Linear time and frequency domain Turbo equalization

Michael Tüchler, Joachim Hagenauer
Lehrstuhl für Nachrichtentechnik
TU München
80290 München, Germany
{micha,hag}@lnt.ei.tum.de

Abstract

For coded data transmission over channels introducing inter-symbol interference, one approach for joint equalization and decoding in the receiver is Turbo Equalization. We rederive existing linear equalization algorithms applicable to Turbo Equalization for 2^m-ary signal alphabets and compare their computational complexity. Moreover, by evaluating the algorithm performance properly, we select for each iteration the most suitable of the two algorithms with lowest computational complexity and achieve at low bit error rates a performance close to that of optimal approaches for equalization, i.e., maximum a-posteriori probability symbol detection.

1. Introduction

We consider a coded data transmission system, where blocks of data bits are encoded to code bits using forward error correction (FEC), which are subsequently interleaved, mapped to symbols from a 2^m-ary signal alphabet and transmitted over a channel with inter-symbol interference (ISI). The channel is modeled in discrete time with the finite-length impulse response filter $h[n] = \sum_{i=0}^{M-1} h_i\, \delta[n-i]$, $h_i \in \mathbb{C}$, of length M. The impulse response has energy $E_h = \sum_{i=0}^{M-1} |h_i|^2$. The coefficients h_i are assumed to be time-invariant and known to the receiver. The noise process is assumed to be independent and identically distributed (i.i.d.) and independent of the data. This system model is valid for many communication systems with frequency selective or multipath channels.

The receiver of such a system can perform joint decoding and equalization using Turbo Equalization (Turbo Equ.), which was pioneered in [4] and enhanced in [1, 2]. However, the used trellis-based detection algorithms (soft-out Viterbi equalization (SOVE), maximum a-posteriori probability (MAP) symbol detection) become prohibitively complex for increasing M and m. In [5, 6, 9, 12], new equalization techniques based on linear filtering were applied to significantly reduce the computational complexity. Among them, we differentiate between minimum mean squared error (MMSE) linear equalization (LE) and matched filtering (MF). The LE algorithm derived in [9] was also implemented in an approximate version (APPLE).

In this paper, we provide a framework to use the linear approaches given a 2^m-ary signal alphabet (LE: Section 4.1, APPLE: Section 4.2, MF: Section 5) and specify how to select the most suitable equalization algorithm for each iteration - an approach, which significantly improves the performance as shown in [9]. In all systems, a convolutional code with MAP-based decoding is used for FEC. We start with a brief system definition, explain next the general approach to derive linear algorithms applicable for Turbo Equ., derive in detail the different algorithms (LE, APPLE, and MF) in the time and, if possible, in the frequency domain, devise an adaptation criterion to switch between the algorithms, compare the computational complexity, and conclude the paper with results and final remarks.

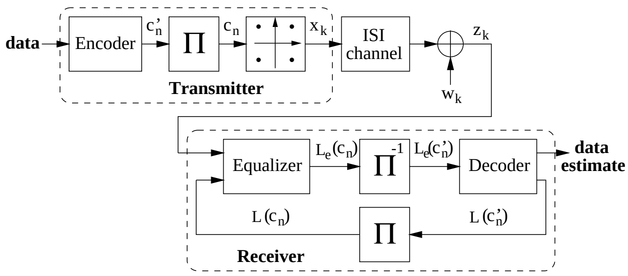

Figure 1. Coded data transmission system.

2. System definition

Consider the communication system in Figure 1 with a receiver performing Turbo Equ. Binary data is encoded using a binary convolutional code to length $L \cdot m$ blocks $\mathbf{c}' \triangleq [c'_0\, c'_1 \cdots c'_{L\cdot m-1}]^T$ of code symbols $c'_n \in \{0, 1\}$. The interleaver permutes $\mathbf{c}'$ to $\mathbf{c} \triangleq [c_0\, c_1 \cdots c_{L\cdot m-1}]^T$ denoted as $\mathbf{c} = \Pi(\mathbf{c}')$. The deinterleaver $\Pi^{-1}(\cdot)$ reverses the permutation $\Pi(\cdot)$. The modulator maps m code bits c_{mk+j}, $j = 0, ..., (m-1)$, to a complex symbol x_k according to the 2^m-

ary symbol alphabet $S=\{s_0, s_1, ..., s_{2^m-1}\}$, where s_i corresponds to the bit pattern $[b_{i,0}\, b_{i,1} ... b_{i,(m-1)}]$, $b_{i,j}\in\{0,1\}$. We require that $\sum_{i=0}^{2^m-1} s_i=0$ and $\frac{1}{2^m}\sum_{i=0}^{2^m-1}|s_i|^2=1$.

