Successive interference cancellation in multistream faster-than-Nyquist signaling

Rusek, Fredrik; Anderson, John B

Published in:
Proceedings of the 2006 International Wireless Communications and Mobile Computing Conference

DOI:
10.1145/1143549.1143753

2006

Successive Interference Cancellation in Multistream Faster-than-Nyquist Signaling *

Fredrik Rusek and John B. Anderson
Dept. of Information Technology
Lund University
Box 118, 221 00 Lund, Sweden
{fredrikr, anderson}@it.lth.se

ABSTRACT

In earlier work we have extended Mazo's concept of faster-than-Nyquist signaling to pulse trains that modulate adjacent subcarriers, a method we called two dimensional Mazo signaling. The signal processing is similar to orthogonal frequency division multiplex (OFDM) transmission. Despite pulses that are faster than the Nyquist limit and subcarriers that significantly overlap, the transmission achieves the isolated pulse error performance. In this paper we review the method and test a receiver based on successive interference cancellation. It virtually achieves the matched filter bound.

Categories and Subject Descriptors

E.4 [**Coding and Information Theory**]: Error control codes

General Terms

Communication theory

1. INTRODUCTION AND SYSTEM MODEL

In this paper we present an improved receiver structure for a recently proposed OFDM-like coded modulation. The coded modulation is called multistream faster-than-Nyquist signaling (MFTN). This scheme was presented in [4] and is a generalization of the standard FTN signaling proposed by Mazo [1]. Decoding for the scheme is challenging; a full Maximum likelihood sequence estimation (MLSE) is far too complex. Simplifications exist, however, and this paper focuses on one with near-optimal error performance. Since the signaling method is new, we will devote the first part of the paper to a review of it.

*(This work was supported in part by the Swedish Research Council (VR), grant number 621-2003-3210.

IWCMC'06, July 3–6, 2006, Vancouver, British Columbia, Canada.

Consider baseband signals of the following form

$$s(t) = \sum_{n=1}^{N} a_n h(t - nT), \tag{1}$$

in which a_n are data values over an M-ary alphabet and $h(t)$ is a unit-energy baseband pulse. This simple form underlies QAM, TCM, and the subcarriers in OFDM, as well as many other transmission systems. Most often, $h(t)$ is a T-orthogonal pulse, meaning that the correlation $\int h(t - nT)h^*(t - mT)dt$ is zero, $m \neq n$. Many signals of type (1) can be stacked in frequency through modulation by a set of subcarriers $\{f_k\}$ to form the inphase and quadrature (I/Q) complex baseband signal

$$\begin{aligned} s(t) &= \sum_{k=1}^{K}\sum_{n=1}^{N}[a_{k,n}^I + \jmath a_{k,n}^Q]h(t - nT)e^{\jmath 2\pi f_k t} \\ &= \sum_{k=1}^{K}\sum_{n=1}^{N} a_{k,n}h(t - nT)e^{\jmath 2\pi f_k t} \end{aligned} \tag{2}$$

This is a generalization to a superposition of $2K$ linear modulations, and it carries $2NK$ data values. The $K \times N$ matrix $\mathcal{A} = \{a_{k,n}\}$ is called the data matrix and consists of all the complex data values. The K rows in this matrix correspond to K subcarriers, and the N columns to "time carriers". If $f_k = f_0 + kf_\Delta$, $k = 1, 2, \ldots$, and f_Δ is equal twice the single-sided bandwidth B of $h(t)$, the $2K$ signals are mutually orthogonal. In OFDM signals, both conditions hold at least approximately: $h(t)$ is orthogonal to its own T-shifts and f_Δ is twice the bandwidth of $h(t)$.

The signal design in that case is based on *orthogonality*. According to classical results, there exist about $2W\tau$ orthogonal signals in W positive Hertz and τ seconds. By means of filters matched to each one, data values that modulate the amplitude of each can be maximum-likelihood (ML) detected independently, and therefore about $2W\tau$ symbols can be transmitted. If h in (2) is set to $\sqrt{1/T}\mathrm{sinc}(t/T)$ and $f_\Delta = 1/T$, the product $2W\tau$ is $2(K/T)(NT) = 2KN$; this shows that Eq. (2) carries as many data values as any scheme based on orthogonality can carry.

For a given number of symbols carried by (2), T may be varied, which trades off W against τ. N may also be traded against K. Only the time–bandwidth product matters, and (2) always carries about twice $W\tau$ symbols. In fact there is no need for subcarriers since (1) alone achieves $2W\tau$ by taking $T = 1/2W$, $K = 1$ and $N \approx \tau/T$ symbols.

If the aim is to achieve the *error rate* of a stacked orthogonal-signal system (2), without necessarily using orthogonal sig-

nals, the story is more interesting, and that is our subject in this paper.

