Finite-Length Equalization

- In practice, equalizers are usually implemented as FIR filters due to their:
  1) Inherent stability
  2) Better numerical properties than IIR filters (under finite-precision effects)
  3) Suitability for adaptive implementation
  4) Suitability for programmable DSP and VLSI implementations
- At any sample instant, an N-tap FIR equalizer processes a block of N received symbols. Hence, a vector channel model is appropriate for analysis.
Finite Length MMSE-LE

- Truncating the infinite length equalizers to the desired length is sub-optimum
- Instead, we derive MMSE equalizers under finite-length constraint

**Input – Output Model**

\[
y_k = \sum_{n=0}^{\nu} p_n x_{k-n} + n_k = \begin{bmatrix} p_0 & p_1 & \ldots & p_\nu \end{bmatrix} \begin{bmatrix} x_k \\ x_{k-1} \\ \vdots \\ x_{k-\nu} \end{bmatrix} + n_k
\]

(1)

Note: Unlike infinite-length case, we don’t assume an analog matched filter
Finite Length MMSE-LE

Note: For fractionally-spaced (oversampled) case, $y_k$, $p_k$ and $n_k$ are $l \times 1$ column vectors

Now collect $N_f$ output symbols which correspond to $N_f+\nu$ input symbols, Equation (1) becomes

$$
\begin{bmatrix}
  y_k \\
  y_{k-1} \\
  \vdots \\
  y_{k-N_f+1}
\end{bmatrix}_{N_f l \leftarrow Y_k} =
\begin{bmatrix}
  p_0 & p_1 & \cdots & p_\nu & 0 \\
  \vdots & \ddots & \ddots & \vdots & \vdots \\
  0 & p_0 & \cdots & p_\nu & 0
\end{bmatrix}_{N_f l \times (N_f + \nu) \leftarrow P}
\begin{bmatrix}
  x_k \\
  x_{k-1} \\
  \vdots \\
  x_{k-N_f-\nu+1}
\end{bmatrix}_{(N_f + \nu) \leftarrow X_k} +
\begin{bmatrix}
  n_k \\
  n_{k-1} \\
  \vdots \\
  n_{k-N_f+1}
\end{bmatrix}_{N_f l \leftarrow N_k}
$$
$e_k = x_{k-\Delta} - r_k = x_{k-\Delta} - w^*Y_k$

Orthogonality Principle:

$E[e_k Y_k^*] = \varphi$

$E[x_{k-\Delta} Y_k^*] = w_{opt}^* E[Y_k Y_k^*]$

Now $x_{k-\Delta} = \begin{bmatrix} 0 \ldots 0 & 1 & 0 \ldots 0 \\ \Delta & N_f + \nu - \Delta - 1 \end{bmatrix} X_k = e_{\Delta+1}^* X_k$

$\Rightarrow e_{\Delta+1}^* R_{XY} = w_{opt}^* R_{YY}$

where $0 \leq \Delta \leq N_f + \nu - 1$
\[
\Rightarrow w_{opt} = e_{\Delta+1}^* \underbrace{R_{XY} R_{YY}^{-1}}_{R_{XY}(\Delta)}
\]

\[
R_{XY}(\Delta) = e_{\Delta+1}^* E[X_k (X_k P^* + N_k^*)]
\]

\[
= e_X e_{\Delta+1}^* P^* = e_X \begin{bmatrix} 0 \ldots 0 \n \Delta \end{bmatrix} p_v^* \ldots p_0^* 0 \ldots 0
\]

\[
R_{YY} = e_XPP^* + R_{nn} = e_XPP^* + (l\sigma_n^2) I_{N_f l}
\]

oversampling

\[
\sigma_{FIR, MMSE-LE}^2 = e_X - R_{XY}(\Delta)R_{YY}^{-1}R_{YX}(\Delta) = e_X - w_{opt}^* R_{YX}(\Delta)
\]

choose \( \Delta \) to minimize \( \sigma_{FIR, MMSE-LE}^2 \)
\[ w_{opt} = \varepsilon_X e_{\Delta+1}^* P^* \left[ \varepsilon_X P P^* + R_{nn} \right]^{-1} \]

\[ = e_{\Delta+1}^* P^* \left[ P P^* + \frac{l}{SNR} I_{ln_f} \right]^{-1} \]

