

Algorithmic advances for software radios

Matteo Frigo

Vanu Inc.

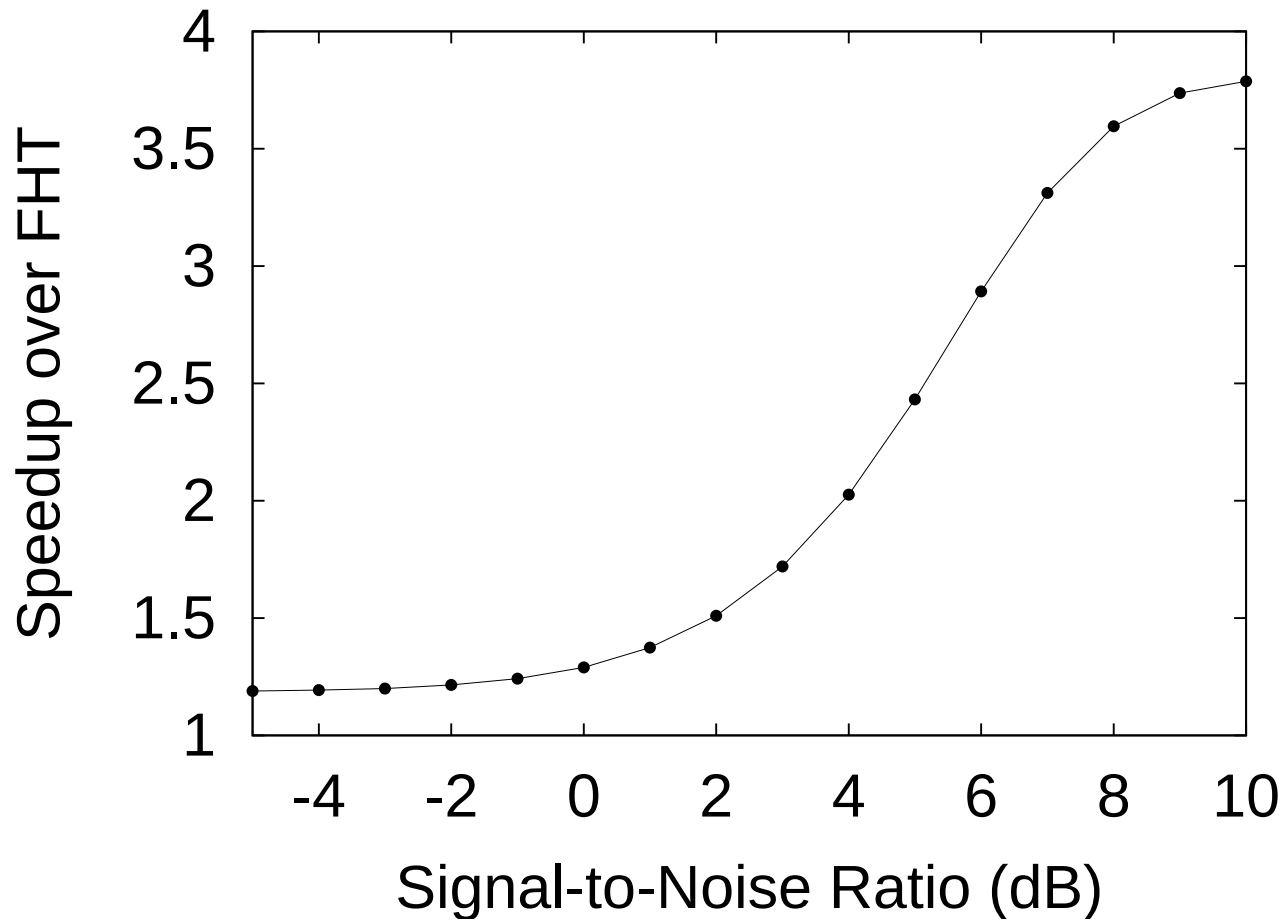
Software radios

- 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

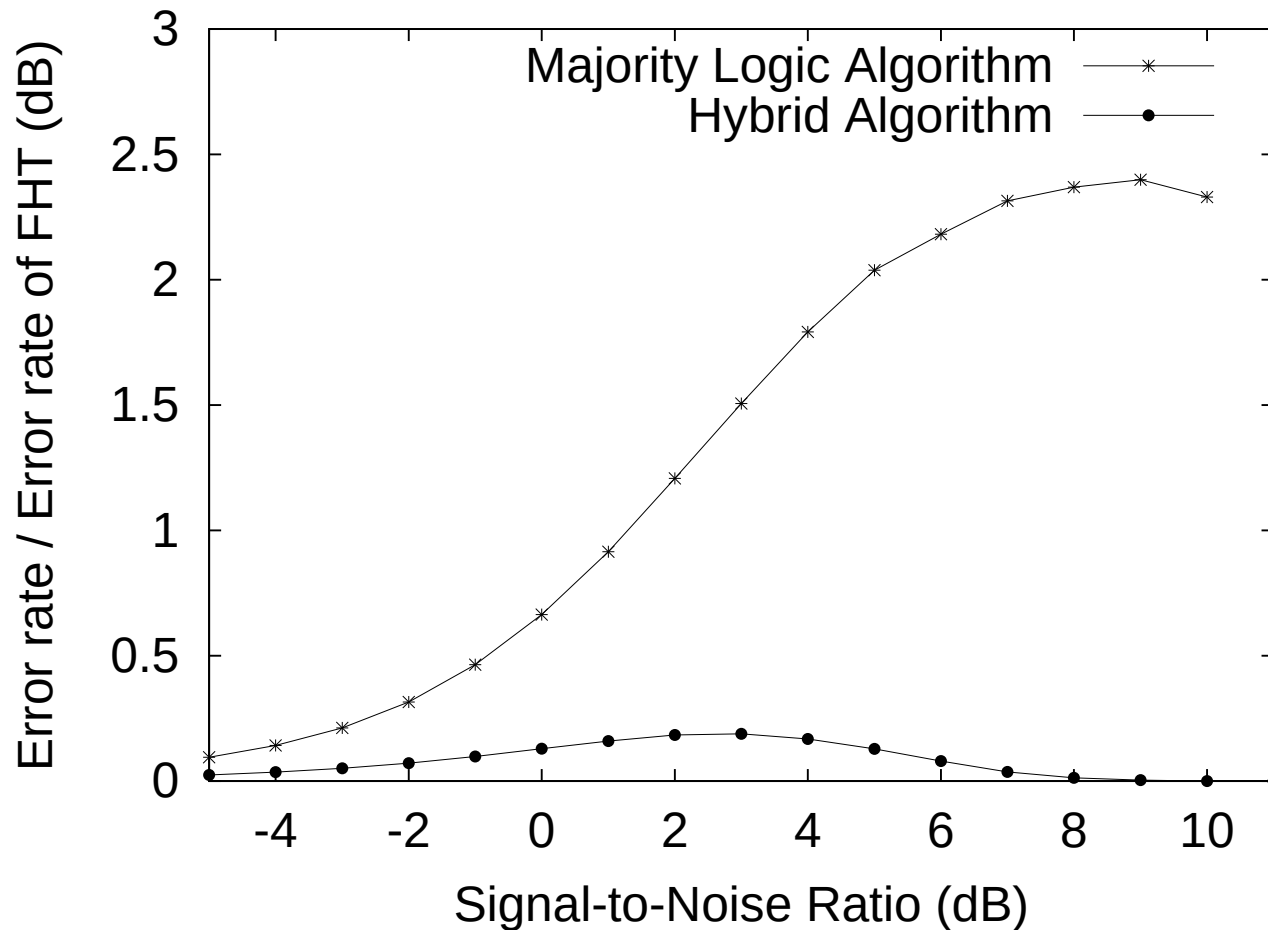
- Demodulation is the bottleneck in a software 802.11b implementation.
- Standard maximum-likelihood demodulator based on the fast Walsh-Hadamard transform (FHT).
- Majority-logic demodulators are efficient but suboptimal.
- Our *Hybrid* algorithm is almost as fast as majority logic and almost as “optimal”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

Speed of Hybrid algorithm



Significantly faster than the fast Walsh-Hadamard transform.

Performance of Hybrid algorithm



Negligible loss of optimality (≤ 0.2 dB).

CCK modulation

Input: 8 bits. **Output:** 8 “complex bits” (± 1 or $\pm i$, 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:

$$\begin{aligned}x_0 &= \phi_3 \\x_1 &= \phi_3 \quad \phi_0 \\x_2 &= \phi_3 \quad \phi_1 \\x_3 &= \phi_3 \quad \phi_1 \quad \phi_0 \\x_4 &= \phi_3 \quad \phi_2 \\x_5 &= \phi_3 \quad \phi_2 \quad \phi_0 \\x_6 &= \phi_3 \quad \phi_2 \quad \phi_1 \\x_7 &= \phi_3 \quad \phi_2 \quad \phi_1 \quad \phi_0\end{aligned}$$

CCK demodulators

- Maximum-likelihood: maximize correlation $|x(\phi) \cdot y|$ of received signal y with transmitted signal $x(\phi)$, over all ϕ . (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

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

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

- Let ϕ_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. (α is a magic number.)
- Otherwise, switch to maximum-likelihood demodulator.

Good value for $\alpha = \arctan(2/3) \approx 33.7^\circ$.

Convolutional codes

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

Runtime of optimal decoders:

Algorithm	Best case	Worst case
Viterbi	$\Theta(2^k L)$	$\Theta(2^k L)$
A^*	$\Theta(L \log L)$	$\Theta(2^k L \log(2^k L))$
Our “Lazy Viterbi”	$\Theta(L)$	$\Theta(2^k L)$

(k = “constraint length”, L = input size.)

For A^* and Lazy Viterbi, runtime depends on SNR.

Convolutional decoders

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

(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.