A Novel Reduced Complexity Forward-Backward Equalization Algorithm*

Nitin Nangare  
Dept. of Electrical Engineering  
Texas A&M University  
College Station, TX 77840-3128  
nitin@ee.tamu.edu

Fredy Neeser  
IBM Research, Zurich Research Laboratory  
Säumerstrasse 4 / Postfach  
Rüschlikon CH-8803, Switzerland  
nfd@zurich.ibm.com

Krishna R. Narayanan  
Dept. of Electrical Engineering  
Texas A&M University  
College Station, TX 77840-3128  
krn@ee.tamu.edu

Abstract

The complexity of channel equalization using the BCJR algorithm grows exponentially with the channel memory and signal cardinality. In this paper we present a novel reduced complexity equalization algorithm where independent forward and backward trellises are formed based on decision feedback sequence estimation (DFSE) idea. Performance of the new equalizer algorithm is evaluated for turbo equalization of the EDGE (Enhanced Data rate for GSM Evolution) system. We also consider the effect of delayed APP calculation on BER performance and show that BER performance can be improved by choosing a proper delay value depending on the channel impulse response.

1 Introduction

BCJR [2] algorithm runs forward and backward over the trellis structure and can be used to calculate the a-posteriori probability (APP) of the symbol transmitted over an inter-symbol interference (ISI) channel. For EDGE system where 8-PSK modulation is used, the full state trellis contains $8^M$ states, where $M$ denotes the ISI channel memory. Reduced complexity equalization for EDGE using DFSE [3] based BCJR algorithm was proposed in [4]. In [4] backward pass is run over the same trellis as forward pass to achieve consistent state definition and hence backward pass is always biased by the decisions made in the forward pass. This bias results in severe performance loss if the channel energy is not concentrated in the initial taps. Effect of this bias on BER performance can be reduced by minimum phase filtering before equalization [3,5]. In order to remove this bias completely, we need to form independent forward and backward trellises. In [6] independent forward and backward trellises are formed using M-BCJR algorithm. In this

*This research was supported in part by IBM Research Lab at Zurich, Switzerland and part of the work was submitted as master’s thesis at Technical University of Munich, Germany [1].
paper we consider DFSE based BCJR algorithm as our reduced state search algorithm and show that BER performance can be improved by selecting independent active states in forward and backward pass. Based on different APP calculations we propose two implementation of the reduced complexity algorithm called matched full state (MFS) equalizer and matched reduced state (MRS) equalizer. This paper is organized as follows. Section 2 discusses the soft-in-soft-out equalizer based on BCJR algorithm and describes the reduced state equalizer similar to [4]. Section 3 describes the MFS equalizer. Concept of delayed APP calculation is introduced in this section. MRS equalizer is described in section 4. Simulation results for turbo equalization of an EDGE system using MFS and MRS equalizers are given in section 5. Proposed equalizer algorithms are analyzed using EXIT chart technique in this section.

2 SISO Equalization

System model for equalization is shown in Fig. 1. Signal $y_t$, received over an ISI channel with channel impulse response $h(t) = [h_0(t), \ldots, h_M(t)]$ and corrupted by complex AWGN noise $w_t \in (0,N_0)$, can be written as, $y_t = \sum_{i=0}^{M} h_i(t) \cdot v_{t-i} + w_t$. Convolutional decoding and equalization is usually performed separately where convolutional code and ISI channel are treated as serially concatenated codes. Performance of this serially concatenated scheme can be improved drastically by using turbo principle [7,8] which iterates the extrinsic information between decoder and equalizer in a process known as turbo equalization [9]. An input sequence $u \triangleq u_0 \cdots u_{T-1}$, where $u_t = [u_{t-0} \cdots u_{t-N-1}]$ is a binary $N$-tuple, is transmitted. The signal mapper in Fig. 1, maps $u_t$ to one of the complex valued signals from $2^N$-PSK signal constellation. We define this mapping function $\mu$ as, $v_t = \mu(u_t)$. Assuming time-invariant ISI channel, we can write the output $x_t$ of an ISI channel as, $x_t = \sum_{m=0}^{L} h_m \cdot v_{t-m} = \sum_{m=0}^{L} h_m \cdot \mu(u_{t-m})$. We define the state of the ISI channel at time $t$ as,

$$s_t \triangleq [u_{t-M}, \ldots, u_{t-2}, u_{t-1}]$$

and the state at time $t+1$ as,

$$s_{t+1} \triangleq [u_{t-M+1}, \ldots, u_{t-1}, u_t]$$

Using the state $s_t$ and input $u_t$, we define the extended state at time $t$ as, $s_t^{(Ext)} \triangleq [u_{t-M}, \ldots, u_{t-1}, u_t]$. For BCJR algorithm in log domain we define the log-probabilities as,

$$A_t(s_t) = \ln P_{s_t,s_0}^{y_0} \{s_t, y_0^{t-1} \}, \quad B_t(s_t) = \ln P_{y_t^{T-1}|s_t} \{y_T^{T-1}|s_t \}, \quad \text{and} \quad \Gamma_t(s_t, s_{t+1}) = \ln P(s_{t+1}|y_t|s_t).$$

