1.5. Training-based Equalization — Adaptive Algorithm

- Principle: minimize

  $$J = E\{[\,s_{n-d} - \underline{f}^H \underline{X}(n)]^2\}$$

  recursively

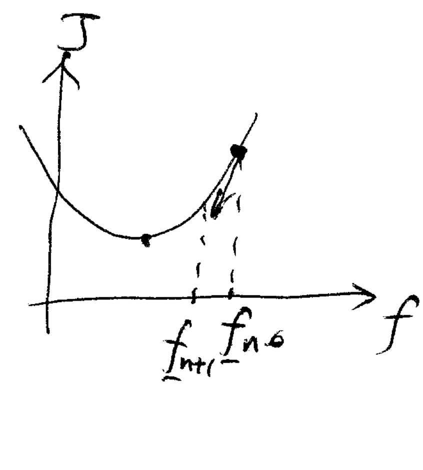
- Advantage:

  i) Reduce computational Complexity

  ii) Track time-variation

- Approach:

  $$\underline{f}_{n+1} = \underline{f}_n - \mu \frac{\partial J}{\partial \underline{f}_n^H}$$



- **Algorithm:**

  i) Initialization: $\underline{f}_0$

  ii) For $n = 1, 2, \cdots, M$ (training)

  $$\underline{f}_{n+1} = \underline{f}_n + \mu \left[s_{n-d} - \underline{f}_n^H \underline{X}(n) \right]^* \underline{X}(n)$$

  $$\varepsilon_n = s_{n-d} - \underline{f}_n^H \underline{X}(n)$$

  iii) Check SER

- Sample algorithms in MATLAB

- Draw plots of:

  1) Convergence in $|\varepsilon_n| \sim n$

  2) SER $\sim$ SNR

  3) SER $\sim$ M

2. Blind Equalization

- Remove training, improve bandwidth efficiency

2.d. • Estimate channel or equalizer with received samples and statistics of symbols

2.1. Adaptive blind equalization:

CMA (constant modulus algorithm)

- Symbol sequences have constant modulus, i.e., higher-order statistics, e.g.

$$R_2 = \frac{E[|S_n|^4]}{E[|S_n|^2]}$$

- Principle: force equalizer output to have the same modulus

$$\min J = E[(|\underline{f}^H \underline{x}(n)|^2 - R_2)^2]$$

- <u>Algorithms:</u>

i) Initialization $\underline{f}_0$

ii) For each $n = 1, 2, \cdots,$

$$\underline{f}_{n+1} = \underline{f}_n - \mu \cdot 2 [\underline{f}^H \underline{x}(n) - R_2] \underline{x}(n) \cdot x^H(n) \underline{f}_n$$

$$\varepsilon_n = |\underline{f}^H \underline{x}(n)|^2 - R_2$$

iii) Check SER.

- Try to implement the adaptive CMA in MATLAB

- Draw plots:
  1. Convergence $|\varepsilon_n|^2 \sim n$
  2. SER $\sim$ SNR
  3. SER $\sim$ sample amount