\[ \sigma_{FIR MMSE-LE}^2 = \varepsilon_X - \varepsilon_X e_{\Delta+1}^* P^* \left[ \varepsilon_X P P^* + \sigma_n^2 I_{ln_f} \right]^{-1} \varepsilon_X P e_{\Delta+1} \]

\[ = \varepsilon_X \left[ 1 - e_{\Delta+1}^* \left( P^* P + \frac{l}{SNR} I_{N_f + \nu} \right)^{-1} P^* P e_{\Delta+1} \right] \]
\[
= l \sigma_n^2 R^{-1}(\Delta + 1, \Delta + 1) \quad \text{(Using MIL)}
\]

where \( R = P^*P + \frac{l}{SNR} I_{N_f+v} \) \( \Leftarrow \) reminiscent of \( Q(D) + \frac{1}{SNR} \)

Therefore:
\[
\Delta_{opt} = \arg\min_{\Delta} R^{-1}(\Delta + 1, \Delta + 1)
\]

\[
SNR_{\text{MMSE–LE}} = \frac{\varepsilon_X}{\sigma_{\text{MMSE–LE}}^2} \quad \Leftarrow \text{biased}
\]

\[
SNR_{\text{MMSE–LE,U}} = SNR_{\text{MMSE–LE}} - 1
\]

\[
\gamma_{\text{MMSE–LE}} = \frac{SNR_{\text{MFB}}}{SNR_{\text{MMSE–LE,U}}} \quad \text{Loss from matched filter bound}
\]

ZF - LE is a special case \( (\sigma_n^2 \rightarrow 0) \)
Bias in FIR MMSE-LE

\[ z_k = \mathbf{w}_{MMSE-LE}^* Y \]

\[ = e_{\Delta+1}^* P^*[P P^* + \frac{l}{SNR} I]^{-1} (PX + N) \]

\[ = e_{\Delta+1}^* \left( P^* P + \frac{l}{SNR} I \right)^{-1} P^* (PX + N) \]

\[ = e_{\Delta+1}^* \left( \underbrace{P^* P + \frac{l}{SNR} I}_{R} \right)^{-1} \left( P^* PX + P^* N \right) \]

\[ = e_{\Delta+1}^* R^{-1} \left[ \left( R - \frac{l}{SNR} I \right) X + P^* N \right] \]
Bias in FIR MMSE-LE

\[ = e_{\Delta+1}^* X - e_{\Delta+1}^* \frac{l}{SNR} R^{-1} X + \tilde{N} \]

Bias factor multiplying \( x_{k-\Delta} \) is equal to

\[ 1 - \frac{l}{SNR} e_{\Delta+1}^* R^{-1} e_{\Delta+1} \]

\[ = 1 - \frac{l\sigma_n^2}{\varepsilon_X} \cdot \frac{\sigma_{\text{MMSE-LE}}^2}{l\sigma_n^2} \]

\[ = 1 - \frac{1}{SNR_{\text{MMSE-LE}}} = \frac{SNR_{\text{MMSE-LE},U}}{SNR_{\text{MMSE-LE}}} \]
Example

$1 + 0.9D^{-1}$ (or equivalently $0.9 + D$) channel, $N_f = 3$, $\varepsilon_X = 1$, $l = 1$, $\text{SNR}_{\text{MFB}} = 10$

\[
P = \begin{bmatrix}
0.9 & 1 & 0 & 0 \\
0 & 0.9 & 1 & 0 \\
0 & 0 & 0.9 & 1
\end{bmatrix}
\]

\[\Rightarrow 10 = \text{SNR} \cdot (1.81) \Rightarrow \text{SNR} = \frac{10}{1.81}\]

\[
R_{YY} = (PP^* + \frac{1}{\text{SNR}} I) = \begin{bmatrix}
1.991 & 0.9 & 0 \\
0.9 & 1.991 & 0.9 \\
0 & 0.9 & 1.991
\end{bmatrix}
\]