Transmitted over the channel is the sequence $\mathbf{x} = [x_0\, x_1 \cdots x_{L-1}]$ after the length $M-1$ prefix or guard interval $[x_{L-M}\, x_{L-M+1} \cdots x_{L-1}]$, where we assume that the transmitter knows M. In many applications, the prescribed prefix is already part of $\mathbf{x}$ due to fixed header and tail sequences. The receiving process of the transmitted x_k is disturbed by complex-valued additive white Gaussian noise (AWGN), i.e., both the real and imaginary part of the noise samples w_k is i.i.d. with pdf $n_{0,\frac{1}{2}\sigma_w^2}(w)$ defined as

$$n_{\mu,\sigma^2}(w) \triangleq \frac{\exp(-\frac{1}{2\sigma^2}(w-\mu)^2)}{\sqrt{2\,\pi\,\sigma^2}}, \quad w,\mu\in\mathbb{R},\ \sigma^2\in\mathbb{R}^+.$$

Thus, we have $E\{\Re\{w_k\}^2\} = E\{\Im\{w_k\}^2\} = \frac{1}{2}\sigma_w^2$ and $E\{|w_k|^2\} = \sigma_w^2$. The receiver observes the sequence $\mathbf{z} = [z_0\, z_1 \cdots z_{L-1}]$ (the first $M-1$ symbols are neglected). Due to the prefix, the channel state at the block ends is equal and we can express z_k as

$$z_k \triangleq \left(\sum_{i=0}^{M-1} h_i\, x_{(k-i) \bmod L}\right) + w_k, \quad k=0,...,(L-1).$$

In case all $s_i, h_i \in S$ are real, we can design a receiver using $\Re\{z_k\}$ only, which yields $E\{\Re\{w_k\}^2\} = \sigma_w^2$ and $E\{\Im\{w_k\}^2\}=0$.

Before proceeding, some frequently used notation is introduced. The $i\times j$ matrix $\mathbf{0}_{i\times j}$ contains all zeros, $\mathbf{1}_{i\times j}$ contains all ones. $\mathbf{I}_i$ is the $i\times i$ identity matrix. The operator $E\{\cdot\}$ is the expectation with respect to the joint probability density function (pdf) of the x_k and w_k. The covariance operator $\mathrm{Cov}(\mathbf{x},\mathbf{y})$ equals $E\{\mathbf{x}\,\mathbf{y}^H\}-E\{\mathbf{x}\}E\{\mathbf{y}^H\}$, where ^{H} is the Hermitian operator. The L-value operator $L(c)$, $c\in\{0,1\}$, equals $L(c)\triangleq \ln\frac{\Pr\{c=0\}}{\Pr\{c=1\}}$, i.e., $L(c)$ is the log likelihood ratio (LLR). The operator $\mathbf{Diag}[\cdot]$ applied to a length N vector returns a $N\times N$ matrix with the vector elements along the diagonal.

3. Linear algorithms for Turbo equalization

We present here the general framework to rederive LE, APPLE, and MF for a 2^m-ary signal constellation using the results in [9]. At first, the statistics $\bar{x}_k \triangleq E\{x_k\}$ and $v_k \triangleq \mathrm{Cov}(x_k, x_k)$ of the symbols x_k are computed using the a-priori information $L(c_n)$ provided by the decoder:

$$\bar{x}_k = \sum_{s_i\in S} s_i\cdot Pr\{x_k=s_i\} = \sum_{s_i\in S}\prod_{j=0}^{m-1} s_i\cdot Pr\{c_{mk+j}=b_{i,j}\},$$
$$v_k = \left(\sum_{s_i\in S}|s_i|^2\cdot Pr\{x_k=s_i\}\right) - |\bar{x}_k|^2. \qquad (1)$$

The equalizer assumes the c_n to be independent (which is locally achieved using interleaving) such that $Pr\{x_k=s_i\}$ is the product of m terms $Pr\{c_{mk+j}=b_{i,j}\}$, which are determined using $L(c_n)$, $n=mk+j$. From the independence assumption follows $\mathrm{Cov}(x_k, x_{k'})=0$, $\forall k'\neq k$, too. Filtering $\bar{x}_k$ with $h[n]$ gives

$$\bar{z}_k \triangleq E\{z_k\} = \sum_{i=0}^{M-1} h_i\, \bar{x}_{(k-i) \bmod L}, \quad k=0,...,(L-1).$$

which is subtracted from the received symbols z_k. This difference is filtered using a length N linear FIR filter with possibly time-varying coefficients $f_{i,k}$, $i=-N_1, 1-N_1, \cdots, N_2$, $(N=N_1+N_2+1)$. The output of this filter are the estimates $\hat{x}_k$.