We need to be precise about the measurement of error and bandwidth. For (1), the ML-receiver error probability depends on h: As the signal-to-noise ratio E_b/N_0 grows, the probability of incorrect detection of an a_n in additive white Gaussian noise (AWGN) with density $N_0/2$ is asymptotically $P_e \sim Q(\sqrt{d^2_{\min} E_b/N_0})$, where $d_{\min}$ is the minimum distance of the signal set and $d_{\min} \le d_{\mathrm{MF}}$. Here E_b is the bit energy $E_s/\log_2 M$ and d_{MF} is the matched-filter bound. It measures the performance of orthogonal-pulse signaling with the same alphabet: The error rate is $Q(\sqrt{d^2_{\mathrm{MF}} E_b/N_0})$. The paper will concentrate on the binary case, for which $d^2_{\mathrm{MF}} = 2$, so the orthogonal-pulse error rate is $Q(\sqrt{2E_b/N_0})$. If the K signal pairs in (2) do not overlap in frequency, the same asymptotic error rate applies there.

As for signal bandwidth, with uncorrelated data symbols the power spectral density $S_k(f)$ of the kth subcarrier is proportional to $|H(f - kf_\Delta - f_0)|^2 + |H(f + kf_\Delta + f_0)|^2$. The normalized bandwidth (NBW) is measured by

$$\mathrm{NBW} \triangleq \frac{W}{R} \qquad \text{Hz/bit/s}, \tag{3}$$

where W is a measure of the positive frequency bandwidth of the entire transmission (2) (such as 99% power bandwidth) and $R = 2K/T$ is the data rate of each subcarrier in bit/s. For Eq. (1), W is taken as the positive baseband bandwidth and K is 1.

The idea that more throughput can somehow be achieved with (1) at the same error probability was proposed by Mazo [1] in 1975. In a method he called faster-than-Nyquist signaling, binary-modulated sinc(t/T) pulses with bandwidth $1/2T$ Hz appear once each T_Δ seconds, where $T_\Delta < T$; this is faster-than-$1/T$, the Nyquist limit to the rate of orthogonal pulse trains with that bandwidth. The signals are no longer orthogonal and full ML sequence detection is required, which compares all N-symbol signals to the full noisy received signal. By finding the minimum distance $d_{\min}$ of this signal set, one can estimate the asymptotic symbol P_e as $\sim Q(\sqrt{d^2_{\min} E_b/N_0})$. Mazo and later papers showed the surprising result that $d^2_{\min}$ is in fact $d^2_{\mathrm{MF}} = 2$ for $T_\Delta/T > .802$; that is, nothing is lost asymptotically by increasing the symbol rate 24.7% above the Nyquist limit.

The reason for this can be seen by analyzing the error events that can occur. As the pulse rate grows another error event eventually has a distance less than the $d^2 = 2$ antipodal event that leads to d_{MF}. But this does not occur immediately as $1/T_\Delta$ grows. A similar phenomenon occurs with other orthogonal $h(t)$ than the sinc pulse; see [2] for the root RC pulse. Moreover, it often appears with coded modulations, both linear and nonlinear; see [3] (for $h(t)$ a Butterworth filter response) and [6], Chapter 6 (for CPM coded modulation). All these cases can be summed up as follows: the wideband error performance is unchanged under filtering until a surprisingly narrow bandwidth, after which it suddenly drops. This threshold bandwidth is the *Mazo limit*. Its significance is that it is pointless to transmit in a wider bandwidth in a linear channel with AWGN, if sufficient receiver processing is available.

Mazo signaled extra fast in time, but in a subcarrier system one can also space the subcarriers too narrowly in frequency. Now the signal is (2) but the subcarriers cannot be separated by filtering because they overlap in frequency. Still, one can hope that P_e remains $\sim Q(\sqrt{2E_b/N_0})$, as it did in Mazo's signaling. We have introduced this idea in [4]. It was called two-dimensional Mazo signaling because the symbols can be associated with points in a lattice spaced every f_Δ and T_Δ. This is illustrated in Fig. 1. Ref. [4] shows that

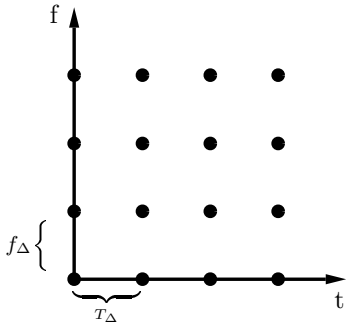

Figure 1: Two dimensional Mazo signaling, in time and frequency. Dots represent symbols separated by f_Δ and T_Δ.

simultaneous frequency and time squeezing indeed increases the symbols transmitted in a given time–bandwidth at the same P_e. Neither compression alone can achieve the same increase.

When f_Δ is less than the subcarrier bandwidth, Eq. (2) is not linear modulation. The signal interrelations that produce $d_{\min}$ work in new ways when both f_Δ and T_Δ can be varied independently. Finding $d_{\min}$ in this new situation is a challenge. Error events encompass both the I and Q signals, and the distance depends on the time the event starts. There are very many events. A branch and bound search among events is needed, and deciding where the search can be pruned is a subtle problem. Nonetheless, $d_{\min}$ can be estimated and it closely predicts the behavior of practical receivers. Some of the details appear in [4].

