EXIT CHART ANALYSIS APPLIED TO ADAPTIVE TURBO EQUALIZATION

Roald Otnes
Unik — University Graduate Center
PO box 70, N-2027 Kjeller, Norway
roaldo@unik.no

Michael Tüchler
Munich University of Technology
Arcisstr. 21, D-80290 Munich, Germany
michael.tuechler@ei.tum.de

ABSTRACT
In this paper we consider iterative channel estimation, equalization, and decoding, or adaptive Turbo equalization, as a receiver technology for digital communication systems where the channel imposes time-varying intersymbol interference. We show how a semianalytical technique called EXIT charts, originally developed by S. ten Brink for analysis of Turbo codes, can be used to predict the performance of the iterative receiver for such systems. To demonstrate the usefulness of the technique, we use EXIT charts to address interesting questions about adaptive Turbo equalization for time-varying channels: Which pattern of training sequences should be used, what is the performance difference between optimal MAP equalization and suboptimal linear equalization, and between the case of a known channel and the case of an estimated channel, and what can be gained by using a recursive precoder in conjunction with the symbol mapper.

1. INTRODUCTION
In this paper, we address the problem of digital communications over a channel imposing time-varying intersymbol interference (ISI) as well as additive white Gaussian noise. We assume a transmitter consisting of an error-correcting code (ECC), an interleaver shuffling the code bits, a symbol mapper mapping the interleaved code bits onto an $M$-ary signal constellation (PSK or QAM), and modulation onto a single carrier frequency. A block diagram of such a system is shown in Fig. 1 including the receiver, which is described below. High frequency (HF, 3-30 MHz) or mobile communications systems (GSM or EDGE) are examples of such system structures.

For the given system and channel model, conventional receivers perform adaptive equalization including (soft) symbol demapping, deinterleave the (soft) information on the code bits (e.g. log-likelihood ratios, LLRs, [1]) from the demapper, and perform (soft-in) decoding of the ECC. Commonly used equalizers for such receivers are the trellis-based maximum likelihood sequence estimator (MLSE) [2] or the decision feedback equalizer (DFE) [3].

An optimal receiver would jointly perform channel estimation, equalization, and decoding, which is an extremely complicated problem especially when an interleaver is present. Recent work attacks this problem using iterative receiver algorithms usually referred to as Turbo equalization or iterative equalization and decoding. This principle to approach the performance of joint equalization and decoding was first introduced in [4] for a known channel impulse response (CIR) as an extension of iterative decoding of concatenated codes. In a receiver employing Turbo equalization, the equalizer and decoder are soft-in soft-out (SISO) modules, which were initially based on the maximum a posteriori probability (MAP) [5] or MLSE algorithm [1] (see for example [4, 6]). Due to the large computational complexity and memory requirements of these algorithms, suboptimal SISO equalizers have been introduced later, e.g., based on soft ISI cancellation and linear filters [7–10]. When the CIR is unknown and time-varying, iterative equalization and decoding can be extended to iterative equalization, estimation, and decoding. However, in existing work [11–13], the necessary SISO joint equalizer and estimator is very complex.