The equalizer output LLRs $L_e(c_n)$, $n=0,...,(L\cdot m-1)$, are the "extrinsic" information (a-posteriori minus a-priori information) about c_n given the channel observations:

$$\begin{aligned} L_e(c_n) &\triangleq L_{apost}(c_n) - L(c_n) \\ &\triangleq \ln\frac{\Pr\{c_n=0|\hat{x}_k\}}{\Pr\{c_n=1|\hat{x}_k\}} - L(c_n) = \ln\frac{p(\hat{x}_k|c_n=0)}{p(\hat{x}_k|c_n=1)}. \end{aligned}$$

It is shown in [8] that this decomposition of the a-posteriori LLR $L_{apost}(c_n)$ yields the best performance in the more general problem of linear MMSE estimation using a-priori information. We must satisfy that $L_e(c_n)$ and hence also $\hat{x}_k$ is not a function of $L(c_n)$ [8]. This is achieved by extending the approach in [8], which is to remove the influence of all $L(c_{mk+j})$, $j=0,...,(m-1)$, on $\hat{x}_k$ and to replace it with the influence of $L(c_{mk+j})=0$, $\forall j$. We assume that $\hat{x}_k$ exhibits a complex Gaussian distribution $p_{\hat{x}_k|x_k=s_i}(x)$, $x\in\mathbb{C}$, conditioned on $x_k=s_i$, $i=0,1,...,(2^m-1)$:

$$\begin{aligned} \mu_{i,k} &\triangleq E\{\hat{x}_k|x_k=s_i\}, \\ \sigma_{i,k}^2 &\triangleq \mathrm{Cov}(\hat{x}_k, \hat{x}_k|x_k=s_i), \\ p_{\hat{x}_k|x_k=s_i}(x) &\approx \frac{1}{\pi\,\sigma_{i,k}^2}\ \exp(-\frac{|x-\mu_{i,k}|^2}{\sigma_{i,k}^2}). \end{aligned}$$

In case all $s_i, h_i\in S$ are real, $p_{\hat{x}_k|x_k=s_i}(x)$ can be a single Gaussian pdf, i.e., $p_{\hat{x}_k|x_k=s_i}(x)=n_{\mu_{i,k},\sigma_{i,k}^2}(x)$, $x,\mu_{i,k}\in\mathbb{R}$. By averaging over all $p_{\hat{x}_k|x_k=s_i}(x)$ with $c_n=0$ or $c_n=1$, respectively, $L_e(c_n)$ is computed as

$$L_e(c_n) = \ln\frac{\sum_{s_i\in S:c_n=0} p_{\hat{x}_k|x_k=s_i}(\hat{x}_k)\cdot Pr\{x_k=s_i|c_n=0\}}{\sum_{s_i\in S:c_n=1} p_{\hat{x}_k|x_k=s_i}(\hat{x}_k)\cdot Pr\{x_k=s_i|c_n=1\}},$$

where $n=mk+j$ and $j=0,...,(m-1)$. This simplifies to

$$\varrho_{i,k} \triangleq \sigma_{i,k}^{-2}|\hat{x}_k-\mu_{i,k}|^2, \qquad (2)$$

$$L_e(c_n) = \ln\frac{\sum_{s_i\in S:c_n=0}\exp\left(-\varrho_{i,k}\right)\cdot\prod_{l=0,\,l\neq j}^{m-1} Pr\{c_{mk+l}=b_{i,l}\}}{\sum_{s_i\in S:c_n=1}\exp\left(-\varrho_{i,k}\right)\cdot\prod_{l=0,\,l\neq j}^{m-1} Pr\{c_{mk+l}=b_{i,l}\}},$$

using the fact that, as shown later, $\sigma^2_{i,k}$ does not depend on i. We assume that no additional a-priori information besides that of the decoder is available. Thus, in the first iteration we have $L(c_n)=0$, $\forall n$, and $L_e(c_n)$ can be computed without the terms $\prod_{\substack{l=0\\l\neq j}}^{m-1} Pr\{c_{mk+l}=b_{i,l}\}$.

4. Turbo equalization using MMSE linear equalization

4.1. Exact implementation

This approach was derived in [8] and applied to Turbo Equ. in [9]. The design rule to obtain the $f_{i,k}$ is to minimize the MMSE cost function $E\{|x_k - \hat{x}_k|^2\}$. In general, the estimate $\hat{x}_k$ is computed from

$$\mathbf{z}_k \triangleq \left[z_{(k+N_1)\bmod L}\; z_{(k+N_1-1)\bmod L} \cdots z_{(k-N_2)\bmod L}\right]^T,$$

a length N vector of received symbols, as follows (Appendix A in [8]):