It is useful to normalize the quantities f_Δ and T_Δ to the orthogonal modulation symbol time T. Define

$$T'_\Delta \triangleq \frac{T_\Delta}{T} \qquad f'_\Delta \triangleq f_\Delta T \tag{4}$$

The orthogonal sinc pulse case has $f_\Delta = 1/T$, $T_\Delta = T$, $f'_\Delta = T'_\Delta = 1$, and the $f_\Delta T_\Delta$ product equals 1. This provides a useful benchmark. The product $f_\Delta T_\Delta = f'_\Delta T'_\Delta$ Hz-s/bit for a scheme is a measure of bandwidth consumption relative to the sinc benchmark, in the limit of N and K. Distance studies with various pulses $h(t)$ show that $d^2_{\min} = 2$ can occur at less than 0.5 Hz-s/bit, i.e., less than 50% of the sinc benchmark. In what follows $h(t)$ is a 30% excess bandwidth root RC pulse. Note that for this, $f'_\Delta = 1.3$ is required for independence of the subcarriers, and a consumption close to 0.5 is even more dramatic.

2. DECODING

Decoding of this type of coded modulation is complex. Full sequence estimation grows exponentionally with the number of subcarriers K, so MLSE decoding is ruled out. We have two desires for the decoder. The first is that it should perform close to the MLSE error performance; the second is that it should be relatively simple. Such a decoding algorithm is the M-algorithm [6]. This was tested in [4], but it only worked well for 2–4 subcarriers, since otherwise it was not clear in what order the symbols should be decoded.

Decoding and equalization of two dimensional ISI signals is a well known problem, especially in the magnetics literature, [7]– [9]. But the ISI pattern is almost always much simpler than here; it is usually limited to only three interfering taps and binary data is used. In some work the ISI pattern is assumed to be separable, which is not the case here. The standard approach in the literature seems to be some form of interference cancellation, as here. But perhaps the most significant difference is that we intentionally added the ISI to save bandwidth; in order to exploit that saving we must be able to have performance near MLSE. This is generally not needed for unintentional ISI where the objective is to get reasonable performance at low complexity.

In [5] the receiver structure was based on successive interference cancellation. This receiver works quite well for time–bandwidth products above $f'_\Delta T'_\Delta \geq .7$, especially if f'_Δ is large, i.e. close to 1. But for smaller products and products where f'_Δ is small, that method does not work at all. The reason is that the decoder is based on a BCJR algorithm in the time direction, and it only considers a single subcarrier. Here we will extend the idea to BCJR algorithms in the frequency direction and also use multidimensional algorithms. But such decoder alone is not capable of resolving all the intersymbol and interchannel interference; therefore we will cascade a second stage decoder after the first. The errors in the output from the first stage appear in bursts, so called error events. By using the second stage decoder we dramatically improve performance.

We briefly describe the overall plan for the total decoder. As mentioned, the decoder consists of two parts. The task of the first part is to produce an estimation of $\mathcal{A}$, which is allowed to contain many isolated error events, and which is fed to the second stage decoder. The internal structure of this first decoder will be based on iterative soft interference cancellation. In this paper the first decoder will pass a hard output to the second decoder. It is well established in the literature that stagewise decoding works better if the message passing is soft, but our hard message approach works surprisingly well. The second decoder will now try to improve the output of the first decoder. This is accomplished by additional decodings and by comparing these new estimations with the output of the first decoder. Its success relies on the fact that a small region of errors can easily be removed.

We start by describing the first stage decoder. It encounters a noisy complex baseband signal $r(t)$ where $r(t) = s(t) + n(t)$, and $n(t)$ is Gaussian noise. The signal $s(t)$ is the data carrying signal formed by (2). The first step in the receiver is to project $r(t)$ onto the basis functions $h(t - nT_\Delta)e^{\jmath 2\pi t f_k}$, i.e. to compute

$$R_{k,n} = \int_{-\infty}^{\infty} r(t)h^*(t - nT_\Delta)e^{-\jmath 2\pi t f_k}dt. \tag{5}$$

The matrix $\mathbf{R}$ represents the received signal. Since

$$s(t) = \sum_n \sum_k a_{k,n}h(t - nT_\Delta)e^{\jmath 2\pi t f_k}, \tag{6}$$

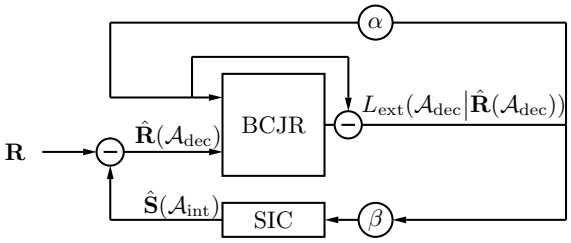

Figure 2: System model for the first stage decoder. The block SIC produces a signal based on estimates of all symbols in $\mathcal{A}_{\text{int}}$. The block BCJR is a BCJR algorithm for the symbols in $\mathcal{A}_{\text{dec}}$.