Forward and backward probabilities are calculated recursively using (3) and (4).

$$A_{t+1}(s_{t+1}) = \sum_{s_t \rightarrow s_{t+1}} \exp \{ A_t(s_t) + \Gamma(s_t, s_{t+1}) \}$$

$$B_t(s_t) = \sum_{s_{t+1} \rightarrow s_{t+1}} \exp \{ B_{t+1}(s_{t+1}) + \Gamma(s_t, s_{t+1}) \}$$

Figure 1: System model for equalization
Finally the APPs, \( \Lambda(u_{t,i}) = \ln \left\{ \frac{P_{U_{t,i} \mid Y_{0:T-1}(u_{t,i}=1|y_0^{T-1})}}{P_{U_{t,i} \mid Y_{0:T-1}(u_{t,i}=0|y_0^{T-1})}} \right\} \) of constituent bits of \( u_t \) can be calculated as,

\[
\Lambda(u_{t,i}) = \ln \sum_{s_{t+1} \in \mathcal{S}_{t+1}^R} \exp\{A_{t+1}(s_{t+1}) + B_{t+1}(s_{t+1})\} - \ln \sum_{s_{t+1} \in \mathcal{S}_{t+1}^R} \exp\{A_{t+1}(s_{t+1}) + B_{t+1}(s_{t+1})\}
\]  

(5)

In order to reduce the number of states involved in (3),(4) and (5), we can use DFSE idea for MAP equalization [4]. To achieve this we divide the full state (1) into two parts, namely reduced state \( s_t^r = [u_{t-D} \cdots u_{t-1}] \) and path history \( \pi_t(s_t^r) = [\hat{u}_{t-M} \cdots \hat{u}_{t-D-1}] \), where \( D \) denotes the depth of the reduced state \( s_t^r \) and \( 1 \leq D \leq M \). For an EDGE system, the number of states in reduced state trellis become \( 8^D \). Using reduced state \( s_t^r \), its associated path history \( \pi_t(s_t^r) \) and current input \( u_t \), we can form the hypothetical full state \( s_t \) and hypothetical extended state \( s_t^{(Ext)} \) at time \( t \) as shown in (6) and (7).

\[
s_t = [\pi_t(s_t^r) : s_t^r] = [\hat{u}_{t-M} \cdots \hat{u}_{t-D-1} : u_{t-D} \cdots u_{t-1}]
\]  

(6)

\[
s_t^{(Ext)} = [\hat{u}_{t-M} \cdots \hat{u}_{t-D-1} : u_{t-D} \cdots u_{t-1} \ u_t]
\]  

(7)

Using this hypothetical extended state and noting that the complex AWGN channel has a transition probability \( P_{Y|X}(y|x) = \frac{1}{\pi N_0} \cdot e^{-|y-x|^2/2N_0} \), we can calculate the log-branch transition probability as follows,

\[
\hat{\Gamma}_t(s_t^{(Ext)}) = \sum_{i=0}^{N-1} u_{t,i} \cdot L_{U_{t,i}} - \frac{1}{N_0} \left| x_t - \sum_{i=0}^{D} h_i \cdot \mu(u_{t-i}) - \sum_{i=D+1}^{M} h_i \cdot \mu(\hat{u}_{t-i}) \right|^2
\]  

(8)

This DFSE idea cannot be directly applied to the BCJR algorithm. Backward pass may result in decisions that are inconsistent with those made during the forward pass and hence it is not always possible to calculate the log-branch transition probability during the backward pass. In the reduced state equalizer (DFSE-MAP) presented in [4], this issue is resolved by making the decisions about the path history bits only during the forward pass. Then the backward pass is run over the same forward sub-trellis. Thus the forward pass and the backward pass use the same equation (8) to calculate the log branch transition probability.

