Algorithmic advances for software radios

Matteo Frigo
Vanu Inc.*

May 31, 2002

A software radio is a communication system in which most hardware is replaced by software, much in the same way as analog circuits were replaced by digital signal processors in the past 30 years.

Software radios are fundamentally different from hardware radios, and new algorithmic ideas are needed to make them viable. Consider for example the IEEE 802.11b standard for wireless local area networks. This standard uses a modulation scheme known as Complementary Code Keying (CCK). The canonical maximum-likelihood CCK demodulator [6] uses an algorithm based on the fast Walsh-Hadamard transform. (Indeed, the existence of such fast algorithm is a good reason for choosing this particular modulation scheme.) In order to produce 8 bits of output, the demodulator uses about 500 integer arithmetic operations, which amounts to 687.5E6 operations per second at the standard data rate of 11 Mbps without counting the overhead of loops, function call, pipeline stalls, and register spills. This computational load is onerous for a software implementation of the 802.11b standard. My best implementation of the CCK demodulator requires about 1375E6 cycles per second of data on a state-of-the-art Athlon 1700+ processor. This implementation was obtained by means of techniques similar to those used in FFTW [4] and it is probably very close to being optimal. On the other hand, an alternative demodulation algorithm that we developed runs about 4 times faster than the algorithm from [6], but while the new algorithm works fine in software, it would not map nicely into a hardware implementation.

This paper reports some recent progress made by Vanu Inc. in developing algorithms for software radios. Specifically, in Section 1 we discuss the new CCK demodulator, and in Section 2 we discuss a new maximum-likelihood decoder for convolutional codes, which is meant to replace the celebrated Viterbi algorithm [9, 3] in our software radios.

The two algorithms share the following feature: The running time is not constant, but it depends on the signal-to-noise ratio (SNR) of the input data. The lower the noise, the faster the algorithm. This behavior may at first appear to be intolerable in a real-time environment, but in fact, communication standards define a minimum SNR at which compliant communcation systems are expected to operate properly. As long as the algorithm is fast enough at the worst-case SNR, the system is correct. Moreover, while the system must still be dimensioned for worst-case behavior, a noise-adaptive strategy can save cycles when the noise is low, therefore extending battery life.

1 A novel CCK demodulator

Complementary Code Keying (CCK) is the modulation scheme used in the IEEE 802.11b wireless standard. We developed a CCK demodulator that combines the best features of the maximum-likelihood demodulator from [6] with the majority-logic demodulators of [8, 5].[1] Maximum-likelihood algorithms enjoy optimal error rate but are slow, while majority-logic algorithms are fast at the expense of a higher error rate. For the case of CCK, the error rate of the majority-logic demodulator of [8] can be up to 2.4 dB worse than optimal. Our *Hybrid* algorithm runs at majority-logic speed in the range of interest (SNR ≥ 8) while incurring at most a 0.2 dB error-rate degradation over the whole range. (See Figure 1.)

We cannot explain the details of the algorithm in the constrained space of this abstract. The basic idea can be simply stated, however. Upon receiving a block of data, the Hybrid algorithm first runs a majority-logic demodulator. Then, it determines whether the answer is “reliable.” If it is, then the algorithm terminates, otherwise it discards the answer and it runs the maximum-likelihood algorithm from scratch. The tricky part in this approach is to detemine whether the majority-logic answer is “reliable” or not. A simple reliability criterion exists [2] that yields the results shown in Figure 1. For this criterion, we obtained an analytical expression of the additive symbol error rate of the Hybrid algorithm compared to an optimal algorithm. Overall, the symbol error rate of the Hybrid algorithm is at most an additive $2^{-\Omega(n)}$

---

*1 Porter Square, Suite 18, Cambridge, MA 02140. athena@vanu.com

[1] In fact, the algorithm applies to the wider class of first-order Reed-Muller codes, but in this paper we focus on CCK only.

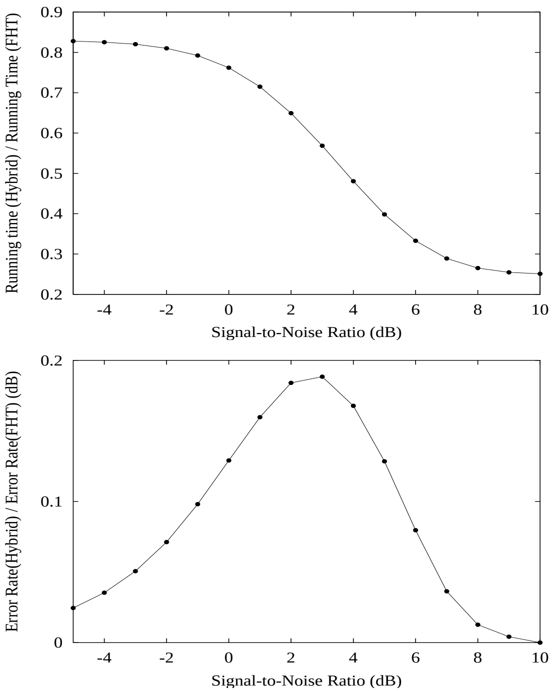

Figure 1: Experimental results for the Hybrid algorithm applied to CCK demodulation. **Top:** Running time of the Hybrid algorithm as a fraction of the running time of the maximum-likelihood fast Hadamard transform (FHT) algorithm. **Bottom:** Ratio of the error rate of the Hybrid algorithm over the error rate of the FHT, expressed in dB. Experiments performed on 1 billion encoded blocks of data subject to a simulated AWGN channel. Both algorithms ran on a Pentium III 1 GHz processor.