It can be shown that $\Delta_{opt} = 2$

Exercise: verify this in MATLAB

\[
R_{YY}(\Delta) = \varepsilon_X P e^{\Delta+1} = \varepsilon_X P \begin{bmatrix}
0 \\
0 \\
1 \\
0
\end{bmatrix} = \begin{bmatrix}
0 \\
1 \\
0.9
\end{bmatrix}
\]
Example

\[ \Rightarrow w = R_{YY}^{-1} R_{YX}(\Delta) = \begin{bmatrix} -0.23 \\ 0.51 \\ 0.22 \end{bmatrix} \]

\[ \sigma_{MMSE-LE}^2 = 1 - \left[ -0.23 \begin{array}{c} 0.51 \\ 0.22 \end{array} \right]_{w^*} \]

\[ R_{YX}(\Delta) = 0.294 \]

\[ SNR_{MMSE-LE,U} = \frac{1}{0.294} - 1 = 2.4 = 3.8 \text{dB} \]

6.2 dB below MFB and 1.9 dB below \( \infty \) - length MMSE - LE
FIR MMSE-DFE

FFF of length $N_f$, FBF of length $N_b$

Error Sequence

$$e_k = x_{k-\Delta} - \begin{bmatrix} b_1^* & b_2^* & \cdots & b_{N_b}^* \end{bmatrix} \begin{bmatrix} x_{k-\Delta-1} \\ \vdots \\ x_{k-\Delta-N_b} \end{bmatrix} + w^* Y_k$$

Let $b^* = \begin{bmatrix} 1 & -b_1^* & -b_2^* & \cdots & -b_{N_b}^* \end{bmatrix}$

$$X_{k-\Delta-N_b}^{k-\Delta} = \begin{bmatrix} x_{k-\Delta} \\ \vdots \\ x_{k-\Delta-N_b} \end{bmatrix}$$

and assume correct past decisions

$$e_k = b^* X_{k-\Delta-N_b}^{k-\Delta} - w^* Y_k$$
Orthogonality Principle

\[ E[e_k Y_k^*] = \phi \]
\[ \implies w^* R_{YY} = b^* \tilde{R}_{XY}(\Delta) \]
where

\[
\tilde{R}_{XY}(\Delta) = E \left[ \begin{array}{c}
 x_{k-\Delta} \\
 \vdots \\
 x_{k-N_b} \\
 x_{k-\Delta-N_b} \\
\end{array} \right] \left[ \begin{array}{cccc}
 x_k^* & x_{k-1}^* & \cdots & x_{k-N_f-\nu+1}^* \\
\end{array} \right] P^*
\]
def
\[ = \mathcal{E}_X J_{\Delta} P^* \]
Definition of $J_\Delta$

(i) $N_f + \nu - \Delta - N_b - 1 \geq 0 \iff \text{more common in practice}$

$$
\begin{bmatrix}
    x_{k-\Delta} \\
    \vdots \\
    x_{k-\Delta-N_b}
\end{bmatrix}
= \begin{bmatrix}
    O_{(N_b+1)\times\Delta} & I_{(N_b+1)} & O_{(N_b+1)\times(N_f+\nu-\Delta-N_b-1)}
\end{bmatrix}
\begin{bmatrix}
    x_k \\
    \vdots \\
    x_{k-\Delta-N_b} \\
    \vdots \\
    x_{k-N_f-\nu+1}
\end{bmatrix}
$$
Definition of $J_\Delta$

(ii) if $N_f + \nu - \Delta - N_b - 1 < 0 \text{ (performance typically bad)}$

Let $S = N_f + \nu - \Delta < N_b + 1$

$J_\Delta = \begin{bmatrix} O_{(N_b+1)\times(\Delta)} & \frac{I_{S\times S}}{O_{((N_b+1)-S)\times S}} \end{bmatrix}$

$\Rightarrow \sigma_e^2(\Delta) = E\left[(b^* X_{k-\Delta-N_b}^k - w^* Y_k) e_k^* \right]$

$= E\left[(b^* X_{k-\Delta-N_b}^k) (X_{k-\Delta-N_b}^k b - Y_k^* w) \right]$

$= b^* \left( \varepsilon_X I_{(N_b+1)} - \tilde{R}_{XY}(\Delta) R_{YY}^{-1} \tilde{R}_{YX}(\Delta) \right) b$
\[ b^* \left( \varepsilon_X I_{N_b+1} - \varepsilon_X J_\Delta P^* (\varepsilon_X PP^* + l\sigma_n^2 I)^{-1} \varepsilon_X PJ_\Delta^* \right) b \]

\[ = b^* \left( \varepsilon_X I_{N_b+1} - \varepsilon_X J_\Delta P^* \left( PP^* + \frac{l}{SNR} I \right)^{-1} PJ_\Delta^* \right) b \]

\[ = \varepsilon_X b^* J_\Delta \left( I_{N_b+1} - P^* \left( PP^* + \frac{l}{SNR} I \right)^{-1} P \right) J_\Delta^* b \]