### 3 Matched Full State (MFS) Equalizer

In MFS equalizer we make independent decisions about the path history bits in forward as well as in backward pass. To make this possible, we need to define different reduced states in forward pass and in backward pass. To define the reduced state of depth \( D \) in forward pass, we divide the full state (1) into forward state \( s_t^{(f)} \) and forward path history \( \pi_t(s_t^{(f)}) \) given by (10) and (11) respectively.

\[
s_t = [\pi_t(s_t^{(f)}) : s_t^{(f)}]
\]  

(9)

\[
s_t^{(f)} = [u_{t-D} \ u_{t-D+1} \cdots u_{t-1}]
\]  

(10)

\[
\pi_t(s_t^{(f)}) = [\hat{u}_{t-M}^{(f)} \hat{u}_{t-M+1}^{(f)} \cdots \hat{u}_{t-D-1}^{(f)}]
\]  

(11)
We note that the forward reduced state equals the reduced state in case of DFSE-MAP equalizer. Using the state \( s_t^{(f)} \), its associated path history \( \pi_t(s_t^{(f)}) \) and current input \( u_t \), we can form hypothetical extended state \( s_t^{(Ext)} \) at time \( t \) as \( s_t^{(Ext)} = [\pi_t(s_t^{(f)}):u_t] = [\hat{u}_{t-M}^{(f)} \cdots \hat{u}_{t-D-1}^{(f)} u_{t-D} \cdots u_{t-1} u_t] \). Using this \( s_t^{(Ext)} \), we can calculate the forward log-branch transition probability as follows,

\[
\Gamma_t^{(f)}(s_t^{(Ext)}) = \sum_{i=0}^{N-1} u_{t,i} \cdot L_{U_{t,i}} - \frac{1}{N_0} \left| y_t - \sum_{i=0}^{D} h_i \cdot \mu(u_{t-i}) - \sum_{i=D+1}^{M} h_i \cdot \mu(\hat{u}_{t-i}^{(f)}) \right|^2
\]  

(12)

We also define the forward path metric associated with the forward trellis path till time \( t \) and transition \( s_t^{(f)} \rightarrow s_{t+1}^{(f)} \) as, \( \Upsilon_t^{(f)}(s_t^{(f)}, s_{t+1}^{(f)}) = A_t(s_t^{(f)}) + \Gamma_t^{(f)}(s_t^{(Ext)}) \). Path history bits \( \pi_{t+1}(s_{t+1}^{(f)}) \) for the next reduced state \( s_{t+1}^{(f)} \) are obtained by identifying the transition \( s_t^{(f)} \rightarrow s_{t+1}^{(f)} \) that maximizes \( \Upsilon_t^{(f)}(s_t^{(f)}, s_{t+1}^{(f)}) \) and a corresponding decision \( \hat{u}_{t-D}^{(f)} \), which is merged with path history bits inherited from \( \pi_t(s_t^{(f)}) \), similar to [4]. To define the reduced state of depth \( D \) in backward pass, we divide the full state (1) into backward state \( s_t^{(b)} \) and backward path history \( \pi_t(s_t^{(b)}) \) given by (14) and (15) respectively.

\[
s_t = [s_t^{(b)} : \pi_t(s_t^{(b)})]
\]

(13)

\[
s_t^{(b)} = [u_{t-M} u_{t-M+1} \cdots u_{t-M+D-1}]
\]

(14)

\[
\pi_t(s_t^{(b)}) = [u_{t-M+D} u_{t-M+D+1} \cdots \hat{u}_{t-1}^{(b)}]
\]

(15)

Using reduced backward state \( s_t^{(b)} \), delayed input \( u_{t-M+D} \) and backward path history \( \pi_{t+1}(s_{t+1}^{(b)}) = [\hat{u}_{t-M+D+1}^{(b)} \cdots \hat{u}_{t}^{(b)}] \) associated with state \( s_{t+1}^{(b)} = [u_{t-M+1} \cdots u_{t-M+D}] \), we can form the hypothetical extended state \( s_t^{(Ext)} \) at time \( t \) as \( s_t^{(Ext)} = [s_t^{(b)} : u_{t-M+D} : \pi_{t+1}(s_{t+1}^{(b)})] = [u_{t-M} \cdots u_{t-M+D}, \hat{u}_{t-M+D+1}^{(b)} \cdots \hat{u}_{t-1}^{(b)}, \hat{u}_{t}^{(b)}] \). The backward log-branch transition probability used in backward recursion can be calculated as,

\[
\hat{\Gamma}_t^{(b)}(s_t^{(Ext)}) = \sum_{i=0}^{N-1} u_{t-M+D,i} L_{U_{t-M+D,i}} - \frac{1}{N_0} \left| y_t - \sum_{i=0}^{M-D-1} h_i \mu(\hat{u}_{t-i}^{(b)}) - \sum_{i=M-D}^{M} h_i \mu(u_{t-i}) \right|^2.
\]  

