Fast Block Noncoherent Decoding

WIM SWELDENS

Bell Laboratories, Lucent Technologies
600 Mountain Avenue, Murray Hill, NJ 07974
wim@lucent.com

November, 1999

Abstract

We present a fast algorithm for exact maximum likelihood multi-symbol noncoherent PSK decoding. While the standard algorithm is exponential in the rate and the block length, our algorithm is rate independent and linear-logarithmic in the block length.

Index Terms—Noncoherent decoding, differential modulation, phase shift key

1 Introduction

We consider communication over a fading channel where the fading is unknown but approximately constant over multiple symbol periods. Coherent decoding requires explicit learning of the channel and its overhead may be excessive especially in mobile or multiple antenna settings. Standard differential modulation techniques like DMPSK encode the data in the phase *difference* between two consecutive PSK symbols [1]. However, the performance of noncoherent differential decoding is inferior to coherent decoding; in [2] it is shown that when the number phases used (M) gets large DMPSK suffers from a 3dB performance loss compared to coherent MPSK decoding. By using maximum likelihood (ML) decoding over blocks of multiple symbols a portion of this loss can be recuperated [2] . However, the computational complexity of the standard ML algorithm is exponential in both the rate $R = \log_2 M$ and the block length T. In [3] a linear complexity algorithm ($O(T)$) that *approximately* computes the ML answer was introduced. In this paper we present a linear-logarithmic ($O(T \log T)$) algorithm for *exact* ML noncoherent decoding. We also show that for the approximate algorithm to have a constant approximation quality, its complexity needs to grow quadratically ($O(T^2)$).

2 Channel Model

We use complex baseband notation: at time t we transmit the signal s_t and we receive the signal x_t at the receiver antenna. We assume a Rayleigh fading channel given by:

$$x_t = \sqrt{\rho}\, h_t\, s_t + w_t, \qquad t = 0, 1, \ldots . \tag{1}$$

The additive noise w_t is independent, identically complex Gaussian distributed $\mathcal{CN}(0,1)$. The complex-valued fading coefficient h_t is $\mathcal{CN}(0,1)$ distributed but not necessarily independent. The signals are normalized to have average energy one: $\mathbf{E}|s_t|^2 = 1$; then ρ represents the expected signal-to-noise ratio (SNR) at the receiver.

For a data rate of R bits per channel use, we need $M = 2^R$ symbols. A common technique is Phase Shift Key (PSK) which uses symbols that are Mth roots of unity

$$v_m = e^{2\pi i m/M} \qquad m = 0, \ldots, M-1. \tag{2}$$

Known channel: Suppose we want to send a data sequence of integers $z_1, z_2, \ldots$ with $z_t \in \{0, \ldots, M-1\}$. In case we assume that the receiver knows the fading coefficient, then the transmitter simply sends

$$s_t = v_{z_t}.$$

The ML coherent decoder is given by

$$\widehat{z}_t = \arg \max_{0 \leqslant m < M} |x_t - h_t v_m| = \lfloor \arg\left(x_t/h_t\right) M/(2\pi) \rceil\ . \tag{3}$$

Here $\arg$ is the phase of a complex number and $\lfloor x \rceil$ stands for the integer closest to x: $\lfloor x \rceil = \lfloor x + 1/2 \rfloor$. In this case the decoding is done per symbol and there is no advantage from using block decoding. Note that the decoding depends only on the phase of the fading coefficient and not its magnitude.

Unknown channel: In case the receiver does not know the fading coefficient, one has to use block modulation and decoding. Typically one assumes that the fading is constant over blocks of length 2 ($h_t \approx h_{t-1}$). The transmitter transmits a block $[1 \;\; v_{z_t}]$, where the first symbol can be thought of as a training symbol. The receiver uses the phase of the first received symbol as an estimate of the fading phase. But using non overlapping blocks of course cuts the data rate in half. Therefore when the fading varies continuously one lets the blocks overlap. The transmitter sends the symbols

$$s_t = v_{z_t}\, s_{t-1} \qquad t = 1, 2, \dots \quad (s_0 = 1). \tag{4}$$

The ML noncoherent decoder is given by

$$\widehat{z}_t = \arg\, \min_m |x_t - x_{t-1}\, v_m| = \lfloor \arg\,(x_t/x_{t-1})\, M/(2\pi) \rceil\,, \qquad t = 1, 2, \dots. \tag{5}$$

Comparing this with (3) we see that indeed x_{t-1} is used to estimate the phase of the fading. This becomes more clear when we substituting (4) in (1) to obtain (using $h_t = h_{t-1}$):

$$x_t = x_{t-1} v_{z_t} + w_t - v_{z_t} w_{t-1} = x_{t-1} v_{z_t} + \sqrt{2}\, w'_t,$$

where w'_t is iid $\mathcal{CN}(0,1)$. Formally this is equivalent to the known channel model with fading x_{t-1} and twice the noise power. This relates to 3dB loss compared with the coherent decoder.

