Efficient Matrix Multiplication Methods to Implement a Near-optimum Channel Shortening Method for DMT Transceivers

Jeffrey Wu, Güner Arslan, Dr. Brian L. Evans
The University of Texas at Austin
Asilomar Conference
October 30, 2000
Why DMT?

Typical frequency response of ADSL channels

Typical noise spectrum of ADSL channels

Solution: Partition channel into subchannels using FFT
Simplest possible DMT transceiver

“–Houston, we have a problem…”
Solution to the ISI problem

- CP creates room between symbols
- TEQ shortens channel to fit within CP
Goal: Given a CP of length $\tau_0$, design the TEQ such that it shortens channel to $\tau_0 + 1$ sample periods.

Objection: But this is generally impossible to do perfectly!

Revised Goal: Design TEQ so that it minimizes ISI in such a way that maximizes channel capacity.

Response: Sounds good, but how do accomplish that?
The min-ISI method [Arslan, Evans, Kiaei, 2000]

**Observation:** Given a channel impulse response of $h$ and an equalizer $w$, there is a part of $h \ast w$ that causes ISI and a part that doesn’t.

- The length of the window is $\approx + 1$
- Heuristic determination of the optimal window offset, denoted as $\approx$, is given by Lu (2000).
Matrix ingredients of the min-ISI method

• The equalizer $\mathbf{w}$. This is a little vector.

• The convolution matrix $\mathbf{H}$, such that $(\mathbf{Hw})_k = (h \ast w)_k$.

• The windowing matrix $\mathbf{D}$. This is a diagonal matrix that isolates the part of $h \ast w$ causing ISI.

• The FFT matrix $\mathbf{Q}$. Takes FFT of $\mathbf{DHw}$.

• The weighting matrix $\text{diag}(\mathbf{S})$. 

\[
\sqrt{\mathbf{S}} \begin{bmatrix} \mathbf{Q} \\ \mathbf{D} \\ \mathbf{H} \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \mathbf{H} \\ \mathbf{D} \end{bmatrix}
\]
The Optimization Problem

Goal: Find \( w \) that minimizes \( \left\| \sqrt{\text{diag}(S)} Q D H w \right\|^2 \)

Translation: Find \( w \) that minimizes a weighted sum of the ISI power gains in each subchannel.

Something is missing here… a constraint!

Constrain \( \| G H w \|^2 = 1 \), where \( G = I - D \)

Translation: Prevent \( w \) from also minimizing the \textit{desired} part of the \( h \ast w \)!

The Matrix Multiplication Problem (finally!)

The optimization problem can be restated as:

Minimize $w^T A w$, where $A$ is defined as

$$
\begin{bmatrix}
H^T & D^T & Q^H & S & Q & D & H
\end{bmatrix} = A
$$

Subject to $w^T B w = 1$, where $B$ is

$$
\begin{bmatrix}
H^T & G^T & G & H
\end{bmatrix} = B
$$

$A$ and $B$ are small, so problem can be solved quickly so long as we can find $A$ and $B$.

Turns out to be a BIG problem for real-time implementation.
The Solution to the Problem – Sliding Windows

Arises frequently with Toeplitz matrices (i.e. $H$)

Sliding windows for single sums:

Sliding windows for double sums:
Fast Algorithm for Matrix $B$

Explicit formula:

$$B_{m,n} = \sum_{k=\Delta}^{\Delta+\nu} h_{k-m} h_{k-n}$$

The sliding window:

$$B_{m+1,n+1} = \sum_{k=\Delta-1}^{\Delta+\nu-1} h_{k-m} h_{k-n}$$

The recursion:

$$B_{m+1,n+1} = B_{m,n} - h_{\Delta+\nu-m} h_{\Delta+\nu-n} + h_{\Delta-1-m} h_{\Delta-1-n}$$

Very nice!
Fast Algorithm for Matrix A

Explicit formula:

\[
A_{m,n} = \sum_{k=0}^{\Delta-1} \sum_{l=0}^{\Delta-1} h_{k-m} h_{l-m} s_{k-l} + \\
\sum_{k=0}^{\Delta-1} \sum_{l=\Delta+\nu+1}^{N-1} h_{k-m} h_{l-m} s_{k-l} + \\
\sum_{k=\Delta+\nu+1}^{N-1} \sum_{l=0}^{\Delta-1} h_{k-m} h_{l-m} s_{k-l} + \\
\sum_{k=\Delta+\nu+1}^{N-1} \sum_{l=\Delta+\nu+1}^{N-1} h_{k-m} h_{l-m} s_{k-l}
\]

where \( s_{k-l} = \sum_{p=0}^{N-1} S_p e^{j2\pi(k-l)p/N} \)

The sliding window:

\[
C_{m+1,n+1} = \sum_{k=a-1}^{b-1} \sum_{l=c-1}^{d-1} h_{k-m} h_{l-m} s_{k-l}
\]

Recursive formula:

\[
C_{m+1,n+1} = C_{m,n} \\
- h_{b-m} h_{d-n} s_{b-d} \\
- h_{b-m} f(c,d,b,n) \\
- h_{d-n} f(a,b,d,m) \\
+ h_{a-1-m} h_{c-1-n} s_{a-c} \\
+ h_{a-1-m} f(c,d,a-1,n) \\
+ h_{c-1-n} f(a,b,d-1,m)
\]

where \( f(\alpha, \beta, \gamma, \delta) = \sum_{k=\alpha}^{\beta-1} h_{k-\delta} s_{k-\gamma} \)
Variants of the algorithm

Row-rotation method
• “Rotates” the rows in the convolution matrix $H$ to simplify the explicit formula of $A$ to one double sum.
• Assumes last few samples of impulse response close to zero. GOOD assumption.
• Virtually same performance as original method.

No-weighting method
• Assuming equal weighting of subchannels in the optimization problem. Equivalent to maximum SSNR method by Melsa, Younce, and Rohrs (1996).
• Simplifies calculation of $A$ to a single sum.
• Almost as good performance as original method.
## Results

<table>
<thead>
<tr>
<th>Method</th>
<th>Channel capacity %</th>
<th>SSNR (dB)</th>
<th>Complexity</th>
<th>MACs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original min-ISI</td>
<td>99.6</td>
<td>37.8</td>
<td>$\frac{1}{2} (N + ?) N_w (N_w + 1) + 5 N (N_w - 1) + NN_w$</td>
<td>132896</td>
</tr>
<tr>
<td>Recursive min-ISI</td>
<td>99.5</td>
<td>37.9</td>
<td>$4(N_w - 1) (N + 4N_w - ? - 2) + N_w (N + N_w - 1) + N$</td>
<td>44432</td>
</tr>
<tr>
<td>Row rotation min-ISI</td>
<td>99.5</td>
<td>37.5</td>
<td>$2(N_w - 1) (N + 2N_w - ? - 2) + N_w (N + N_w - 1) + N$</td>
<td>25872</td>
</tr>
<tr>
<td>Original max SSNR</td>
<td>97.9</td>
<td>58.9</td>
<td>$\frac{1}{2} N N_w (N_w + 1)$</td>
<td>78836</td>
</tr>
<tr>
<td>No-weighting min-ISI</td>
<td>97.8</td>
<td>55.4</td>
<td>$N N_w + 5 N_w (N_w - 1)$</td>
<td>10064</td>
</tr>
</tbody>
</table>

$N = \text{size of FFT}$

$N_w = \text{size of TEQ}$

$? = \text{size of cyclic prefix}$

Channel used: CSA loop 1

System margin: 6dB

$N = 512$, $N_w = 17$, $? = 32$