(16)

Note that because of different reduced state definitions, it is possible to calculate different log-branch transition probability in forward and backward pass. Similar to the forward path metric we define the backward path metric \( \Upsilon_t^{(b)}(s_t^{(b)}, s_{t+1}^{(b)}) \), associated with the backward trellis path till time \( (t + 1) \) and transition \( s_t^{(b)} \leftarrow s_{t+1}^{(b)} \) as,

\[
\Upsilon_t^{(b)}(s_t^{(b)}, s_{t+1}^{(b)}) = B_{t+1}(s_{t+1}^{(b)}) + \hat{\Gamma}_t^{(b)}(s_t^{(Ext)})
\]  

(17)

Now backward path history bits are calculated during backward pass. Each state \( s_t^{(b)} \) can be reached from time \( t + 1 \) through \( 2^N \) branches. For each of these branches we can calculate the backward path metric associated with it using (17). The state \( s_t^{(b)} \) inherits path history bits from \( \pi_{t+1}(s_{t+1}^{(b)}) \), where \( s_t^{(b)} \leftarrow s_{t+1}^{(b)} \) is the transition maximizing \( \Upsilon_t^{(b)}(s_t^{(b)}, s_{t+1}^{(b)}) \).
Reduced forward trellis and reduced backward trellis can be mapped onto a full state trellis using (9) and (13). Since the two reduced sub-trellises formed in MFS equalizer are independent of each other, corresponding active full states might not match. There are four different cases for the states as shown in Fig. 2. We list them as,

**Case 0**  Neither sub-trellis passes through \( s_t \).

**Case 1**  Both sub-trellises passes through \( s_t \).

**Case 2**  Only forward sub-trellis passes through \( s_t \).

**Case 3**  Only backward sub-trellis passes through \( s_t \).

If we ignore these inconsistencies, then we evaluate the (5) only over those states, where forward state is consistent with backward full state, i.e. only over case 1. In order to utilize all available soft information provided by independent forward and backward sub-trellises, we can account for these inconsistencies by using different metrics from table (1) in evaluating (5). Here \( A_{t,\text{min}} \) and \( B_{t,\text{min}} \) are the minimum of \( A_t \) and \( B_t \) respectively over all states \( s_t \) at time \( t \). By taking the minimum value we do not introduce bias to the selected state over all other states.

**Table 1: Accounting for state inconsistencies**

<table>
<thead>
<tr>
<th>Case</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>( A_{t,\text{min}}(s_t) + B_{t,\text{min}}(s_t) )</td>
</tr>
<tr>
<td>1</td>
<td>( A_t(s_t) + B_t(s_t) )</td>
</tr>
<tr>
<td>2</td>
<td>( A_t(s_t) + B_{t,\text{min}}(s_t) )</td>
</tr>
<tr>
<td>3</td>
<td>( A_{t,\text{min}}(s_t) + B_t(s_t) )</td>
</tr>
</tbody>
</table>

3.1 Effect of delayed APP calculation

Effect of delayed APP calculation was first shown in [6]. We can also use (18) to calculate the APPs of constituent bits of \( u_t \), where \( T = t + 1 + \Delta \) and \( 0 \leq \Delta < M - 1 \). For \( \Delta > 0 \), we delay the APP calculation of the equalized bits by time \( t = \Delta \).

\[
A(u_{t,i}) = \ln \sum_{\forall s_T: u_{t,i}=1} \exp\{A_T(s_T) + B_T(s_T)\} - \ln \sum_{\forall s_T: u_{t,i}=0} \exp\{A_T(s_T) + B_T(s_T)\}
\]  

(18)