Using Matrix Inversion Lemma

\[ (A + BCD)^{-1} = A^{-1} - A^{-1} B (DA^{-1} B + C^{-1})^{-1} DA^{-1} \]

\[ = l\sigma_n^2 \left\{ b^* J_\Delta (P^* P + \frac{l}{SNR} I)^{-1} J_\Delta^* b \right\} \]

\[ \sigma_e^2(\Delta) = l\sigma_n^2 b^* \tilde{Q}^{-1}(\Delta) b \quad \rightarrow \quad (*) \]

reminiscent of

\[ R_{ee}(D) = \frac{b(D)b^*(D^{-*})}{Q(D) + \frac{1}{\text{SNR}_{MFB}}} \]
Cholesky Factorization

Any Hermitian positive-definite matrix $R$ can be factored in the form $R = L \ D \ L^*$.

$$L^* = \begin{bmatrix}
1 & l_{12} & \cdots & \cdots \\
0 & 1 & l_{23} & \cdots \\
\vdots & 0 & \ddots & \vdots \\
0 & \cdots & 0 & 1
\end{bmatrix}$$

- Going back to Equation(*) and defining the factorization $\widetilde{Q}(\Delta) = L_\Delta D_\Delta L_\Delta^*$

$$\Rightarrow \widetilde{Q}^{-1}(\Delta) = L_\Delta^* D_\Delta^{-1} L_\Delta^{-1}$$
Cholesky (Triangular) Factorization

Any Hermitian positive-definite N×N matrix \( R \) can be factored in the form \( R = LDL^* \) where \( L \) is lower triangular monic and \( D \) is diagonal with positive elements \( d(1), d(2), \ldots, d(N) \)

**Algorithm**

Initialize: \( v(1) = R(1,1) \); \( R(2:N,1) = R(2:N,1) / v(1) \)

For \( j = 2 : N \),
   For \( k = 1 : j-1 \),
      \( v(k) = \text{conj}(R(j,k)) * R(k,k) \);
   end
   \( v(j) = R(j,j) - R(j,1:j-1) * v(1:j-1) \);
   \( R(j,j) = v(j) \);
   \( R(j+1:N,j) = (R(j+1:N,1:j-1) * v(1:j-1)) / v(j) \);

For \( k = 1 : N \),
   \( L(k,k) = 1 \);
   \( d(k) = R(k,k) \);
   For \( j = 1 : k-1 \),
      \( L(k,j) = R(k,j) \);
   end
end

See function

*Chol* in *MATLAB*
An important application of Cholesky factorization is the solution of systems of linear equations

\[ \mathbf{R}\mathbf{x} = \mathbf{y} \quad \text{Original linear system of equations} \]

\[ \mathbf{LDL}^* \mathbf{x} = \mathbf{y} \quad \text{Cholesky factorization of } \mathbf{R} \]

\[ \mathbf{LDv} = \mathbf{y} \Rightarrow \mathbf{L} \mathbf{v} = \mathbf{y} \quad \text{solve first triangular system of equations for } \mathbf{v} \text{ by back substitution} \]

\[ \mathbf{L}^* \mathbf{x} = \mathbf{v} \quad \text{solve second triangular system of equations for } \mathbf{x} \text{ by back substitution} \]
**Cholesky Factorization**

$D_{\Delta}$ consists of positive elements and $b$ has a "1" as its first element. Hence to minimize $\sigma_e^2(\Delta)$, choose

$$b^* = e_1^* L^*_\Delta$$  \hspace{1cm} \text{(first row of } L_{\Delta})$$

$$\Rightarrow b = L_{\Delta} e_1$$  \hspace{1cm} \text{Satisfies monicity constraint}

$$\Rightarrow \sigma_e^2(\Delta) = l \sigma_n^2 e_1^* L_{\Delta}^* \left( (L_{\Delta}^{-1})^* D_{\Delta}^{-1} L_{\Delta}^{-1} \right) L_{\Delta} e_1$$

$$= l \sigma_n^2 e_1^* D_\Delta^{-1} e_1 = l \sigma_n^2 D_\Delta^{-1} (1,1) = l \sigma_n^2 / D_\Delta (1,1)$$

choose $\Delta$ to minimize $\sigma_e^2(\Delta) \Rightarrow \Delta_{opt} = \arg \max_{\Delta} D_{\Delta} (1,1)$
Proof

