Algorithmic advances for software radios

Matteo Frigo

Vanu Inc.
Software radios

- Replace most communication hardware with software.
- (E.g., our demo GSM basestation runs on COTS Pentium III PC.)
- Software radios require new algorithms.
- “Hybrid” CCK demodulator for 802.11b: up to 4 times faster than standard demodulator.
- “Lazy” Viterbi algorithm: up to 10 times than Viterbi for CDMA.
CCK demodulation

- Demodulation is the bottleneck in a software 802.11b implementation.
- Standard maximum-likelihood demodulator based on the fast Walsh-Hadamard transform (FHT).
- Majority-logic demodulators are efficient but suboptimal.
- Our *Hybrid* algorithm is almost as fast as majority logic and almost as “optimal” as the FHT.
Significantly faster than the fast Walsh-Hadamard transform.
Performance of Hybrid algorithm

Negligible loss of optimality ($\leq 0.2$ dB).
CCK modulation

Input: 8 bits. Output: 8 “complex bits” (±1 or ±i, where \( i = \sqrt{-1} \)).

Map the 4 pairs of input bits into 4 complex bits \( \phi_0, \phi_1, \phi_2, \phi_3 \).

Output vector \( x(\phi) \), where:

\[
\begin{align*}
x_0 &= \phi_3 \\
x_1 &= \phi_3 \quad \phi_0 \\
x_2 &= \phi_3 \quad \phi_1 \\
x_3 &= \phi_3 \quad \phi_1 \quad \phi_0 \\
x_4 &= \phi_3 \quad \phi_2 \\
x_5 &= \phi_3 \quad \phi_2 \quad \phi_0 \\
x_6 &= \phi_3 \quad \phi_2 \quad \phi_1 \\
x_7 &= \phi_3 \quad \phi_2 \quad \phi_1 \quad \phi_0
\end{align*}
\]
CCK demodulators

- Maximum-likelihood: maximize correlation $|x(\phi) \cdot y|$ of received signal $y$ with transmitted signal $x(\phi)$, over all $\phi$. (Can be computed via fast Walsh-Hadamard transform.)

- Majority-logic: compute

  $\phi_0 \approx x_1/x_0 \approx x_3/x_2 \approx x_5/x_4 \approx x_7/x_6$;
  $\phi_1 \approx x_2/x_0 \approx x_3/x_1 \approx x_6/x_4 \approx x_7/x_5$;
  $\phi_2 \approx x_4/x_0 \approx x_5/x_1 \approx x_6/x_2 \approx x_7/x_3$. 
The Hybrid algorithm

• Compute the quantities

\[
\bar{\phi}_0 = (x_1x_0^* + x_3x_2^* + x_5x_4^* + x_7x_6^*) ; \\
\bar{\phi}_1 = (x_2x_0^* + x_3x_1^* + x_6x_4^* + x_7x_5^*) ; \\
\bar{\phi}_2 = (x_4x_0^* + x_5x_1^* + x_6x_2^* + x_7x_3^*) .
\]

Absent noise: the \( \bar{\phi}_k \)'s are purely real or imaginary.

• Let \( \phi_k \) be the real or imaginary axis closest to \( \bar{\phi}_k \).

• If \( |\angle \phi_k - \angle \bar{\phi}_k| \leq \alpha \) we are done. (\( \alpha \) is a magic number.)

• Otherwise, switch to maximum-likelihood demodulator.

Good value for \( \alpha = \arctan \left( \frac{2}{3} \right) \approx 33.7^\circ \).
Convolutional codes

Commonly used in TDMA/GSM cellular phones and other wireless standards.

Runtime of optimal decoders:

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Best case</th>
<th>Worst case</th>
</tr>
</thead>
<tbody>
<tr>
<td>Viterbi</td>
<td>$\Theta(2^k L)$</td>
<td>$\Theta(2^k L)$</td>
</tr>
<tr>
<td>$A^*$</td>
<td>$\Theta(L \log L)$</td>
<td>$\Theta(2^k L \log(2^k L))$</td>
</tr>
<tr>
<td>Our “Lazy Viterbi”</td>
<td>$\Theta(L)$</td>
<td>$\Theta(2^k L)$</td>
</tr>
</tbody>
</table>

$(k = “\text{constraint length”}, \ L = \text{input size})$)

For $A^*$ and Lazy Viterbi, runtime depends on SNR.
## Convolutional decoders

<table>
<thead>
<tr>
<th>Decoder</th>
<th>$k$</th>
<th>Pentium III cycles/bit</th>
<th>PowerPC 7400 cycles/bit</th>
<th>StrongARM cycles/bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lazy</td>
<td>6</td>
<td>201</td>
<td>200</td>
<td>226</td>
</tr>
<tr>
<td>Viterbi Optimized</td>
<td>6</td>
<td>316</td>
<td>239</td>
<td>310</td>
</tr>
<tr>
<td>Karn Unoptimized</td>
<td>6</td>
<td>1143</td>
<td>626</td>
<td>892</td>
</tr>
<tr>
<td>Lazy</td>
<td>7</td>
<td>205</td>
<td>203</td>
<td>232</td>
</tr>
<tr>
<td>Karn Optimized</td>
<td>7</td>
<td>558</td>
<td>486</td>
<td>641</td>
</tr>
<tr>
<td>Karn Unoptimized</td>
<td>7</td>
<td>2108</td>
<td>1094</td>
<td>1535</td>
</tr>
<tr>
<td>Karn SSE</td>
<td>7</td>
<td>108</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Lazy</td>
<td>9</td>
<td>235</td>
<td>225</td>
<td>343</td>
</tr>
<tr>
<td>Karn Unoptimized</td>
<td>9</td>
<td>8026</td>
<td>3930</td>
<td>5561</td>
</tr>
<tr>
<td>Karn SSE</td>
<td>9</td>
<td>310</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

($k =$ constraint length.)
Conclusion

- Software radios require new algorithms.
- Noise-adaptive algorithms can save power and improve battery life.
- The flexibility of software radios allows the best algorithm to be chosen at run time.