we have that the signal part $\mathbf{S}$ of $\mathbf{R}$ equals

$$\begin{aligned} S_{k,n} &= \int_{-\infty}^{\infty} \sum_{m,l} a_{l,m}h(t - mT_\Delta)e^{\jmath 2\pi t(l-k)f_\Delta}h^*(t - nT_\Delta)dt \\ &= \int_{-\infty}^{\infty} \sum_{m,l} a_{l,m}\lambda'[m,n,l,k]dt \\ &= \int_{-\infty}^{\infty} \sum_{m,l} a_{l,m}\lambda[m,n,l-k]dt. \end{aligned} \tag{7}$$

and the noise part $\mathbf{N}$ is given by

$$N_{k,n} = \int_{-\infty}^{\infty} n(t)h^*(t - nT_\Delta)e^{-\jmath 2\pi t f_k}dt. \tag{8}$$

Then $\mathbf{R}$ can be written as $\mathbf{R} = \mathbf{S} + \mathbf{N}$. The last equality of (7) holds since the system is time variant (that is $\lambda'[m,n,l,k]$ depends on both m and n and not only on $m - n$), but frequency invariant.

The first decoder will be an iterative one. Although its task is to output hard estimates to the second decoder, it will internally work with soft values and try to maximize the *a posteriori* probability (APP) of an individual bit, i.e.

$$\hat{a}_{k,n}^{I/Q} = \arg \max_{a \in \{-1,1\}} Pr(\hat{a}_{k,n}^{I/Q} = a \,|\, \mathbf{R}). \tag{9}$$

Here and throughout, superscript I/Q means "I respectively Q". When this superscript is omitted we intend a complex a. Instead of working with probabilities it is convenient to work with log-likelihood ratios (LLRs)

$$L(a_{k,n}^{I/Q}) = \log \frac{\Pr\{a_{k,n}^{I/Q} = 1\}}{\Pr\{a_{k,n}^{I/Q} = -1\}} \tag{10}$$

Since the data symbols are independent we can as usual express the conditional LLR $L(a_{k,n}^{I/Q}|\mathbf{R})$ as

$$L(a_{k,n}^{I/Q} \,|\, \mathbf{R}) = L_{\text{ext}}(a_{k,n}^{I/Q} \,|\, \mathbf{R}) + L(a_{k,n}^{I/Q}) \tag{11}$$

where $L_{\text{ext}}(a_{k,n}^{I/Q}|\mathbf{R})$ denotes the extrinsic information about $a_{k,n}^{I/Q}$ contained in $\mathbf{R}$.

The true APPs of the data bits can be found by a multidimensional BCJR algorithm, but as with MLSE, the complexity grows exponentially with K, and the APPs have to be approximated by simpler means. One way is by an iterative method, and we use successive interference cancellation. A system model is shown in figure 2.

The symbols $\{a_{k,n}\}$ are grouped into two sets: $\mathcal{A}_{\rm dec}$ and $\mathcal{A}_{\rm int}$; the symbols in set $\mathcal{A}_{\rm dec}$ are symbols that we try to decode at the moment and the symbols in set $\mathcal{A}_{\rm int}$ are treated as interference. The transmitted signal $s(t)$ can be expressed as $s(t) = s_{\rm dec}(t) + s_{\rm int}(t)$, where $s_{\rm dec}(t)$ and $s_{\rm int}(t)$ are the contributions from symbols in $\mathcal{A}_{\rm dec}$ and $\mathcal{A}_{\rm int}$, respectively. In each iteration of the decoding process a soft estimate $\hat{s}_{\rm int}(t)$ of $s_{\rm int}(t)$ is formed based on soft information about all symbols in $\mathcal{A}_{\rm int}$:

$$\hat{s}_{\rm int}(t) = \sum_{\mathcal{B}} b_{k,n}^{I/Q} h(t - nT_\Delta) e^{j2\pi t f_k}, \tag{12}$$

where

$$\mathcal{B} = \{(k,n) : a_{k,n} \in \mathcal{A}_{\rm int}\} \tag{13}$$

and $b_{k,n}^{I/Q}$ are the soft estimates of $a_{k,n}^{I/Q}$, defined by

$$\begin{aligned} b_{k,n}^{I/Q} &= P\{a_{k,n}^{I/Q} = 1\} - P\{a_{k,n}^{I/Q} = -1\} \\ &= \tanh(\beta L_{\rm ext}(a_{k,n}^{I/Q} \,|\, \mathbf{R})/2) \end{aligned}$$

Then $\hat{s}_{\rm int}(t)$ is projected onto the basis functions, and the projection is denoted $\hat{\mathbf{S}}(\mathcal{A}_{\rm int})$. Finally, the tentative received signal when decoding symbols in $\mathcal{A}_{\rm dec}$ is formed as

$$\hat{\mathbf{R}}(\mathcal{A}_{\rm dec}) = \mathbf{R} - \hat{\mathbf{S}}(\mathcal{A}_{\rm int}). \tag{14}$$

Together with the extrinsic information about the symbols $a_{k,n}^{I/Q} \in \mathcal{A}_{\rm dec}$, $L_{\rm ext}(a_{k,n}^{I/Q}|\hat{\mathbf{R}}(\mathcal{A}_{\rm dec}))$, the signal $\hat{\mathbf{R}}(\mathcal{A}_{\rm dec})$ is fed to a further decoding algorithm. Observe that in each iteration, many "subiterations" have to be done in order that all symbols appear in some $\mathcal{A}_{\rm dec}$.