$$\begin{aligned}
\bar{\mathbf{z}}_k &\triangleq \left[\bar{z}_{(k+N_1)\bmod L} \cdots \bar{z}_{(k-N_2)\bmod L}\right]^T,\\
\mathbf{v}_k &\triangleq \mathbf{Diag}[v_{(k+N_1)\bmod L} \cdots v_{(k-M+1-N_2)\bmod L}],\\
\mathbf{s} &\triangleq \mathbf{H}\left[\mathbf{0}_{1\times N_1} \quad 1 \quad \mathbf{0}_{1\times(N_2+M-1)}\right]^T,\\
\mathbf{f}_k &= (\sigma_w^2 \mathbf{I}_N + \mathbf{H}\mathbf{V}_k\mathbf{H}^H)^{-1}\,\mathbf{s},\\
\hat{x}_k &= \bar{x}_k + v_k\,\mathbf{f}_k^H(\mathbf{z}_k - \bar{\mathbf{z}}_k) \text{ (in general)},
\end{aligned}$$

where $\mathbf{H}$ is the $N\times(N{+}M{-}1)$ channel convolution matrix

$$\mathbf{H} \triangleq \begin{bmatrix} h_0 & h_1 \cdots h_{M-1} & 0 & \cdots & & 0\\ 0 & h_0 \;\; h_1 & \cdots & h_{M-1} & 0\cdots & 0\\ & & \ddots & & & \\ 0 & \cdots & 0 & h_0 & h_1\cdots h_{M-1} \end{bmatrix}.$$

Due to the mod $-L$ operation we circularly equalize on a "tail-biting" block $\bar{\mathbf{z}}$, which is possible using the prefix in the transmitter.

However, $\hat{x}_k$ depends on $L(c_{mk+j})$, $j=0,...,(m-1)$, over $\bar{x}_k$ and v_k. As mentioned, we set $L(c_{mk+j})=0$, $\forall j$, yielding $\bar{x}_k=0$, and $v_k=1$ and recompute $\mathbf{f}_k$ and $\hat{x}_k$ by replacing $\bar{x}_k$ and v_k with 0 and 1:

$$\begin{aligned}
\mathbf{f}'_k &= (\sigma_w^2\mathbf{I}_N + \mathbf{H}\,\mathbf{V}_k\,\mathbf{H}^H + (1-v_k)\,\mathbf{s}\,\mathbf{s}^H)^{-1}\,\mathbf{s},\\
\hat{x}_k &= 1\cdot \mathbf{f}'^H_k(\mathbf{z}_k - \bar{\mathbf{z}}_k + (\bar{x}_k - 0)\,\mathbf{s}).
\end{aligned}$$

We can express $\mathbf{f}'_k$ as scaled version of $\mathbf{f}_k$ using the matrix inversion lemma and $\Sigma_k \triangleq \sigma_w^2\mathbf{I}_N + \mathbf{H}\mathbf{V}_k\mathbf{H}^H$:

$$\mathbf{f}'_k = (\Sigma_k + (1-v_k)\,\mathbf{s}\,\mathbf{s}^H)^{-1}\,\mathbf{s} = (1+(1-v_k)\,\mathbf{s}^H\mathbf{f}_k)^{-1}\,\mathbf{f}_k.$$

In [8], a recursive algorithm was derived to compute Σ_k^{-1} from Σ_{k-1}^{-1} with a number of operations only proportional to N^2 and M^2. With the final expression for the estimates:

$$\hat{x}_k = (1+(1-v_k)\,\mathbf{s}^H\mathbf{f}_k)^{-1}\,\mathbf{f}_k^H(\mathbf{z}_k - \bar{\mathbf{z}}_k + \bar{x}_k\,\mathbf{s}),$$

we can compute the statistics $\mu_{i,k}$ and $\sigma^2_{i,k}$:

$$\begin{aligned}
\mu_{i,k} &= \frac{\mathbf{f}_k^H(E\{\mathbf{z}_k|x_k=s_i\} - \bar{\mathbf{z}}_k + \bar{x}_k\,\mathbf{s})}{D} = \frac{s_i\,\mathbf{f}_k^H\mathbf{s}}{D},\\
\sigma^2_{i,k} &= \frac{\mathbf{f}_k^H \mathrm{Cov}(\mathbf{z}_k,\mathbf{z}_k|x_k=s_i)\mathbf{f}_k}{|D|^2} = \frac{\mathbf{f}_k^H\mathbf{s}\,(1-v_k\,\mathbf{s}^H\mathbf{f}_k)}{|D|^2},
\end{aligned}$$

where $D = 1+(1-v_k)\,\mathbf{s}^H\mathbf{f}_k$. The derivation to obtain $L_e(c_n)$ is completed by finding an expression for $\varrho_{i,k}$:

$$\varrho_{i,k} = (\mathbf{f}_k^H\mathbf{s}\,(1-v_k\,\mathbf{s}^H\mathbf{f}_k))^{-1}\,|\,\mathbf{f}_k^H(\mathbf{z}_k - \bar{\mathbf{z}}_k + \bar{x}_k\,\mathbf{s}) - s_i\,\mathbf{f}_k^H\mathbf{s}\,|^2.$$