Using Lagrange multipliers technique to minimize

\[ b^* \tilde{Q}(\Delta)^{-1} b \text{ subject to } b^*e_1 = 1 \]

\[ \Rightarrow b_{opt} = \frac{\tilde{Q}(\Delta)e_1}{e_1^*\tilde{Q}(\Delta)e_1} = \frac{L_\Delta D_\Delta L_\Delta^* e_1}{D_\Delta (1, 1)} = \frac{L_\Delta D_\Delta e_1}{D_\Delta (1, 1)} = \frac{D_\Delta (1, 1) L_\Delta e_1}{D_\Delta (1, 1)} = L_\Delta e_1 \]
FeedForward Filter

\[ w^* = b^* \tilde{R}_{XY}(\Delta) R_{YY}^{-1} \]

\[ = e_1^* L_\Delta^* J_\Delta P^* (PP^* + \frac{l}{\text{SNR}} I)^{-1} \]

\[ = e_1^* L_\Delta^* J_\Delta (P^* P + \frac{l}{\text{SNR}} I)^{-1} P^* \]

\( (a) \)

(tapped-delay line)

(matched filter)

\( (a) \) follows from identity: \( P^* (PP^* + \frac{l}{\text{SNR}} I)^{-1} = (P^* P + \frac{l}{\text{SNR}} I)^{-1} P^* \)
Important Special Case

When $N_b = \nu$ (no. feedback taps equals channel memory), $\Delta_{opt} = N_f - 1$

$$ J_\Delta = \begin{bmatrix} 0_{(N_b+1) \times \Delta_{opt}} & I_{N_b+1} \end{bmatrix} P^* P + \frac{l}{SNR} I = LDL^* $$

$$ \sigma_e^2 (\Delta) = l \sigma_n^2 \begin{bmatrix} 0_{1 \times (N_f-1)} & b^* \end{bmatrix} L^{-*} D^{-1} L^{-1} \begin{bmatrix} 0_{(N_f-1) \times 1} \\ b \end{bmatrix} $$

$$ \Rightarrow \begin{bmatrix} 0_{(N_f-1) \times 1} \\ b_{opt} \end{bmatrix} = L e_{N_f} \Rightarrow \sigma_{e,\text{min}}^2 = l \sigma_n^2 D^{-1} (N_f, N_f) $$

$$ w_{opt}^* = \begin{bmatrix} 0_{1 \times (N_f-1)} & b^* \end{bmatrix} (P^* P + \frac{l}{SNR} I)^{-1} P^* $$

$$ = e_{N_f}^* L^* (L^{-*} D^{-1} L^{-1}) P^* $$

$$ = D^{-1} (N_f, N_f) e_{N_f}^* L^{-1} P^* $$

Hint: post-cursor ISI span from $[\Delta + 1, N_f + \nu - 1] = [N_f, N_f + N_b - 1]$
**Cholesky Factorization**

\[
\begin{bmatrix}
1 & 0 \\
1 & 0 & 0 \\
\vdots & \ddots & \ddots & \ddots \\
x & \cdots & x & 1
\end{bmatrix}
\]

\[
\begin{bmatrix}
1 & 0 \\
0 & 1 & 0 & 0 \\
\vdots & \ddots & \ddots & \ddots \\
x & \cdots & x & 1
\end{bmatrix}
\]

\[
SNR = \frac{e_{N_f}^* L^{-1}}{N_0^2} \quad \xrightarrow{d_{N_f-1}} \quad SNR_{MMSE-DFE}
\]

\[
L^{-1} = \begin{bmatrix}
1 & 0 \\
x & \cdots & x & 1 & 0 & \cdots & 0 \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\
x & \cdots & x & 1 & 0 & \cdots & 0
\end{bmatrix}
\]

\[
e_{N_f}^* L^{-1} \quad \xrightarrow{d_{N_f-1}^{-1}} \quad P^* \quad \xrightarrow{w_{opt}^*}
\]