We attempt to reduce the computational burden of these schemes by separating out equalizer and estimator and carefully perform iterative equalization, estimation, and decoding over three receiver modules. We consider a separate estimator providing estimates of the CIR and a separate equalizer using the CIR estimates to obtain estimates of the transmitted data, see Fig. 2. The advantage of this construction is a high flexibility in choosing estimation and equalization algorithms. Joint equalization, estimation, and decoding is performed by iterating equalization and decoding tasks, where the estimator improves its CIR estimates by incorporating feedback information from the decoder by transforming the code bit LLRs $L_d(c_k)$ into soft training symbols $\bar{y}_n = E\{y_n\}$ (the expectation given the LLRs $L_d(c_k)$ about the unknown transmitted symbols $y_n$). The improvement of the CIR estimates over the iterations has been analyzed in [14–16]. Moreover, this work shows that incorporating $E\{y_n\}$ for estimation is better than using hard-decisions $\hat{y}_n$ derived from the $L_d(c_k)$. Clearly, better CIR estimates improve the performance of the SISO equalizer and finally that of the entire system performance. For linear SISO equalizers, an alternative to separate estimation and equalization is to update the equalizer coefficients directly using common algorithms for adaptive equalization, incorporating soft information fed back from the decoder [17]. In this case, the coefficient update rule extends the algorithm to compute an approximate version of the optimal linear SISO equalizer in [10] to the case that the CIR is time-varying. We note that re-
luted algorithms for Turbo equalization given unknown, time-varying channel characteristics have been proposed, e.g., for HF communications [18–20], for GSM [21], for GPRS [22,23], and for EDGE [17].

Analytical expressions for the performance of a receiver employing Turbo equalization are not known, and the task of deriving analytical expressions is even more complicated when channel estimation is included in the iterative process. Therefore, the performance of Turbo equalization for a particular system is usually studied via time-consuming simulations. Alternatively, a semi-analytical approach called extrinsic information transfer charts (EXIT charts) [24] can be used, which obtains mutual information transfer functions separately for each SISO module in the receiver via off-line simulations. These transfer function maps a parameter of the distribution of the LLRs entering a SISO module, the mutual information to the corresponding input on the transmitter side, to the parameter of the distribution of the LLRs exiting a SISO module. Because the output of one SISO module is input to the other SISO module and vice versa, the two transfer function can be drawn in a two-dimensional chart and the performance of the receiver is given by a trajectory between the two curves [24]. The benefits of this method are to predict the system performance without actually simulating data transmission and to gain insight about the convergence of the iterative procedure. The benefit of using EXIT charts to analyze the performance of Turbo equalization was presented in [10] for the case of a known, time-invariant channel. In this paper, we show how EXIT charts can be applied to receivers performing iterative equalization, estimation, and decoding, which is referred to as adaptive Turbo equalization in the sequel.

2. SYSTEM MODEL

Consider the system model in Fig. 1. A block of data bits $a_n$ is encoded with a rate-$R_c$ convolutional code to $N_{int}$ code bits $c_{k'} \in \{+1,-1\}$ including trellis termination. The bits $c_{k'}$ are interleaved to $c_k$, which are mapped to $N_{int}/Q$ data symbols $y_n$ from a $2^Q$-ary signal constellation. The data symbols $y_n$ are multiplexed with $N_t$ training symbols $t_n$ known to the receiver to form a block of $N_{block} = N_{int} + N_t$ symbols $x_n$. The overall code rate including coding and training overhead is $R = R_cN_{int}/(N_{int} + QN_t)$. The $x_n$ are transmitted over the time-varying ISI channel with equivalent discrete-time CIR $h_n = [h_{n,0} \cdots h_{n,M-1}]$. The length of the CIR is $M$ symbol intervals. The received symbols $z_n$ are given by

$$z_n = w_n + \sum_{l=0}^{M-1} h_{n,l} x_{n-l},$$

where the $w_n$ are complex white Gaussian noise samples with variance $\sigma_w^2$, i.e., their probability density function (pdf) is $p(w) = \exp(-|w|^2/2\sigma_w^2) / (\pi\sigma_w^2)$. The (adaptive) SISO equalizer processes the received symbols $z_n$, the training symbols $t_n$, and the a priori LLRs $L_d(c_k)$ fed back from the decoder. During first-time equalization, all $L_d(c_k)$ are zero. The equalizer outputs the LLRs $L_o(c_k)$, which, deinterleaved to $L_o(c_k')$, are input to the SISO decoder. The SISO decoder computes estimates $\hat{a}_n$ of the transmitted data bits and outputs the LLRs $L_d(c_k)$, which, interleaved to $L_d(c_k')$, are used as a priori LLRs by the SISO equalizer in the next iteration.