3 Block Decoding

If the fading coherence interval is sufficiently large, one can recuperate a portion of the noncoherent decoding loss [2]. We assume that the fading is approximately constant over T symbol intervals and will group the sent and received symbols into non overlapping blocks of length T. We will use boldface to denote vectors of length T; the components of a vector $\mathbf{x}$ are given by x_t for $1 \leqslant t \leqslant T$. The channel model can now be written as

$$\mathbf{x} = \sqrt{\rho}\, h\, \mathbf{s} + \mathbf{w}, \tag{6}$$

where $\mathbf{x}, \mathbf{s}, \mathbf{w} \in \mathbf{C}^T$ and $h \in \mathbf{C}$. We also consider the additive group $G = \mathbf{Z}_M^T$: the elements are length T vectors with integer components between 0 and $M-1$ and addition is done component wise and modulo M. The group G clearly has M^T elements. The all one element is $\mathbf{1} \in G$ and the standard unit vectors are $\mathbf{e}_t \in G$ for $1 \leqslant t \leqslant T$. Let $\eta = v_1 = \exp(2\pi i/M)$ and define the exponential map for $\mathbf{g} \in G$ as

$$\eta^{\mathbf{g}} = [\eta^{g_1} \ldots \eta^{g_T}] \in \mathbf{C}^T.$$

The transmitter uses uncoded vectors $\eta^{\mathbf{g}}$ for $\mathbf{g} \in G$. After averaging over the unknown channel phase, the likelihood function at the receiver is a monotone function of

$$\mathcal{L}(\mathbf{g}) = |\mathbf{x}^* \eta^{\mathbf{g}}|,$$

and the ML noncoherent decoder thus is

$$\widehat{\mathbf{g}} = \arg \max_{\mathbf{g} \in G} \mathcal{L}(\mathbf{g}). \tag{7}$$

The ML decoder is unaffected by the phase of $\mathbf{x}^* \eta^{\mathbf{g}}$; integer vectors $\mathbf{g}$ that differ by a multiple of the all one vector $\mathbf{1}$ hence are indistinguishable at the receiver: $\mathcal{L}(\mathbf{g}) = \mathcal{L}(\mathbf{g} + \mathbf{1})$. Our codebook thus consists of equivalence classes of G each with T elements that differ by a multiple of $\mathbf{1}$. As representatives we can take the vectors with first component $g_1 = 0$; hence there are M^{T-1} distinct codewords in the codebook.

A naive implementation of the ML noncoherent decoder computes the likelihood $\mathcal{L}(\mathbf{g})$ for every codeword g; its cost is $O(2^{R(T-1)})$, i.e., exponential in both the rate and the block length. This is a high price to pay in comparison with the standard two symbols differential decoding (5) which is constant per symbol. In this paper we present an algorithm for *exactly* computing the noncoherent ML decoder with complexity $O(T \log T)$. The cost per symbol is thus $O(\log T)$.

4 Fast Block Decoding

First observe from (3) that the coherent decoding decision does not depend on the magnitude of the fading coefficient, but only on its phase. We thus let $h = e^{-i\varphi}$. In case φ is known then the coherent ML decoder as a function of φ is given by

$$\mathbf{\Gamma}(\varphi) = \arg \min_{\mathbf{g}\in G} \left\|\mathbf{x} - e^{-i\varphi}\eta^{\mathbf{g}}\right\| = \arg \max_{\mathbf{g}\in G} \mathrm{Re}\,(\mathbf{x}^* e^{-i\varphi}\eta^{\mathbf{g}}) = \lfloor(\arg \mathbf{x} + \varphi\,\mathbf{1})M/(2\pi)\rceil \mod M.$$