Feed-forward Filter

Tapped Delay Line

Matched Filter
Bias in MMSE-DFE

- Input to decision device is given by

\[
\hat{z}_k = x_{k-\Delta} - \tilde{b}_{opt}^* x + w^* H x + w^* n \\
\equiv x_{k-\Delta} - e_{+1}^* L^* x \\
+ d_{-1}^{-1} e_{+1}^* L^{-1} (L D L^* - \frac{1}{SNR} I_{N_f + \nu}) x \\
+ w^* n \\
= x_{k-\Delta} - \frac{1}{SNR_{MMSE-DFE}} e_{+1}^* L^{-1} x + w^* n \\
= (1 - \frac{1}{SNR_{MMSE-DFE}}) x_{k-\Delta} + \text{ISI term} + \text{noise term}
\]

- Bias factor is given by

\[
\alpha = 1 - \frac{1}{SNR_{MMSE-DFE}}
\]
Example

\(1 + 0.9D^{-1} \text{ channel, } \text{SNR}_{\text{MFB}} = 10\text{dB} \Rightarrow \text{SNR} = \frac{10}{1.81}\)

\(l = 1, N_f = 2, N_b = 1, \nu = 1, \Delta = 1 \quad \left(\text{when } N_b = \nu \right) \quad \left(\Delta_{\text{opt}} = N_f - 1\right)\)

\(N_f + \nu - \Delta - N_b - 1 = \phi \Rightarrow J_\Delta = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\)

\(P^*P + \frac{1}{\text{SNR}} I = \begin{bmatrix} 0.991 & 0.9 & 0 \\ 0.9 & 1.991 & 0.9 \\ 0 & 0.9 & 1.181 \end{bmatrix}\)

\(\Rightarrow \tilde{Q}^{-1}(\Delta) = J_\Delta (P^*P + \frac{1}{\text{SNR}} I)^{-1} J_\Delta\)
\[
\tilde{Q} = \begin{bmatrix} 2.0501 & -1.5623 \\ -1.5623 & 2.0373 \end{bmatrix}
\]

\[
\tilde{Q} = \begin{bmatrix} 1.1736 & 0.9 \\ 0.9 & 1.181 \end{bmatrix} = \text{LDL}^* 
\]

\[
\tilde{Q} = \begin{bmatrix} 1 & 0 \\ 0.7668 & 1 \end{bmatrix} \begin{bmatrix} 1.1735 & 0 \\ 0 & 0.4908 \end{bmatrix} \begin{bmatrix} 1 & 0.7668 \\ 0 & 1 \end{bmatrix}
\]

\[
\Rightarrow b_{opt} = \begin{bmatrix} 1 & 0.7668 \end{bmatrix}^t
\]

\[
\sigma_e^2 = \frac{\sigma_n^2}{1.1735} = \frac{1.81}{(10)(1.1735)} = 0.1542
\]

\[
\text{SNR}_{\text{MMSE-DFE,U}} = \frac{1}{0.1542} - 1 = 5.48 = 7.39\text{dB}
\]
which is 2.6dB below MFB but better
than $\infty$ - length MMSE - LE

$$w^*_\text{opt} = b^* J_\Delta (P^* P + \frac{l}{\text{SNR}} I)^{-1} P^*$$

$$= \begin{bmatrix} 1 & 0.7668 \end{bmatrix} \begin{bmatrix} -1.8618 & 2.0501 & -1.5623 \\ 1.4188 & -1.5623 & 2.0373 \end{bmatrix} \begin{bmatrix} 0.9 \\ 1 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0.1556 & 0.7668 \end{bmatrix}$$

Note: An alternative approach to DFE computation is
given in Section 3.7.4 of Cioffi's notes
DMT Performance

Notes: 1) This calculation is optimistic since N=8 is not sufficient for flat sub-channels approximation to hold 2) CIR is real, hence frequency response is conjugate symmetric

\[
h(D) = (1 + 0.9D^{-1}) \quad \text{SNR} = 10/1.81 \quad \text{FFT size} = 8 \quad SNR_{MFB} = 10
\]

Bandwidth & energy allocation:

E(0)=1/7; E(1)=2/7; E(2)=2/7; E(3)=2/7; E(4)=0 (not used).