Fig. 3 shows the MFS equalizer performance for an EDGE system for different values of decision delay \( \Delta \) with \( M = 3 \) and \( D = 2 \). For the simulations we consider three

\[1\] In fig. 2 we can notice some of the illegal full state transitions which occur due to incorrect decisions made in DFSE trellis.
different channels of memory 3 (i.e. #of states= 8^3). First channel is the worst case ISI channel, \( h_{\text{worst}} = [0.38, 0.60, 0.60, 0.38] \), taken from [10]. Second channel is the minimum phase channel, \( h_{\text{min}} = [0.60, 0.60, 0.38, 0.38] \), where most of the channel energy is concentrated in the initial taps, and the third channel is the maximum phase channel, \( h_{\text{max}} = [0.38, 0.38, 0.60, 0.60] \), where most of the channel energy is concentrated in the later channel taps. For the minimum phase channel, we gain about 2-2.5 dB with \( D = 2 \) as shown in Fig 3(a). Also in case of the maximum phase channel, we gain about 2-2.5 dB with \( D = 0 \) as shown in Fig 3(b). In general we can say that, for a minimum phase channel \( D = M - 1 \) is optimum, which is the maximum possible delay value and for a maximum phase channel \( D = 0 \) is optimum, which is the minimum possible delay value.

Effect of delayed APP calculation on the BER performance can be explained with the help of Fig. 4. Here we consider minimum phase channel with \( M = 2 \) and \( D = 1 \). The forward transition \( s_{t+1} \rightarrow s_{t+2} \), at time \((t+1)\), involving full states, \( s_{t+1} = [u_{t-1}, u_t] \) and \( s_{t+2} = [u_t, u_{t+1}] \) is shown in Fig. 4. Since symbol \( u_t \) is present in both the state definitions, we can evaluate the APPs of \( u_t \) at time \((t+1)\) or at time \((t+2)\) by marginalizing (18) over the remaining bits. For \( \Delta = 0 \), we evaluate the APPs of \( u_t \) at time \((t+1)\) and for \( \Delta = 1 \) we evaluate the APPs of \( u_t \) at time \((t+2)\). When channel is of minimum phase, most of the channel energy is concentrated in the initial taps of the channel. This makes the forward pass more reliable than backward pass. Fig. 4, shows the effect of \( u_t \) on received samples. For three tap channel shown in Fig. 4, \( u_t \) affects three samples, \( y_t, y_{t+1} \) and \( y_{t+2} \). When \( \Delta = 0 \), we use \( A_{t+1} \) and \( B_{t+1} \) for the APP calculation of \( u_t \). In this case symbol \( u_t \) affects more reliable forward probabilities \( A_{t+1} \) through channel tap \( h_0 \) and it affects less reliable backward probabilities \( B_{t+1} \) through channel taps \( h_1 \) and \( h_2 \). When \( \Delta = 1 \), we use \( A_{t+2} \) and \( B_{t+2} \) for the APP calculation of \( u_t \). In this case symbol
\( \mathbf{u}_t \) affects \( A_{t+2} \) thorough channel taps \( h_0 \) and \( h_1 \) and it affects \( B_{t+2} \) thorough channel tap \( h_2 \). Thus for minimum phase channel when \( \Delta = 1 \), symbol \( \mathbf{u}_t \) affects more reliable forward probabilities through more number of higher energy channel taps as compared to the case when \( \Delta = 0 \). In general we can say that for minimum phase channel, maximum possible delay is the optimum delay. Similarly for maximum phase channel, minimum possible delay is the optimum delay. Note that the effect of delayed APP calculation is present only when the reliability of forward and backward probabilities is different. So BER of full state equalizer and DFSE-MAP equalizer are not affected by delayed APP calculation.

4 Matched Reduced State (MRS) Equalizer

In matched reduced state equalizer a-posteriori LLRs are calculated after matching the reduced states of the forward and backward sub-trellises. We write the reduced forward state \( s_t^{(f)} \) (10) at time \( t \) and reduced backward state \( s_{t+M-D}^{(b)} \) (14) at time \( (t + M - D) \) in the following equations.

\[
\begin{align*}
  s_t^{(f)} &= [u_{t-D} \ u_{t-D+1} \ \ldots \ u_{t-1}] \quad (19) \\
  s_{t+M-D}^{(b)} &= [u_{t-D} \ u_{t-D+1} \ \ldots \ u_{t-1}] \quad (20)
\end{align*}
\]

From (19) and (20), we see that \( s_t^{(f)} = s_{t+M-D}^{(b)} = s^{(r)} \). So we can calculate the a-posteriori LLRs of \( \mathbf{u}_t \) by using (21).

\[
\begin{align*}
  \Lambda(\mathbf{u}_{t,i}) &= \ln \sum_{u_{t,i} = 1}^{s^{(r)}} \exp\{A_{t+1}(s^{(r)}) + B_{t+1+M-D}(s^{(r)})\} \\
  &- \ln \sum_{u_{t,i} = 0}^{s^{(r)}} \exp\{A_{t+1}(s^{(r)}) + B_{t+1+M-D}(s^{(r)})\} \quad (21)
\end{align*}
\]