In [5] the particular choice $\mathcal{A}_{\rm dec} = \{a_{K_i,n}, \forall n\}$ was used, where K_i is a fixed value for every subiteration; this corresponds to row i in the data matrix $\mathcal{A}$ and will be referred to as rowwise decoding. If f'_Δ is small, the interference from $\{a_{K_i-1,n}, \forall n\}$ and $\{a_{K_i+1,n}, \forall n\}$ grows large which limits the performance. To avoid this we will here consider different partitions of $\mathcal{A}$; essentially, we will consider the partition

$$\mathcal{A}_{\rm dec} = \{a_{k,N_i}, \ldots, a_{k,N_{i+L-1}}, \forall k\} \tag{15}$$

for subiteration i. This corresponds to L adjacent columns in the data matrix and is refered to as columnwise decoding.

The decoding algorithm that will be used to decode the signal $\hat{\mathbf{R}}(\mathcal{A}_{\rm dec})$ in (14) is non trivial. First, we do not assume the Forney model for the ISI, but instead consider the Ungerboeck model. This results in colored noise and a non causal ISI response, which prohibits the standard BCJR algorithm. However, in [10] a BCJR type algorithm was derived for the Ungerboeck model, and this algorithm will be modified to a multi-dimensional form.

The probability of receiving $\hat{\mathbf{R}}(\mathcal{A}_{\rm dec})$ given $\{a_{k,n}\}$ is sent becomes (apart from a constant of proportionality)

$$\Pr(\hat{\mathbf{R}}(\mathcal{A}_{\rm dec})|\{a_{k,n}\}) \sim \exp\{\frac{1}{4N_0} J(\{a_{k,n}\})\}, \tag{16}$$

where

$$\begin{aligned} J(\{a_{k,n}\}) &= 2{\rm Re}\left(\sum_{k,n} a_{k,n}^* \hat{\mathbf{R}}(\mathcal{A}_{\rm dec})_{k,n}\right) \\ &- \sum_{k,l}\sum_{m,n} a_{k,n}^* a_{m,l} \lambda[m,n,l-k] \end{aligned} \tag{17}$$

It is possible to compute $J(\{a_{k,n}\})$ in a recursive manner. If we consider the partition given in (15), and let $\bar{a}_k = \{a_{k,1}, \ldots, a_{k,L}\}$ and column $N_i = 1$ we get

$$\begin{aligned} J(\{\bar{a}_1 \ldots \bar{a}_s\}) &= J(\{\bar{a}_1 \ldots \bar{a}_{s-1}\}) \\ &+ 2{\rm Re}\left(\sum_{j=1}^{L} a_{s,j}^* \hat{\mathbf{R}}(\mathcal{A}_{\rm dec})_{s,j}\right) \\ &- 2{\rm Re}\left(\sum_{t=1}^{L} a_{s,t}^* \sum_{r<s}\sum_{p=1}^{L} a_{r,p}\lambda[t,p,r-s]\right) \\ &- {\rm Re}\left(\sum_{t=1}^{L} a_{s,t}^* \sum_{p=1}^{L} a_{s,p}\lambda[t,p,r-s]\right) \end{aligned} \tag{18}$$

By observing that $\lambda[x,x,y] = 0$ if $|y| > 1$, the third term of (18) simplifies into

$$-2{\rm Re}\left(\sum_{t=1}^{L} a_{s,t}^* \sum_{p=1}^{L} a_{s-1,p}\lambda[t,p,r-s]\right) \tag{19}$$

Based on this metric computation, the algorithm in [10] can be used as decoding method. Note that the extrinsic information fed to the SIC and to the decoder is first attenuated by coefficients α and β. The values for α and β were determined by simulation. We have used slowly increasing values over the iterations for both α and β. A particular choice of the coefficients is given in section 3.

The decoder requires the noise variance as input. Since there is noise both from the AWGN process as well as from symbols belonging to the set $\mathcal{A}_{\rm int}$ this variance has to be estimated for every iteration [11].

Finally, as indicated in figure 2 the output from the decoding block is $L_{\rm ext}(\mathcal{A}_{\rm dec}|\hat{\mathbf{R}}(\mathcal{A}_{\rm dec}))$, which hopefully equals $L_{\rm ext}(\mathcal{A}_{\rm dec}|\mathbf{R})$ after the final iteration. Let $\hat{\mathcal{A}}_1$ denote the hard output from this first stage decoder.