As we pointed out above, coherent ML decoding is done component wise. We next ask ourselves whether there is some value $\widehat{\varphi}$ for φ for which the coherent decoder gives the same answer as the noncoherent decoder (7): $\widehat{\mathbf{g}} = \mathbf{\Gamma}(\widehat{\varphi})$? To answer this note that

$$\mathbf{\Gamma}(\varphi) = \arg \max_{\mathbf{g}\in G} |\mathbf{x}^*\eta^{\mathbf{g}}| \cos(\arg\,(\mathbf{x}^*\eta^{\mathbf{g}}) - \varphi).$$

We know that $\mathbf{g} = \widehat{\mathbf{g}}$ maximizes the first factor in this product. Hence if we choose $\widehat{\varphi} = \arg\,(\mathbf{x}^*\eta^{\widehat{\mathbf{g}}})$ then the cosine is maximal as well at $\mathbf{g} = \widehat{\mathbf{g}}$. Thus

$$\widehat{\mathbf{g}} = \mathbf{\Gamma}(\widehat{\varphi}) \quad \text{with} \quad \widehat{\varphi} = \arg\,(\mathbf{x}^*\eta^{\widehat{\mathbf{g}}}).$$

This of course does not lead to an algorithm as $\widehat{\varphi}$ in turn depends on $\widehat{\mathbf{g}}$. But it shows that a value for φ exists where the coherent decoder agrees with the nocoherent decoder. Hence we can in theory find the answer by scanning all possible values for φ, computing the coherent decoder, and picking the one with the largest noncoherent likelihood:

$$\widehat{\mathbf{g}} = \mathbf{\Gamma}(\widehat{\varphi}) \quad \text{with} \quad \widehat{\varphi} = \arg \max_{\varphi\in[0,2\pi)} \mathcal{L}(\mathbf{\Gamma}(\varphi)).$$

The main observation underlying the algorithm is that when scanning all phases $\varphi \in [0, 2\pi)$, $\mathbf{\Gamma}(\varphi)$ takes on only T distinct codewords. We call them the *test words* $\mathbf{g}^{[t]}$ for $1 \leqslant t \leqslant T$. This can be seen as follows. First observe that

$$\mathbf{\Gamma}(\varphi + 2\pi/M) = \mathbf{\Gamma}(\varphi) + \mathbf{1}, \tag{8}$$

which is a codeword equivalent to $\mathbf{\Gamma}(\varphi)$. Hence we can restrict ourselves to $\varphi \in [0, 2\pi/M)$. Lets start with $\varphi = 0$ and $\mathbf{\Gamma}(0)$. The first test word is $\mathbf{g}^{[1]} = \mathbf{\Gamma}(0)$. Because of (8), each component x_t decodes to $g_t^{[1]}$ when $\varphi = 0$ and to $g_t^{[1]} + 1$ when $\varphi = 2\pi/M$. For each component x_t there is a value for φ in $[0, 2\pi/M)$ where the cross over happens; we call this the *cross over* angle α_t. Note that since the ML decoder is done componentwise and we only scan over $2\pi/M$ this cross over happens only once. The cross over angle for component t can be found as

$$\alpha_t = (g_t^{[1]} + 1/2)2\pi/M - \arg x_t.$$

Figure 1 illustrates this. The closest multiple of $2\pi/M$ to $\arg x_t$ is $g_t^{[1]}$; however when $\arg x_t + \varphi$ is larger than $(g_t^{[1]} + 1/2)2\pi/M$ the closest multiple is $g_t^{[1]} + 1$.

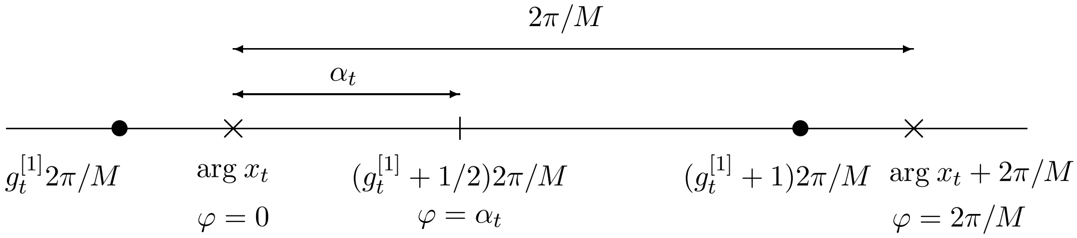

Figure 1: When $\varphi = 0$, the component x_t decodes to $g_t^{[1]}$. However, for $\varphi > \alpha_t$, x_t will decode to $g_t^{[1]} + 1$.