3. EXIT CHARTS FOR ADAPTIVE TURBO EQUALIZATION

The EXIT chart describes the convergence of the iterative receiver algorithm by investigating the exchange of mutual information between the equalizer and the decoder. Observed is the mutual information $I_o = I(L_o(c_k);c_k)$ at the equalizer output and $I_d = I(L_d(c_k');c_k)$ at the decoder output, where the mutual information is defined as

$$I(L;C) = \frac{1}{2} \sum_{c \in \{+1,-1\}} \int_{-\infty}^{\infty} p(l|c) \log_2 \frac{2p(l|c)}{p(l|+1)+p(l|-1)} \, dl,$$

where $p(l|c)$ denotes the pdf of the LLRs modelled as outcomes of the random variable (r.v.) $L$ conditioned on its corresponding transmitted bit $c$ represented by the r.v. $C$. The mutual informations $I_o$ and $I_d$ can be found by calculating the integral above numerically from histograms of $c_kL_o(c_k)$ and $c_kL_d(c_k')$, respectively. These histograms are used to estimate $p(l|+1)$ and $p(l|-1)$ [24], which...
mutual information transfer functions

Figure 3: Simulation setup for generating mutual information transfer functions.

(a) Decoder

(b) Equalizer

Figure 4: Transfer function (solid lines) from $I_e$ (input) to $I_d$ (output), and bit error rate (dashed lines) as function of $I_e$, of a MAP-based decoder for rate-1/2 convolutional codes with memory 2 or 6.

can usually be assumed equal. The range of $I_d$ and $I_e$ is $[0, 1]$, where $I_d = 0$ or $I_d = 1$ correspond to no or perfect knowledge about $c_k$, given $L_d(c_k)$. The evolution of $I_d$ and $I_e$ over the iterations in a real receiver is called the system trajectory of the iterative algorithm.

The purpose of the EXIT chart is to predict the system trajectory from mutual information transfer functions, without performing simulations on the complete iterative receiver. The simulation setup to generate the transfer function of the SISO decoder is shown in Fig. 3 (top). Code bits $c_k$ are generated by encoding a block of independent data bits $a_m$ with the ECC. The input LLRs $L_e(c_k)$ to the decoder are drawn independently from a normal distribution with average value $c_k \sigma_L^2/2$ and variance $\sigma_L^2$. This is a common distribution of LLRs in an iterative receiver [24], and a one-to-one relationship exists between $\sigma_L^2$ and $I_e$. In this paper we use the optimal MAP algorithm [5] for SISO decoding. A number of blocks is simulated for different values of $\sigma_L^2$ in order to estimate the output mutual information $I_e$ and the bit error rate of the data bit estimates $\hat{a}_m$. The result is shown in Fig. 4 for two different rate-1/2 convolutional codes: a memory-2 code with generator polynomials $1+D^2$ and $1+D+D^2$, and a memory-6 code with generator polynomials $1+D+D^3+D^5+D^9$ and $1+D^3+D^5+D^9$. Note that the stronger code (memory-6) has a steeper transfer function than the weaker code, and therefore requires a lower input $I_e$ in order to produce high-quality output LLRs and low bit error rates. An increase/decrease in code rate $R_c$ would move all curves in Fig. 4 towards the right/left.