For the case that $L(c_n)=0$, $\forall n$, we have $\bar{x}_k = \bar{z}_k = 0$ and $v_k=1$, $\forall k$, yielding a time-invariant $\mathbf{f}_k \triangleq \mathbf{f}_{NA}$:

$$\begin{aligned}
\mathbf{f}_{NA} &\triangleq (\sigma_w^2\mathbf{I}_N + \mathbf{H}\mathbf{H}^H)^{-1}\mathbf{s},\\
\varrho_{i,k} &= (\mathbf{f}_{NA}^H\mathbf{s}\,(1-\mathbf{s}^H\mathbf{f}_{NA}))^{-1}|\mathbf{f}_{NA}^H\mathbf{z}_k - s_i\,\mathbf{f}_{NA}^H\mathbf{s}|^2,
\end{aligned}$$

where $_{NA}$ stands for "No A-priori information".

4.2. Approximate implementation

The costly computation of the vector $\mathbf{f}_k$ for each k is neglected by simply using the vector $\mathbf{f}_{NA}$ despite the presence of non-zero a-priori information $L(c_n) \neq 0$ [8]. The estimates $\hat{x}_k$ are now given by

$$\hat{x}_k = \mathbf{f}_{NA}^H(\mathbf{z}_k - \bar{\mathbf{z}}_k + \bar{x}_k\,\mathbf{s}).$$

However, computing the statistics $\mu_{i,k}$ and $\sigma^2_{i,k}$ becomes more difficult:

$$\begin{aligned}
\mu_{i,k} &= \mathbf{f}_{NA}^H(E\{\mathbf{z}_k|x_k=s_i\} - \bar{\mathbf{z}}_k + \bar{x}_k\,\mathbf{s}) = s_i\,\mathbf{f}_{NA}^H\mathbf{s},\\
\sigma^2_{i,k} &= \mathbf{f}_{NA}^H(\sigma_w^2\mathbf{I}_N + \mathbf{H}\,\mathbf{V}_k\,\mathbf{H}^H - v_k\,\mathbf{s}\,\mathbf{s}^H)\,\mathbf{f}_{NA}.
\end{aligned}$$

In [8], $\sigma^2_{i,k}$ was approximated by a crude time average. Here, the average is over all $\sigma^2_{0,k}$ (any i can be selected) corresponding to each x_k, $k=0,...,(L-1)$:

$$\begin{aligned}
\hat{\sigma}^2 &\triangleq \frac{1}{L}\sum_{k=0}^{L-1}\sigma^2_{0,k}\\
&= \sigma_w^2\mathbf{f}_{NA}^H\mathbf{f}_{NA} + \left(\frac{1}{L}\sum_{k=0}^{L-1} v_k\right)\mathbf{f}_{NA}^H(\mathbf{H}\,\mathbf{H}^H - \mathbf{s}\,\mathbf{s}^H)\mathbf{f}_{NA}.
\end{aligned}$$

The exponents $\varrho_{i,k}$ are given by

$$\varrho_{i,k} = \hat{\sigma}^{-2}|\,\hat{x}_k - s_i\,\mathbf{f}_{NA}^H\mathbf{s}\,|^2 \tag{3}$$

for general $L(c_n)\in\mathbb{R}$. For $L(c_n)=0$, $\forall n$, yielding $\bar{x}_k=0$ and $v_k=1$, $\forall k$, we have especially $\hat{\sigma}^2 = \mathbf{f}_{NA}^H\mathbf{s}\,(1-\mathbf{s}^H\mathbf{f}_{NA})$ and $\hat{x}_k = \mathbf{f}_{NA}^H\mathbf{z}_k$.

5. Turbo equalization using matched filtering

This approach was first introduced in [4] and modified yielding better results in [9]. The estimator filter coefficients are set to yield a matched filter to the ISI channel response $h[n]$. The algorithm in [9] to compute the estimate $\hat{x}_k$ is used without adaptation to a 2^m-ary signal alphabet:

$$
\begin{aligned}
d_i &= \sum_{l=0}^{M-1} h_l\, h_{l-i}^*, \quad i = -(M-1), ..., (M-1), \\
\hat{x}_k &= E_h\, \bar{x}_k + \sum_{i=0}^{M-1} h_i^*\, (z_{(k+i) \bmod L} - \bar{z}_{(k+i) \bmod L}), \\
\mu_{i,k} &= E_h\, s_i, \\
\sigma_{i,k}^2 &= E_h\, \sigma_w^2 - E_h^2\, v_k + \sum_{l=1-M}^{M-1} v_{(k-l) \bmod L}\, |d_l|^2.
\end{aligned}
$$

The exponents $\varrho_{i,k}$ for general $L(c_n) \in \mathbb{R}$ are given by

$$\varrho_{i,k} = \sigma_{i,k}^{-2} |\hat{x}_k - E_h\, s_i|^2. \tag{4}$$

For $L(c_n) = 0$, $\forall n$, we have especially $\hat{x}_k = \sum_{i=0}^{M-1} h_i^* z_{k+i}$ and $\sigma_{i,k}^2 = E_h\, \sigma_w^2 - E_h^2 + \sum_{l=1-M}^{M-1} |d_l|^2$.