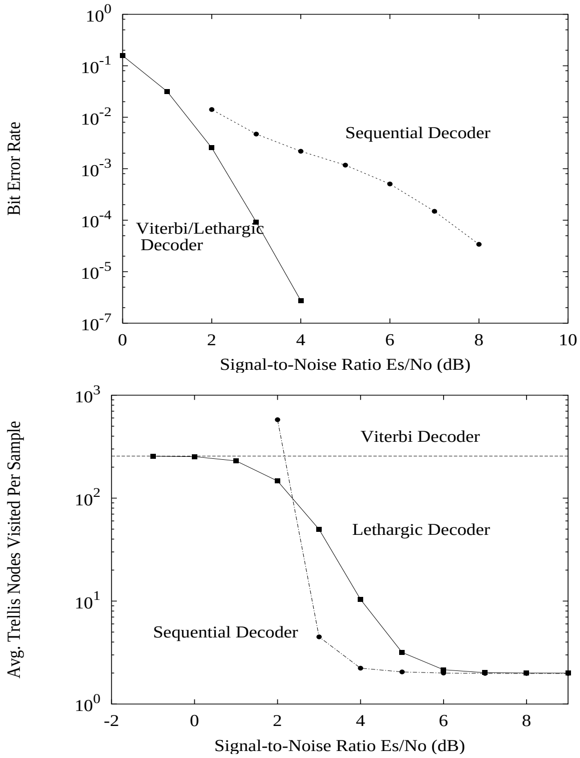

Figure 2: **(Top)** *Bit error rate;* **(Bottom)** *Average number of explored nodes per information symbol. Both are given as a function of the SNR, for the Lethargic Viterbi, Viterbi and Sequential decoders, under AWGN. The code is a rate-*$1/2$*, constraint length* 9 *code used in CDMA, generator polynomials (753,541) (octal). For the sequential decoder, experiments were performed on blocks of 100 encoded information bits.*

worse than that of an optimal algorithm, where n is the length of a codeword.[2]

We further remark that the Hybrid algorithm adapts to the noise conditions implicitly, without any explicit measurement of the noise level.

2 A fast maximum-likelihood decoder for convolutional codes

Convolutional error-correcting codes are used in many communication standards, including TDMA, GSM, and IS-95 (CDMA) cellular phones, and they are usually decoded by the venerable Viterbi algorithm [9, 3]. This algorithm is simple, allows for compact hardware implementations, and its error rate is optimal. A software implementation of the Viterbi decoder can also be made very efficient.

Unfortunately, the execution time of the Viterbi decoder grows exponentially with a certain parameter (the “constraint length”) of the error-correcting code being used. This parameter is 6 for TDMA cellular phones (which is still manageable), and 9 for IS-95 (which is too high). To overcome this problem, other decoder structures, such as sequential decoders [1], have been investigated in the literature. Under good Signal-to-Noise Ratio (SNR) conditions, these decoders are more efficient than the Viterbi algorithm. In addition to being suboptimal, at low SNR these decoders become prohibitively slow [1], however.

Our *lethargic Viterbi*[3] decoder exhibits the same error-correcting properties as the Viterbi decoder, but it runs much faster when the SNR is “high”. (See Figure 2.) In the limit SNR $\to \infty$, the lethargic algorithm runs in constant time per output symbol, irrespective of the constraint length.

The Viterbi algorithm computes the shortest path on a certain structured graph called a “trellis”. The lethargic Viterbi algorithm computes the same shortest path by means of

---

[2] The notation $\Omega(n)$ denotes some function that grows *at least* linearly with n.

[3] The name derives from the fact that the algorithm avoids doing anything until strictly required to do so, and it also postpones actions that have to be performed anyway.

A^* search [7], but the A^* algorithm was modified to exploit the structure of the trellis. In particular, the lethargic algorithm expands a node in constant time, while A^* generally incurs a slowdown logarithmic in the size of the graph. This simplification is a consequence of the fact that the weights on the graph edges can be assumed to be (small) integers and of the particular topology of the trellis.

References

[1] R. M. Fano. A heuristic discussion of probabilistic decoding. *IEEE Transactions on Information Theory*, IT-9:64–73, 1963.

[2] Jon Feldman, Matteo Frigo, and Ibrahim Abou-Faycal. A noise-adaptive strategy for first-order Reed-Muller decoding. In *Proceedings of the IEEE Semiannual Vehicular Technology Conference*, Fall 2002. To appear.

[3] G. D. Forney. The Viterbi algorithm. *Proceedings of the IEEE*, 61:268–278, 1973.

[4] Matteo Frigo and Steven G. Johnson. FFTW: An adaptive software architecture for the FFT. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 1381–1384, Seattle, WA, May 1998.

[5] K. Paterson and A. Jones. Efficient decoding algorithms for generalised Reed-Muller codes. Technical report, Hewlett-Packard Labs, November 1998.

[6] Bob Pearson. Complementary code keying made simple, application note 9850, May 2000. `http://www.intersil.com/data/an/an9/an9850/an9850.pdf`.

[7] S. Russel and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice-Hall, Upper Saddle River, New Jersey, 1995.

[8] R. van Nee. OFDM codes for peak-to-average power reduction and error correction. In *Proc. IEEE Globecom ’96, London, England*, pages 740–744, November 1996.

[9] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Inform. Theory*, IT-13:260–269, April 1967.