To find the test words, we need to sort the cross over angles at a cost of $O(T \log T)$. Let u_t be the index of the sorted cross over angles so that

$$\alpha_{u_t} \leqslant \alpha_{u_{t+1}} \quad \text{for} \quad 1 \leqslant t < T.$$

We can now build the remaining $T - 1$ test words as follows. Let the angle φ scan from 0 to $2\pi/M$. At $\varphi = 0$ we decode to $\mathbf{g}^{[1]}$. The first cross over angle we encounter is α_{u_1}. Thus the next test vector is obtained by adding one (modulo M) to the u_1th component: $\mathbf{g}^{[2]} = \mathbf{g}^{[1]} + \mathbf{e}_{u_1}$. In general

$$\mathbf{g}^{[t]} = \mathbf{g}^{[t-1]} + \mathbf{e}_{u_{t-1}} \quad \text{for} \quad 2 \leqslant t \leqslant T.$$

For $t = T + 1$ we would end up with $\mathbf{g}^{[1]} + \mathbf{1}$ which is equivalent to $\mathbf{g}^{[1]}$. We thus have T test words. Figure 2 illustrates how the cross over angles divide the interval $[0, 2\pi/M)$ into T pieces each corresponding to one test word.

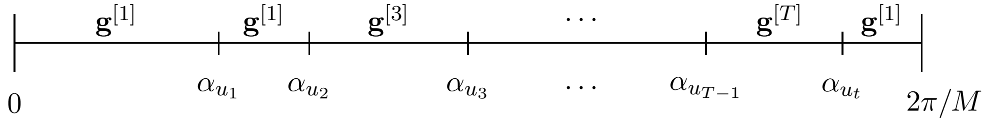

Figure 2: The interval $[0, 2\pi/M)$ is cut up in T pieces by the cross over angles α_{u_t}. If the channel phase is between $\alpha_{u_{t-1}}$ and α_{u_t}, the the coherent ML answer is $\mathbf{g}^{[t]}$.

The ML answer now is

$$\widehat{\mathbf{g}} = \mathbf{g}^{[\hat{t}\,]} \quad \text{with} \quad \widehat{t} = \arg \max_{1 \leqslant t \leqslant T} \mathcal{L}(\mathbf{g}_t).$$

Naively computing $\mathcal{L}(\mathbf{g})$ costs $O(T)$ operations per vector. However, because of their special structure, we can compute the likelihoods of all T test words in $O(T)$. We first compute the inner products $P^{[t]} = \mathbf{x}^* \eta^{\mathbf{g}^{[t]}}$. The first one costs $O(T)$, but each remaining one can be computed recursively at constant cost as

$$P^{[t]} = P^{[t-1]} + x_t^* \, \eta^{g_t^{[1]}} \, (\eta - 1).$$

The likelihoods are now given as $\mathcal{L}(g_t) = |P^{[t]}|$. Because of the sorting, the overall complexity of the algorithm is $O(T \log T)$. The cost per symbol is thus $O(\log T)$.

Because of the equivalence in G, the actual rate is $R(T-1)/T$. The non overlapping blocks all have their first symbol equal to one. By multiplying an entire block with the last symbol of the previous block one can let the blocks overlap by one and the rate becomes R. This can be seen as a block differential scheme.

5 Matlab Program

The above algorithm can be implemented using a 9 line Matlab program.

```
  function g = decode(x,M)

1 eta = exp(2*pi*i/M);
2 arg = angle(x)*M/(2*pi);
3 g = round(arg);
4 [void, u] = sort(g-arg);
5 p = conj(x).*eta.^g;
6 v = [ sum(p) ; p(u)*(eta-1) ];
7 [void,best] = max(abs(cumsum(v)));
8 g(u(1:best-1)) = g(u(1:best-1)) + 1;
9 g = mod(g-g(1),M);
```

Line comments:

2: Compute the phases of the $\mathbf{x}$ vector in multiples of $2\pi/M$.

4: Sort the cross over angles. There is no need to include the constant π/M.

5: Compute the terms of the first inner product $P^{[1]}$.

6: Arrange all the terms of the recursion in a vector.

7: Compute all the inner products, take their absolute values and keep the index of the largest one.

8: Build the best test word $\mathbf{g}^{[\hat{t}]}$.

9: Find the representative of the equivalence class.