6. Algorithm adaptation

The APPLE and the MF approach derived in Sections 4.2 and 5 require a number of operations per received symbol increasing only linearly with N or M, which is much better than LE (N^2) or MAP equalization (2^{mM}). However, both APPLE and MF suffer a significant performance loss compared to, e.g., LE. In [9], a scheme was proposed to properly switch between APPLE or MF depending on $L(c_n)$, which is unfortunately based on empirical performance evaluation requiring the transmitted data. We propose a novel approach using the signal-to-noise ratio (SNR) of $\hat{x}_k$ to decide, prior to equalization, which algorithm to use:

$$\text{SNR} = \frac{|E\{\hat{x}_k | x_k = s_i\}|^2}{\text{Cov}(\hat{x}_k, \hat{x}_k | x_k = s_i)} = \frac{|\mu_{i,k}|^2}{\sigma_{i,k}^2} = \frac{|K \cdot s_i|^2}{\sigma_{i,k}^2},$$

where $K = \mathbf{f}_{NA}^H \mathbf{s}$ for APPLE and $K = E_h^2$ for MF. We suggest that the algorithm yielding the highest average SNR

$$\bar{s} \triangleq \frac{1}{L \cdot 2^m} \sum_{k=0}^{L-1} \sum_{i=0}^{2^m-1} \frac{|K \cdot s_i|^2}{\sigma_{i,k}^2} = \frac{1}{L} \sum_{k=0}^{L-1} \frac{K^2}{\sigma_{0,k}^2} \geq \frac{K^2}{\frac{1}{L} \sum_{k=0}^{L-1} \sigma_{0,k}^2},$$

should be used. The average variance $\hat{\sigma}^2 \triangleq \frac{1}{L} \sum_{k=0}^{L-1} \sigma_{0,k}^2$ is computed as in Section 4.2 yielding the lower bounds

$$
\begin{aligned}
\bar{v} &\triangleq \frac{1}{L} \sum_{k=0}^{L-1} v_k, \\
\text{APPLE:} \quad \frac{K^2}{\hat{\sigma}^2} &= \frac{\mathbf{f}_{NA}^H \mathbf{s}\, \mathbf{s}^H \mathbf{f}_{NA}}{\sigma_w^2 \mathbf{f}_{NA}^H \mathbf{f}_{NA} + \bar{v} \cdot \mathbf{f}_{NA}^H (\mathbf{H}\mathbf{H}^H - \mathbf{s}\, \mathbf{s}^H) \mathbf{f}_{NA}}, \\
\text{MF:} \quad \frac{K^2}{\hat{\sigma}^2} &= \frac{E_h^2}{E_h\, \sigma_w^2 + \bar{v} \cdot (\sum_{l=1-M}^{M-1} |d_l|^2 - E_h^2)},
\end{aligned}
$$

on $\bar{s}$. Thus, the receiver uses the algorithm with largest $\frac{K^2}{\hat{\sigma}^2}$ for equalization. The bound is tight whenever $\sigma_{0,k}^2$ is constant in k, e.g., for $\bar{v} = 0$ (symbols x_k are known to the receiver) and $\bar{v} = 1$ (no a-priori information). The average SNR $\bar{s}$ and its lower bound $\frac{K^2}{\hat{\sigma}^2}$ are monotonically decreasing in $\bar{v} \in [0, 1]$ [8]. The maximum $\bar{s}$ is $\frac{E_h}{\sigma_w^2}$ using MF for $\bar{v} = 0$, which is the SNR of an AWGN channel with noise variance σ_w^2. We thus expect the Turbo Equ. system performance to be below than that of coded data transmission over the equivalent AWGN channel, since the equalizer is at most able to provide the same SNR of the estimates $\hat{x}_k$.

We will not consider LE as alternative algorithm due to the much larger computational effort. For MAP equalization, an analysis using $\bar{s}$ is not possible. We rely here, if applicable at all, on the EXIT charts introduced in [9, 11].

7. Frequency domain implementation

In [12], the APPLE and MF algorithm were implemented in the frequency domain. The adaptation to 2^m-ary signal alphabets is straightforward. Table 1 depicts this implementation (the DFT operator is the Discrete Fourier Transform).

