1.5. Training-based Equalization — Adaptive Algorithm

- **Principle:** minimize
  \[ J = E[(S_n - d - f^H X(w))^2]^\frac{3}{2} \]
  recursively

- **Advantage:**
  i) Reduce computational complexity
  ii) Track time-variation

- **Approach:**
  \[ f_{n+1} = f_n - \mu \frac{\partial J}{\partial f_n} \]
Algorithm:

i) Initialization: \( f_0 \)

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

\[
\begin{align*}
    f_{n+1} &= f_n + \mu [ S_{a-d} - f_n^H x(n) ] x^H(n) \\
    E_{n+1} &= S_{a-d} - f_n^H x(n)
\end{align*}
\]

iii) Check SER

Sample algorithms in MATLAB

Draw plots of:

1) Convergence in \( |E_n| \sim n \)

2) SER \sim \text{SNR}

3) SER \sim M
2. Blind Equalization

- Remove training, improve bandwidth efficiency

2.1. Estimate channel or equalizer with received samples and statistics of symbols

2.1.1. Adaptive blind equalization. CMA (constant modulus algorithm)

- Symbol sequences have constant modulus, i.e., higher-order statistics,
  \[ R_z = \frac{E[S_n^4]}{E[S_n^2]} \]
• Principle: force equalizer output to have the same modulus

\[ \min J = E(\left| f^H x(n) \right|^2 - R_2)^2 \]

• Algorithms:

  i) Initialization \( f_0 \)

  ii) For each \( n = 1, 2, \ldots \)

\[ f_{n+1} = f_n - \mu \cdot 2 \left[ f^H x(n) - R_2 \right] x(n) \cdot f_n \]

\[ E_n = \left| f^H x(n) \right|^2 - R_2 \]

  iii) Check SER.
• Try to implement the adaptive CMA in MATLAB

• Draw plots:
  1. Convergence $|\Delta u|^2 \sim n$
  2. SER $\sim$ SNR
  3. SER $\sim$ Sample amount