Subchannel gains:

\[
H(m) = h_0 + h_1 e^{-j\frac{\pi m}{4}}
\]

Subchannel 0 has real constellation for symmetry conditions while Subchannels 1, 2, 3 use complex constellation

H(0)=1.9; H(1)=1.76 \angle 21.3^\circ, H(2)=1.35 \angle 42^\circ, H(3)=0.733 \angle 60.25^\circ

Subchannel SNRs:

\[
SNR(i) = N.E(i).SNR|H(i)|^2 : 1 \leq i \leq 3
\]

SNR(0)=13.6 dB; SNR(1)=12.9 dB; SNR(2)=10.6 dB; SNR(3)=5.3 dB

Decision-Point (Geometric) SNR:

\[
SNR_{geom} = 10\log_{10}(22.8*(19.5)^2*(11.4)^2*(3.4)^2)^{\frac{1}{N}} = 9\text{dB} \quad \text{dB} \quad \overline{N} = 7
\]

1 dB loss from MFB !!

Factor of N/(N+v) due to CP power loss (for fair comparison with single-carrier)
Performance Summary

\((1 + 0.9D^{-1})\) channel

- Matched Filter Bound : 10 dB
- Viterbi Detection : 10 dB
  (achieves MFB for any 2-tap channel!)
- DMT with size-8 FFT : 9 dB
- Infinite-length MMSE-DFE : 8.36 dB
- Infinite-length ZF-DFE : 7.42 dB
- FIR MMSE-DFE (Nf=3,Nb=1) : 7.91 dB
- FIR MMSE-DFE (Nf=2,Nb=1) : 7.39 dB
- FIR ZF-DFE (Nf=3,Nb=1) : 6.51 dB
- Infinite-length MMSE-LE : 5.7 dB
- FIR MMSE-LE \((N=3)\) : 3.79 dB
- Infinite-length ZF-LE : 0.2 dB

Can be improved with larger FFT size
Can be improved with transmitter optimization
Performance Results
Infinite-Length vs. Finite-Length MMSE-DFE

- As number of feedforward filter taps increases, Cholesky factorization converges to a spectral factorization and the feedback filter settings converge to the canonical spectral factor

- For this example, 5 feedforward taps result in less than 0.1 dB degradation from infinite number of feedforward taps
MMSE-DFE vs. MMSE-LE

- For same number of feedforward taps, MMSE-DFE is superior to MMSE-LE

- Typically, increasing number of feedforward taps beyond certain value results in small performance improvement. This value depends on channel charac. (ISI severity, min. phase ?)
Effect of Decision Delay

- Decision-point SNR could vary significantly with decision delay parameter especially for short filter lengths.

- For MMSE-LE, optimum delay is given by index of smallest diagonal element of $R^{-1}$.

- For MMSE-DFE, optimum delay is index of largest diagonal element in Cholesky factorization.
DFE Simulation

- Channel: \( h(D) = 1 + 0.9D^{-1} \)
- BPSK Input, white noise
- DFE with 3 FF taps and 1 FB tap
- Optimized Delay = 2 symbol periods
- Low-SNR Scenario: SNR = 10 dB
- High-SNR Scenario: SNR = 30 dB
- Histogram of received symbols at input and output of DFE
High-SNR Scenario

Input to Equalizer

Input to Slicer
Low-SNR Scenario

$h(D) = [0.9 \ 1]$, SNR = 10 dB

Input to Equalizer

Input to Slicer
On the Number of DFE Taps

• Longer filters generally improve performance but more complex

• For BPSK & QPSK, feedback filtering requires no multiplies

• Most of DFE complexity in feedforward filtering (full multiplies), especially when it is fractionally spaced

• Feedback filter length not more than channel memory

• For minimum-phase channels, feedforward filter is single tap!

• Channel delay spread (in symbol intervals) gives rough estimate of required number of taps
  • For LE : \( N_f \approx 2\nu \rightarrow 3\nu \)

Rules of Thumb !