Input :
- $[z_0 \cdots z_{L-1}]^T$, $[L(c_0) \cdots L(c_{L\cdot m-1})]^T$, $h[n]$, and σ_w^2,
Initialization :
$[Z_0 \cdots Z_{L-1}]^T \leftarrow$ DFT $[z_0 \cdots z_{L-1}]^T$
$[H_0 \cdots H_{L-1}]^T \leftarrow$ DFT $[h_0\, h_1 \cdots h_{M-1}\, \mathbf{0}_{1\times(L-M)}]^T$
$\mu \leftarrow \frac{1}{L} \sum_{k=0}^{L-1} \frac{\|H_k\|^2}{\sigma_w^2 + \|H_k\|^2}$
Prior to equalization :
- compute: $[\bar{x}_0 \cdots \bar{x}_{L-1}]^T$ and $[v_0 \cdots v_{L-1}]^T$,
- decide: use APPLE or MF by comparing $\frac{K^2}{\hat{\sigma}^2}$,
Equalization :
$[\bar{X}_0 \cdots \bar{X}_{L-1}]^T \leftarrow$ DFT $[\bar{x}_0 \cdots \bar{x}_{L-1}]^T$
APPLE: $\hat{X}_k \leftarrow \frac{H_k^* Z_k}{\sigma_w^2 + \|H_k\|^2} + (\mu - \frac{\|H_k\|^2}{\sigma_w^2 + \|H_k\|^2}) \bar{X}_k, \forall k$
MF: $\hat{X}_k \leftarrow H_k^* Z_k + (1 - \|H_k\|^2) \bar{X}_k, \forall k$
$[\hat{x}_0 \cdots \hat{x}_{L-1}]^T \leftarrow \text{DFT}^{-1}\, [\hat{X}_0 \cdots \hat{X}_{L-1}]^T$
Past to equalization :
- compute: $[L_e(c_0) \cdots L_e(c_{L\cdot m-1})]^T$.

Table 1. Frequency domain equalization.

8. Complexity comparison

In this section, the computational complexity of MAP equalization, LE, APPLE, and MF is compared. We assume that the statistics $\bar{x}_k$ and v_k are available for all k and skip the computation to obtain $L_e(c_n)$ including $\mu_{i,k}$ and $\sigma_{i,k}^2$ (both mappings $\bar{x}_k$, $v_k \leftarrow L(c_n)$ and $L_e(c_n) \leftarrow \hat{x}_k$ strongly depend on S). Any overhead due to initialization (one-time computations for all iterations), e.g., to compute $\mathbf{f}_{MA}$ for APPLE or H_k, Z_k, $\forall k$, is neglected. Table 2 gives the number of real multiplications and additions per iteration required to equalize L symbols z_k yielding L estimates $\hat{x}_k$. The DFT is carried out using a radix-2 FFT requiring roughly $2L \log_2(L)$ real multiplications and $2L \log_2(L)$ real additions for $L = 2^l$, $l = 1, 2, ...,$ [10]. For the complexity of LE see [8]. For MAP equalization, we considered only the computation of all γs and the α, β recursion [7].

approach	domain	real multiplications	real additions
MF	time	$L(10M)$	$L(10M-2)$
MF	frequency	$4L\log_2(L)+8L$	$4L\log_2(L)+2L$
APPLE	time	$L(4N+8M)$	$L(4N+4M-4)$
APPLE	frequency	$4L\log_2(L)+8L$	$4L\log_2(L)+2L$
LE	-	$L(16N^2+4M^2+10M-4N-4)$	$L(8N^2+2M^2-10N+2M+4)$
MAP equ.	-	$L(3\cdot 2^{mM}+2\,m\,2^{m(M-1)})$	$L(3\cdot 2^{mM}+2(m-1)2^{m(M-1)})$

Table 2. Computational complexity of equalization per iteration per block.

9. Results and Conclusions

We tested the bit error rate performance (simulation of at least 1000 data bit errors) of a Turbo Equ.-based receiver. Data is encoded (code generator $G(D) = (1, \frac{1+D^2}{1+D+D^2}))$ to length $L = 2048$ blocks of code symbols c_n including S-random (S=30) interleaving [3]. The c_n are modulated to x_k according to Table 3 (includes also Eqs. (1) and (2)). The time-invariant channel impulse response is $h[n]=0.227\,\delta[n]+0.46\,\delta[n-1]+0.688\,\delta[n-2]+0.46\,\delta[n-3]+0.227\,\delta[n-4]$. The system SNR is $\frac{E_b}{N_0}=\frac{1}{2\,\sigma_w^2}$. The filter parameters for LE and APPLE are $N_1=9$ and $N_2=5$. Figure 2 depicts the BER results after 5 iterations: MAP equalization and LE perform best followed by switched APPLE/MF. Using APPLE and MF alone is not satisfactory. Similar results were obtained for unknown $h[n]$ (including training) and/or fading coefficients h_i.