The transfer function of the equalizer is similarly generated using the setup shown in Fig. 3 (bottom). A block of $N_{int}$ independent bits $c_k$ is passed through all blocks in Fig. 1 from the symbol mapper to the received symbols $z_n$. The ECC generating the $c_k$ is omitted and dependencies within the interleaved bits $c_k$, which exist in a real system, are neglected. In a parallel path, the $c_k$ are multiplied by independent samples from a normal distribution with average value $\sigma_L^2/2$ and variance $\sigma_L^2$, to form the input LLRs $L_d(c_k)$. Since the adaptive SISO equalizer processes also received symbols $z_n$, its transfer function will depend on $E_b/N_0$ in contrast to the SISO decoder. Fig. 5 shows the transfer function for the following example: The block of transmitted symbols $x_n$ consists of 248 initial known training symbols, followed by 256 data, 31 training, 256 data, 31 training symbols and so on, until the end of the block. The signal constellation is Gray-coded 8-PSK, and the CIR is of the form $h_n = [h_{n,0} 0 0 0 0 h_{n,5}]$. The two nonzero taps $h_{n,0}$ and $h_{n,5}$ are drawn according to independent Rayleigh fading with a Gaussian Doppler spectrum [25] having a $2\nu$ Doppler spread (fading rate) of $1/(2400 T_s)$, where $T_s$ is the symbol period. The adap-
The EXIT chart in Fig. 6 (left) combines the equalizer transfer function at 10 dB $E_b/N_0$ and the decoder transfer function for the memory-2 code. Since the output LLRs from the equalizer are input to the decoder and vice versa, both transfer functions are drawn in the same plot with the axes being flipped for the decoder transfer function. The system trajectory of the iterative receiver can be predicted by representing each equalization task as a vertical line and each decoding task as a horizontal line, bouncing between the two transfer functions (the dashed lines in Fig. 6). The trajectory starts in the (0,0) point and approaches a fixed point, which is the leftmost crossing point of the two transfer functions. Each input mutual information $I_e$ to the decoder is associated with a bit error rate as shown in Fig. 4 and (with flipped axes) in Fig. 6 (right). We find from the EXIT chart in Fig. 6 that at 10 dB $E_b/N_0$, iterative decoding converges after 4 times equalization and decoding to a bit error rate of $3 \cdot 10^{-4}$ corresponding to $I_e = 0.82$ in the fixed point.

To verify the predicted performance, we track the trajectory of a real system by calculating the mutual information at the output of equalizer and decoder after each iteration. We use an interleaver of length $N_{int} = 24576$. The real system trajectory at 10 dB $E_b/N_0$ is shown as solid lines in Fig. 6. We see that the actual performance is close to the predicted performance for the first few iterations, but they depart at later iterations. We find that in reality, equalization and decoding tasks must be performed 5 or 6 times in order for the receiver to converge, i.e. one or two times more than the prediction of 4. The simulated bit error rate is shown in Fig. 7. At 10 dB $E_b/N_0$ the bit error rate after convergence is $7 \cdot 10^{-4}$, which is a bit more than the predicted bit error rate. The EXIT chart analysis assumes that all LLRs are independent, and is therefore in theory valid only for an infinite interleaver length $N_{int}$. This is the reason for the slight differences between predicted and actual performance in the example discussed above, and we find that the differences become smaller when $N_{int}$ is increased. For time-varying channels, $N_{int}$ should be large compared to the fading rate, i.e., there should be several fades within each interleaver block, in order to ensure that all blocks have the same statistical properties such that the EXIT chart analysis becomes exact. For smaller $N_{int}$, the performance predicted by the EXIT chart can be viewed
as a practical bound on the achievable performance.

4. USING THE EXIT CHART

Now that we have demonstrated by an example how EXIT charts can be applied to adaptive Turbo equalization, we are ready to discuss their benefits, some of which are

1. The EXIT chart is a nice way to visualize what happens in an iterative receiver, which may otherwise seem “magic” to people not familiar to the subject.
2. We can investigate the effect of modifying one of the two SISO algorithms without performing simulations on the complete iterative receiver.
3. We can see how well the ECC/decoder and channel/equalizer (“outer” and “inner” code) are fitted to each other.
4. The transfer function of the equalizer can be used to calculate the maximum achievable code rate for specific channel conditions.

The first point was demonstrated thoroughly by the example in the previous section. To demonstrate the remaining points, we give some further examples.