• For DFE : \( N_b = \nu; N_f = \nu \rightarrow 2\nu \)
Packet-Based Data Transmission

**Example:** U.S. TDMA Digital Cellular Standard IS-136 (also known as D-AMPS)

- Data rate = 24 kbaud/sec = 48 kbits/sec
- Training sequence in each packet for channel estimate initialization
- 324 bits in a 6.7 msec time slot (burst), Symbol duration = 41.6 micro-seconds. Hence, no equalizer needed for delay spreads < 4 micro-second
- At 60 mph and 900MHz carrier: 80 Hz Doppler
- Coherence time 12.5 msec is comparable to slot duration therefore, channel tracking within burst is needed
Computational Complexity

• Measured in Million Instructions Per Second (MIPS)
  Instruction = 1 real multiply or add
• Useful complexity measure for programmable DSP implementations (estimate algorithm execution time)
• MIPS count dominated by no. of complex multiplies and external I/O access (maximize on-chip memory usage)

• Assumptions : Each complex multiply = 6 instructions

• MIPS = no. of instructions * update rate (in Hz)

• Implementation Efficiency = Theoretical MIPS/Measured MIPS
  (includes I/O access, data transfer,..)
Computational Complexity

- MIPS estimate for computing DFE coefficients and tracking channel estimate within each burst.

- For IS-136, equalizer is needed in hilly terrain (HT) environment to combat channel frequency selectivity due to multipath reflections and $\sqrt{RC}$ transmit and receive filters and to mitigate CCI.
Further Reduction in Computational Cost

- MMSE–DFE coefficients can be updated every $K$ symbol periods instead of every symbol period, without significant performance degradation.
- MIPS estimate is directly proportional to $K$. 

![Graph showing coefficient update rate vs MIPS estimate and No. of feedback taps.](image)
Extensions

Fractionally-spaced feedforward filter

- Channel output is sampled at an integer multiple of the symbol rate ($\frac{1}{T}$) to reduce sensitivity to sampling phase offsets
- Increase in computational complexity
Equalizer Computation Algorithms Summary

- Map the channel impulse response coefficients and the noise auto-correlation sequence into the equalizer coefficients.

- For infinite-length MMSE-DFE, requires computing a spectral factorization (root-finding routine).

- FIR MMSE-DFE used in practice (lower implementation complexity, easy to adapt, robust under finite precision).

- Computing FIR MMSE-DFE coefficients requires matrix inversion or Cholesky (triangular) factorization.
GSM

- **2G** digital cellular standard; evolved to EDGE/GPRS (2.5 G)
- Started as a European standard and has grown to become worldwide standard
- Most US operators replaced D-AMPS with GSM
- 3 current flavors (tri-band):
  - Original GSM (900 MHz)
  - DCS 1800 (at 1.8 GHz, in Europe)
  - PCS 1900 (at 1.9 GHz, primarily in US)
Technical Specs for PCS 1900

- **Bandwidth**: 200 kHz
- **Multiple Access**: combined TDMA/FDMA
- **Bit Rate**: 270.83 Kbps
- **Modulation**: Binary GMSK, $BT=0.3$
- **EDGE uses 8PSK**
- **8 time slots (users) per frequency sub-channel**
- **298 frequency channels with maximized spacing within each cell**
- **K=4 re-use factor**
- **Base station has 3 antennas each covering 1/3 sector of the cell**
TDMA Frame Structure

• Each frame consists of 8 time slots each of 0.577 msec duration for a total frame duration of 4.615 msec. Slot duration much less than channel coherence time even at 1.9 GHz (time non-selective)

• Each time slot consists of 26 training bits in the middle with 57 coded data bits on each side (in addition to control and guard bits) for a total of ~ 130 bits per slot. Bit duration about 4.5 micro-sec
Frequency Hopping

- To improve performance (especially in high interference scenarios), carrier frequency hopping is used to introduce frequency diversity so that users don’t experience long signal fading events.

- Hopping rate = \( 1/(4.615) \) msec = 216.7 hops/sec and provide \( \sim 2 \) dB SNR gain.
GSM Equalization

• Channel is frequency-selective (5-tap channel impulse response including ISI effects of Gaussian transmit filter)
• 26-bit midamble for channel estimation
• Equalization either using a DFE or a Viterbi equalizer with 16 states
• For EDGE: full Viterbi is formidable (8^4 states), hence reduced-complexity equalization is used (hybrid DFE/Viterbi) or channel shortening followed by Viterbi