As already mentioned, the output of first stage decoder is not good enough, and the BER lies in general far away from the MLSE performance. We will now describe the second stage decoder. This decoder was first tested in [12], we extend it somewhat here. The main idea goes as follows. Assume that the difference between $\mathcal{A}$ and $\hat{\mathcal{A}}_1$ is limited to two carriers only, say carrier k and $k+1$. The other carriers are error-free, and their interference can be removed from k and $k+1$. It has been demonstrated [4] that decoding of two carriers alone can be done via the M-algorithm with virtually MLSE performance. Therefore we can assume that the error event will be eliminated when the signal is decoded. Denote the output of this decoding as $\hat{\mathcal{A}}^k$. In fact, in an actual decoding the above procedure must be repeated for all pairs $\{k, k+1\}$, $k = 1, \ldots, K-1$, since we do not know where the error event is located in advance. This gives us $K-1$ additional data matrices to choose from.

A final output from the second stage decoder could be selected from the candidate set $\mathcal{C} \triangleq \{\hat{\mathcal{A}}_1, \hat{\mathcal{A}}^1, \ldots, \hat{\mathcal{A}}^{K-1}\}$, but then only error events in two carriers could be corrected and the above procedure has to be repeated many times if there are many error events in $\hat{\mathcal{A}}_1$. Instead, we try to combine the data matrices in the best possible way when constructing the final output. First, since symbols separated by many symbol intervals have very small crosscorrelation, it follows that these can be independently corrected; in fact, since there is only overlap between adjacent carriers it is enough if the

error events are separated by only a single carrier in order to make the error events perfectly independent.

Now, find time and frequency indexes in $\mathcal{C}$ such that for these indexes there is more than one alternative for that symbol in $\mathcal{C}$, i.e. find positions where two or more of the $\hat{\mathcal{A}}^l$ have different symbol values. Then, as described above, form different regions of differences that can be independently considered. In our example error event above, this region will definitely involve carrier k and $k+1$ and possibly also $k-2$, $k-1$, $k+2$ and $k+3$ depending on the outcome of the decodings. On each of these carriers we have maximum three different suggestions (on carrier k, we can have different opinions in $\hat{\mathcal{A}}^{k-1}$, $\hat{\mathcal{A}}^k$ and $\hat{\mathcal{A}}_1$) for the data symbols; on each carrier each of the suggestions is considered as a supersymbol. One supersymbol consists of several symbols. As output in this region we take the combination of supersymbols, i.e. rows of symbols, that minimizes the ML metric

$$\int_{-\infty}^{\infty} |r(t) - s(t)|^2 dt \tag{20}$$

Since only adjacent carriers overlap, the memory in the frequency direction equals one; finding the best combination can therefore be done with a Viterbi algorithm with three states. Of course, when minimizing (20) all $K \times N$ symbols are needed to form a tentative signal $s(t)$, but since only symbols close to the ambiguity area in $\mathcal{C}$ will affect the result, and all these symbols are constant (there is only one opinion for them in $\mathcal{C}$), the calculation does not have to involve all symbols but ony a few around the ambiguity area in $\mathcal{C}$.

The procedure summarized above works very well when the errors from the first decoder are limited to only two subcarriers, as in [12]. But in cases where error events affect many carriers but only a few symbol intervals, it is better to use columnwise decoding. This means that in what is described above, k and $k+1$ denotes columns instead of subcarriers. In fact, when doing columnwise decoding we will consider up to four columns at once. In our receiver tests presented in the next section we try several types of rowwise and columnwise decoding.

3. NUMERICAL RESULTS

Here we present some receiver tests. Throughout the tests we have used $K = 20$ subcarriers. Different receiver solutions are needed in the different parts of the (f_Δ, T_Δ) plane. In figure 3 the so called Mazo limit for 30% root raised cosine pulses is shown. The circles indicate the systems that will be tested; note that one setup violates the Mazo limit.

We start with the two simulations for the case $f'_\Delta = .7$. By inspection it was seen that the interference from $\mathcal{A}_{\text{int}}$ is large if $\mathcal{A}_{\text{dec}}$ is taken as one row in the data matrix. However, it is even larger if we take $\mathcal{A}_{\text{dec}}$ as a few columns. Therefore, we alternate between rowwise and columnwise decoding in the first decoder; every third decoding is a columnwise one. In the columnwise decodings we have used $L = 3$ columns, i.e.

$$\mathcal{A}_{\text{dec}} = \{a_{k,N_1}, a_{k,N_1+1}, a_{k,N_1+2}, \forall k\} \tag{21}$$

Since every symbol is a QPSK symbol and $\lambda[x, x, y] = 0$ if $|y| > 1$, the number of states is $4^3 = 64$. Since every state can reach every other state in the trellis the branching factor of the trellis also becomes 64. Moreover, in the different

