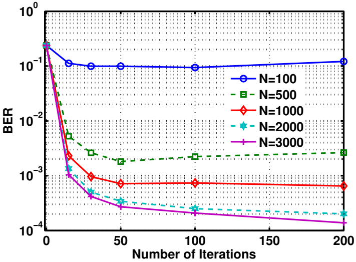
$$\mathbf{p}_{n+1} = \mathbf{p}_n - \mu\overline{\mathbf{y}_{L_p}(n)}(z(n) - s_n)$$

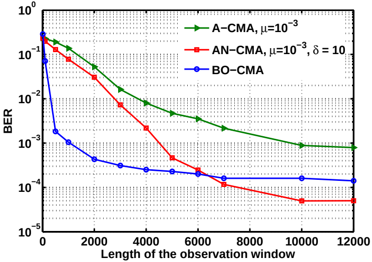
Remarks

- s_n replaced with $\hat{s}_n$ after initial convergence (Decision Directed)
- s_n replaced with $\text{fct}(z(n))$ which plays the role of “training”

Numerical illustrations: simulations

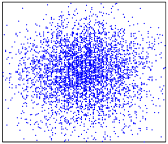
- PolMux 16-QAM, 112Gbits/s, range 1000km
- CD=1000ps/nm
- DGD=50ps
- OSNR=20dB



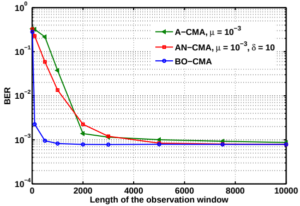


- Blockwise algorithm converges with $N = 1000$ and few iterations
- Adaptive algorithms need more samples to converge
- BER target (@10^{-3}) satisfied

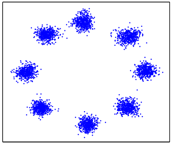
- PolMux 8-PSK, 60Gbits/s, range 800km
- SSMF fiber OSNR=23.7dB



⇒



⇒



It works!

References

[VanTrees2013] H. VanTrees, K. Bell, Detection Estimation and Modulation Theory Part I, Wiley, 2013

[Sato1975] Y. Sato, A method of self-recovering equalmization for multilevel amplitude modulation systems, IEEE Trans. on Communications, 1975

[Godard1980] D. Godard, Self-recovering equalization and carrier tracking in two-dimensional data communications systems, IEEE Trans. on Communications, Nov. 1980

[Savory2008] S. Savory, Digital filters for coherent optical receivers, Optics Express, 2008

[Selmi2011] M. Selmi, Advanced signal processing tools for QAM-based optical fiber communications, PhD thesis, Telecom ParisTech, 2011