In Fig. 8 we show the effect of changing the pattern of training symbols. When the interval between training sequences is increased, the overall code rate $R$ will increase, causing the curves to move upward at high $I_d$. However, the channel estimate will be poorer when $I_d$ is low because it fails in tracking the channel variations between training sequences, and the curves with the most training have the highest $I_e$ in the left part of the EXIT chart. Finding the optimal training pattern is therefore a trade-off between initial convergence at low $E_b/N_0$ and error rates after convergence at high $E_b/N_0$, and for this example 128 or 256 data symbols between each training sequence seem to be good choices. The transfer function for the two rate-1/2 codes discussed earlier are also shown in this EXIT chart. Note that the memory-2 code has a transfer function almost parallel to most equalizer transfer functions, while the memory-6 code has a “knee” at low $I_d$ which may introduce a fixed point at low $I_d$ when $E_b/N_0$ is decreased compared to this plot. Therefore, a strong outer code is not always a good choice when using Turbo equalization in the receiver: When $E_b/N_0$ is decreased, the stronger code will fail before the weaker code in terms of initial convergence because the weaker code is better matched to the equalizer. It is possible to construct a code matched to a given equalizer transfer function according to some optimization criterion, e.g., to obtain values of $I_d$ and $I_e$ as high as possible after a certain number of iterations [26].

In Fig. 9 we have compared the equalizer transfer function for different scenarios. The example is the same as in Sec. 3, except that the channel length has been reduced to $M = 3$, $h_n = [h_{n,0} 0 h_{n,2}]$, to make MAP equalization feasible. We see that at $I_d = 1$, MAP equalization has the same performance as the much simpler approach of soft ISI cancellation combined with linear equalization (LE), for the case of a known channel as well as for an estimated channel. Thus, the performance using LE is similar to the performance using MAP equalization, after convergence of the iterative receiver. The major differences between LE and MAP equalization are the number of iterations needed before convergence, and the initial convergence at low $E_b/N_0$. We also find that the transfer functions when the channel is known at all times are much flatter than when the channel is estimated, indicating that there is much more to gain from doing iterations when the channel estimate is improved over the iterations. At $I_d = 1$, the curves for a known and for an estimated channel are
quite close.

We have in Fig. 9 also investigated the use of a recursive precoder in conjunction with the symbol mapper. In [27,28] is shown that Turbo equalization can have performance similar to Turbo-coded systems when a recursive precoder is used: A “Turbo cliff” at a certain $E_b/N_0$, above which the bit error rate is very small (negligible compared to the error rates in Fig. 7). Here we have used the simple precoder defined by $s_{n,i} = s_{n-1,i} \cdot \text{C}_{\text{In}+i}$, $i \in \{0,1,2\}$, where three precoded bits $s_{n,i}$ are mapped onto each symbol $y_n$ using an 8-PSK constellation. Note that the transfer function goes to the (1,1) point when using a precoder (here, the channel is known). This explains the Turbo cliff behavior: When the iterative decoding does converge, it converges to the (1,1) point where error rates are negligible. The price to pay compared to the non-precoded case is that $I_d$ is decreased at low $I_d$, such that the $E_b/N_0$ required for convergence is increased if the same ECC is used. Also, non-precoded Turbo equalization can be used as a receiver technology in present-day communication systems designed without Turbo equalization in mind, whereas the introduction of recursive precoding requires changes to the transmitter. Future work in precoded Turbo equalization includes complexity reduction by deriving algorithms based in linear filters, and the inclusion of channel estimation.

It can be shown [29, 30] that the maximum rate of an outer code which stays underneath the transfer function of the equalizer, such that the iterative receiver can converge, is equal to the area under that transfer function. An interesting point to note is that the area under the curve for a precoded system is exactly the same as the area under the curve for the corresponding non-precoded system [26], such that the same overall rate can be achieved for both systems if the ECC is optimized.

5. REFERENCES