6 Comparison with Approximate Algorithm

In [3] Warrier and Madhow introduce an approximate algorithm for computing the noncoherent ML decoder. They take L equally spaced guesses $\varphi_l = 2\pi/(ML)$ ($0 \leqslant l < L$) for the unknown phase and thus have L test words $\mathbf{\Gamma}(\varphi_l)$. Clearly the complexity is $O(TL)$. With the use of the cross over angles defined above we can analyze the probability that the approximate algorithm returns the ML answer.

We make the following simplifying assumption: The cross over angles α_t and the phase φ are independent and distributed uniformly on the interval $[0, 2\pi/M)$. This is only true in the low SNR regime.

Since the interval is cyclic, the T cross over angles divide the interval into T subintervals. We denote a general subinterval with I and its length in units of $2\pi/M$ is the random variable K where $0 \leqslant K \leqslant 1$. The subinterval that contains $\widehat{\varphi}$ gives the ML answer and is denoted I_{ML}. In case one of the L guesses φ_l lies in I_{ML}, the approximate algorithm will return the ML answer. The length of I_{ML} in units of $2\pi/M$ is K_{ML}.

For K to be larger than k, $T-1$ cross over angles need to lie in an interval of length $1-k$. Hence the marginal cumulative probability for K is

$$P(K<k) = 1-(1-k)^{T-1}.$$

To find the cumulative probability of K_{ML} we need to factor in the probability that a particular interval is the ML interval. This is simply the length of the interval. Doing this for all T intervals yields:

$$P(K_{\mathrm{ML}}<k) = T\int_{k'=0}^{k'=k} k' dP(K<k') = 1-(1-k)^{T-1}(Tk-k+1).$$

If $K_{\mathrm{ML}} > 1/L$ then for sure one of the guesses lies in the subinterval I_{ML} and the approximate algorithm returns the ML answer. In case $K_{\mathrm{ML}} < 1/L$, then the chance that one of the L guesses is in I_{ML} is LK_{ML}. Thus the probability that the approximate algorithm is ML is given by

$$\begin{aligned} P_{\mathrm{ML}} &= 1-\int_{k=0}^{k=1/L}(1-Lk)dP(K_{\mathrm{ML}}<k) \\ &= 1-L\int_{0}^{1/L}P(K_{\mathrm{ML}}<k)dk \\ &= \frac{2L}{T+1}-\left(1-\frac{1}{L}\right)^{T}\frac{T+2L-1}{T+1}. \end{aligned}$$

For constant T, increasing L makes P_{ML} go to one. For growing T with constant L, P_{ML} will go to zero. To avoid the probability to go to zero with growing T, one needs to let L grow proportionally to T. For example if $L=T$ the probability P_{ML} converges to $2-3/e \approx .89$. For the approximate algorithm to have a nonzero asymptotic probability to find the ML answer, its complexity thus becomes quadratic ($O(T^2)$).

7 Future Work

There are several possibilities for future work. One can build a running version of this algorithm. The sliding window has length T. The algorithm keeps track of T sorted cross over angles and T inner products. Whenever the window moves by one, a new cross over angle gets inserted in the sorted list and one gets dropped and the inner products need to be updated.

One can build a QAM version. In [3] an algorithm is introduced to use the fast PSK noncoherent decoding an a component in QAM noncoherent decoding. However, the algorithm is still exponential albeit with a smaller base.

Finally one could envision a generalization of this technique to multiple antenna differential modulation, see [4].

Acknowledgments

Special thanks to Michel Mandjes and Mark Hansen for help in the probability analysis of the approximate algorithm and to Jim Mazo for several discussions on differential modulation and for improving the exposition.

References

[1] J. G. Lawton, “Investigation of digial data communication systems,” Tech. Rep. UA-1420-S-1, Cornell Aeronautical Laboratory, Inc., 1961.

[2] D. Divsalar and M. K. Simon, “Maximum-likelihood differential detection of uncoded and trellis codes amplitude phase modulation over awgn and fading channels — metrics and performance,” *IEEE Trans. on Commm*, vol. 42, no. 1, pp. 76–89, 1994.

[3] D. Warrier and U. Madhow, “Noncoherent communication in space and time,” *Submitted to IEEE Inf. Th.*, 1999.

[4] B. Hochwald and W. Sweldens, “Differential unitary space time modulation,” tech. rep., Bell Laboratories, Lucent Technologies, March, 1999. Download available at `http://mars.bell-labs.com`.