$(c_{2k}\,c_{2k+1})$	00	10	01	11
x_k	$\frac{1+\imath}{\sqrt{2}}$	$\frac{-1+\imath}{\sqrt{2}}$	$\frac{1-\imath}{\sqrt{2}}$	$\frac{-1-\imath}{\sqrt{2}}$

$$\bar{x}_k \leftarrow \frac{1}{\sqrt{2}}\left(\tanh\frac{L(c_{2k})}{2} + \tanh\frac{L(c_{2k+1})}{2}\,\imath\right)$$
$$v_k \leftarrow 1-|\bar{x}_k|^2 \qquad (\imath=\sqrt{-1})$$
$$L_e(c_{2k}) \leftarrow \sqrt{8}|\mu_{0k}|\sigma_{0k}^{-2}\,\Re\{\hat{x}_k\}$$
$$L_e(c_{2k+1}) \leftarrow \sqrt{8}|\mu_{0k}|\sigma_{0k}^{-2}\,\Im\{\hat{x}_k\}$$

Table 3. QPSK modulation.

In conclusion, the novel switched APPLE/MF approach yields a comfortable gain to one-time MAP equalization and decoding. We think, that this and the LE algorithm are most suitable for low coplexity Turbo Equ. using higher order signal alphabets, e.g., 8PSK in the the Enhanced Data rates for GSM evolution (EDGE) standard. Part of on-going work is an accurate performance analysis and the implementation of channel parameter estimation into the iterative algorithm.

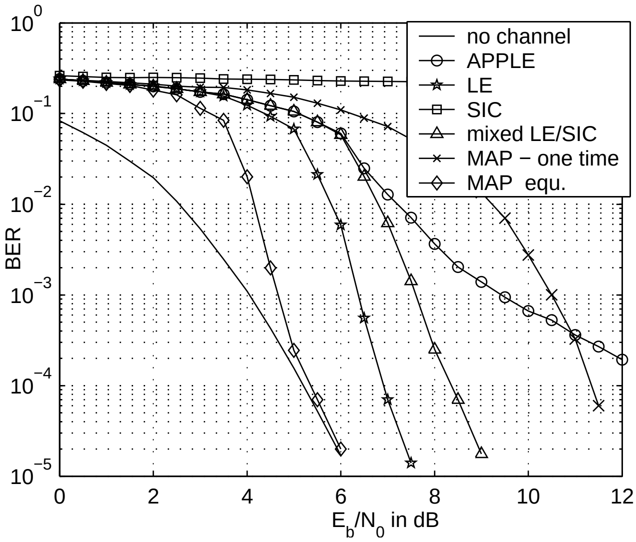

Figure 2. BER performance comparison.

References

[1] A. Anastasopoulos and K. Chugg. Iterative equalization/decoding for TCM for frequency-selective fading channels. *Record on the 31th Asilomar Conf. on Signals, Systems & Computers*, 1:177–181, November 1997.

[2] G. Bauch and V. Franz. A comparison of soft-in/soft-out algorithms for "turbo detection". *Proceedings on the Intern. Conf. on Telecomm. (ICT '98)*, pages 259–263, June 1998.

[3] C. Heegard and S. Wicker. *Turbo Coding*. Kluwer Academic Publishing, Boston, 1999.

[4] A. Glavieux, C. Laot, and J. Labat. Turbo equalization over a frequency selective channel. *Intern. Symposium on Turbo codes & related topics*, pages 96–102, Sep 1997.

[5] C. Douillard et al. Iterative correction of intersymbol interference: Turbo equalization. *European Trans. on Telecomm.*, 6(5):507–511, Sep-Oct 1995.

[6] D. Raphaeli and A. Saguy. Linear equalizers for turbo equalization: A new optimization criterion for determining the equalizer taps. *Proc. of the 2nd Intern. Symp. on Turbo Codes & Related Topics*, pages 371–374, Sep 2000.

[7] L.R. Bahl et al. Optimal decoding of linear codes for minimizing symbol error rate. *IEEE Transactions on Information Theory*, 20:284–287, March 1974.

[8] M. Tüchler, A. Singer, and R. Kötter. Minimum mean squared error (MMSE) equalization using priors. *submitted to IEEE Transactions on Signal Processing*, 2000.

[9] M. Tüchler, R. Kötter, and A. Singer. Turbo equalization: principles and new results. *submitted to IEEE Trans. on Communications*, 2000.

[10] J. Proakis and D. Manolakis. *Digital Signal Processing, 3rd Ed.* Prentice Hall, Upper Saddle River, New Jersey, 1996.

[11] S. ten Brink. Convergence of iterative decoding. *Electronic Letters*, 35(10):806–808, May 1999.

[12] M. Tüchler and J. Hagenauer. Turbo equalization using frequency domain equalizers. *Proc. of the Allerton Conference, Monticello, IL, U.S.A.*, October 2000.