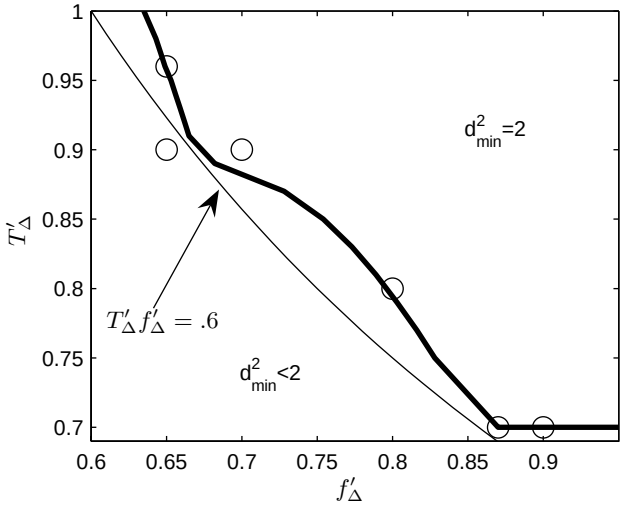

Figure 3: The Mazo limit for 30% root raised cosine pulses. Below the bold line $d^2_{\min}$ falls below 2. The thin line shows a constant product of $T'_\Delta f'_\Delta = .6$. The circles show points that have been decoded.

iterations the value N_1 is selected differently. Let i denote the i-th columnwise iteration and p the p-th subiteration within this i-th columnwise iteration; then N_1 in (21) is selected as

$$N_1 = (p-1)L + (i \mod L) + 1 \tag{22}$$

The feedback values α and β are chosen as $\alpha = \min(1, 0.1i)$ and $\beta = \min(1, 0.15i)$; we believe that much better tunings are possible, but it is time consuming to find them. In total we performed 9 iterations in the first decoder. For the second stage decoder we also lternate between rowwise and columnwise decodings. For the rowwise decoding we used an M-algorithm that considers two rows at once [4]. We performed two columnwise decodings, one with $L = 3$ and one with $L = 4$. When $L = 4$ the number of states becomes 256, but we used the M-algorithm with $M = 64$ to limit the state complexity of all used decoders.

The receiver tests are shown in figure 4. The receiver performance is very close to the reference curve, $Q(\sqrt{2E_b/N_0})$, which is a rather tight lower bound to the MLSE performance for high E_b/N_0. The dashed line is a receiver test for $(T'_\Delta, f'_\Delta) = (.7, .9)$ taken from [12]; the decoder used was the same as we used, but only rowwise decoders were applied. The error floor is then present since there is a probability that error events from the first decoder are such that rowwise decoders alone cannot eliminate them.

We strongly suspect that our decoder also suffers from an error floor, but at a much lower BER. The reason is that if error events from the first decoder involve too many subcarriers and span many symbol intervals, the second decoder will fail.

The next receiver test is for the case $f'_\Delta = T'_\Delta = .8$. Now we alternate between rowwise and columnwise decodings both in the first and in the second decoder. In the first decoder we set $L = 2$ and in the second we start by $L = 3$ but increase to $L = 4$ (and use the M-algorithm) in the final

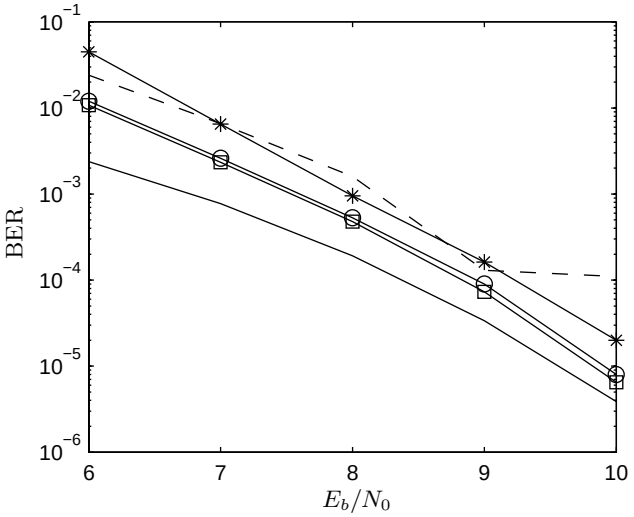

Figure 4: Receiver tests; circles mark $(T'_\Delta, f'_\Delta) = (.7, .9)$, asterixes mark $(.7, .87)$ and squares mark $(.8, .8)$. The unmarked solid line is the basic reference $Q(\sqrt{2E_b/N_0})$. The dashed curve is a receiver test for $(T'_\Delta, f'_\Delta) = (.7, .9)$ taken from [12].

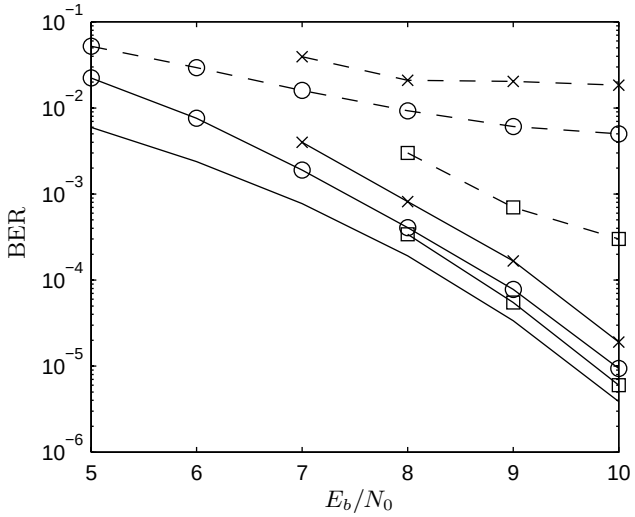

Figure 5: Receiver tests; solid lines are outputs from the second decoder, dashed lines are outputs from the first decoder. Squares mark $(T'_\Delta, f'_\Delta) = (.96, .65)$, circles mark $(.9, .7)$ and asterixes mark $(.9, .65)$. The unmarked solid line is the basic reference $Q(\sqrt{2E_b/N_0})$.

iteration. For the rowwise decoding in the second decoder we again used an M-algorithm applied to two rows at once and $M = 64$. The test is shown in figure 4.

The final tests are the three cases where $T'_\Delta \geq .9$. Unlike $T'_\Delta = .7$, now every third decoding in the first decoder is a rowwise decoding. In the second decoder three columnwise decodings with $L = 2$, $L = 3$ and $L = 4$ are performed, and this is followed by one rowwise decoding. The results are given in figure 5. It is seen that the second decoder has a dramatic effect on the BER; in one case there is a 1000-fold reduction. The improvement is less dramatic for the receiver tests in figure 4.

This decoding method is without doubt complex; we have in total performed between 10 and 15 iterations in the tests, and in each iteration decoders with up to 64 states are used. Further research could look into reduced complexity decoding algorithms such as the T–BCJR [13] or the recently proposed M*–BCJR [14]. Especially the latter seems promising.

4. CONCLUSIONS

A two-stage decoder based on soft intererence cancellation has been proposed for a multicarrier faster-than-Nyquist coded modulation scheme. The decoder has a performance close to the theoretical limit, but complexity is rather high. Several systems around the Mazo limit have been tested. The decoder can in general be used for an arbitrary multicarrier modulation scheme suffering from both intersymbol and interchannel interference.

5. REFERENCES

[1] J.E. Mazo, "Faster-than-Nyquist Signaling," *Bell Syst. Tech. J.*, vol. 54, pp. 1451–1462, Oct. 1975.

[2] A.D. Liveris and C.N. Georghiades, "Exploiting Faster-Than-Nyquist Signaling," *IEEE Trans. Commun.*, vol. 51, pp. 1502–1511, Sep. 2003.

[3] F. Rusek and J.B. Anderson, "M-ary Coded Modulation by Butterworth Filtering," *Proc.*, 2003 Int. Symp. Information Theory, Yokohama, p. 184, June 2003.

[4] F. Rusek and J.B. Anderson, "The Two Dimensional Mazo Limit," *Proc.*, 2005 Int. Symp. Information Theory, Adelaide, pp. 970–974, Sept. 2005.

[5] F. Rusek and J.B. Anderson, "Improving OFDM: Multistream Faster than Nyquist Signaling," *Proc.* 6th Int. ITG Conf. Source and Channel Coding, Munich, Apr. 2006.

[6] J.B. Anderson and A. Svensson, *Coded Modulation Systems*, Plenum/Kluwer, New York, 2003.

[7] M. Marrow and J.K. Wolf, "Iterative Detection of 2-Dimensional ISI Channels," in *Proc.* 2003 Int. Workshop Information Theory, Paris, pp. 131-134, April 2003.

[8] J.B. Soriaga, H.D. Pfister and P.H. Siegel, "On Achievable Rates of Multistage Decoding on Two-Dimensional ISI Channels," *Proc.*, 2005 Int. Symp. Information Theory, Adelaide, pp. 1348–1352, Sept. 2005.

[9] J.A. O'Sullivan, et. al., "Iterative Detection and Decoding for Separable Two-Dimensional Intersymbol Interference," *IEEE Trans. on Magnetics*, vol. 39, pp. 2115-2120, July 2003.

[10] G. Colavolpe and A. Barbieri, "On MAP Symbol Detection for ISI Channels Using the Ungerboeck Observation Model," *IEEE Commun. Letters*, vol. 9, pp. 720-722, Aug. 2005.

[11] M. Kobayashi, J. Boutros and G. Caire, "Successive Interference Cancellation with SISO Decoding and EM Channel Estimation," *IEEE Journal on Selected Areas in Comm.*, vol. 19, pp. 1450-1460, Aug. 2001.

[12] I. Casasnovas and B. Pachan, "Decoding for Compressed Multicarrier Modulation," Master thesis, April 2006, Information Tech. Dept., Lund University, Sweden.

[13] V. Franz and J.B. Anderson, "Concatenated Decoding with Reduced-Search BCJR Algorithm," *IEEE Journal on Selected Areas in Comm.*, vol. 16, no. 2, pp. 186-195, Feb. 1998.

[14] M. Sikora and D.J. Costello, "A new SISO Algorithm with Application to Turbo Equalization," *Proc.*, 2005 Int. Symp. Information Theory, Adelaide, pp. 2031–2035, Sept. 2005.