Equation (21) shows that, in order to calculate the a posteriori LLRs of \( \mathbf{u}_t \) we need \( A(t+1) \) and \( B(t+1+M-D) \). Here we ignore \( A \) and \( B \) metric between time \( (t+1) \) to time \( (t+1+M-D) \). So this method is sub-optimal method where we calculate the a posteriori probability of \( \mathbf{u}_t \), given all received samples \( y_0^{T-1} \) except \( y_{t+1}^{t+M-D} \). This loss is reduced as depth of reduced state \( D \) approaches channel memory \( M \).

4.1 Effect of Delayed APP Calculation

Similar to MFS equalizer, we can calculate the APPs of constituent bits of \( \mathbf{u}_t \) by using (22),

\[
\begin{align*}
  \Lambda(\mathbf{u}_{t,i}) &= \ln \sum_{u_{t,i} = 1}^{s^{(r)}} \exp\{A_{t+1+\Delta}(s^{(r)}) + B_{t+1+M-D+\Delta}(s^{(r)})\} \\
  &- \ln \sum_{u_{t,i} = 0}^{s^{(r)}} \exp\{A_{t+1+\Delta}(s^{(r)}) + B_{t+1+M-D+\Delta}(s^{(r)})\} \quad (22)
\end{align*}
\]

where \( s_{t+1+\Delta}^{(f)} = s_{t+1+M-D+\Delta}^{(b)} = s^{(r)} \) and \( 0 \leq \Delta < D - 1 \). Fig. 5, shows the MRS equalizer performance for different values of decision delay \( \Delta \) with \( M = 3 \) and \( D = 2 \). For the
minimum phase channel we gain about 1 dB with $\Delta = 1$, as shown in Fig 5(a). Also in case of the maximum phase channel we gain about 1 dB with $\Delta = 0$, as shown in Fig 5(b).

In general we can say that, for a minimum phase channel the maximum possible delta is optimum and for a maximum phase channel the minimum possible delta is optimum.

## 5 Simulation Results

Fig. 6, shows the turbo equalization performance of an EDGE system (MCS-5 down-link scheme [11–13]) for the worst case ISI channel $h_{\text{worst}}$. Only 3 turbo iterations are done considering the simulation time required. We observe that for # states = 8 and 64, we gain about 2.5 dB and 2.0 dB resp. with the MRS equalizer as compared to the DFSE-MAP equalizer. The MFS equalizer with $\Delta = 1$ and # states = 64 performs 0.5 dB better as compared to DFSE-MAP equalizer with # states = 64. Turbo equalization does not work for the MFS equalizer when $\Delta$ is such that we calculate the APPs of a symbol located at either forward path history position or at backward path history position. So for $M = 3$, turbo equalization with the MFS equalizer works only for the case ($D = 2$ and $\Delta = 1$) shown in Fig. 6. For all other cases ($D = 2$ with $\Delta = 0/2$, etc.).
and $D = 1$ with $\Delta = 0/1/2$) MFS equalizer outputs high values of LLRs (makes early hard decisions) which has a detrimental effect in an iterative algorithm. Fig. 7, compares the EXIT [14] curves of different reduced state equalizers for worst case ISI channel. Although the extrinsic information given by the reduced state equalizer is not the true extrinsic information, we use exit chart to get some insight into the turbo equalization using reduced state equalizers. We see that all the exit curves for reduced state equalizers lie below the exit curve of full state equalizer. Also none of the reduced state equalizers except MFS equalizer can achieve the ultimate output mutual information when complete input mutual information is available. This indicates that even for infinite iterations, DFSE-MAP and MRS equalizer can not achieve the same performance as full state equalizer. This observation is in accordance with the fact that DFSE based algorithms can not achieve the ML performance even when perfect decision feedback is available. We observe that for DFSE-MAP equalizer, loss in the ultimate output mutual information is exactly proportional to the loss in the channel energy due to shortening of the channel impulse response. For MRS equalizer this loss is even more as we do not account all the received samples for APP calculations. When $D = M - 1$ for MFS equalizer then all the available channel energy can be used for APP calculation. This results in no loss for MFS equalizer when perfect decision feedback is available.

6 Conclusion

For DFSE-MAP equalizer, backward pass is always biased by the path history decisions made during forward pass. So if decisions made during forward pass are unreliable then decisions made during backward pass also become unreliable. This results in the loss in BER performance for DFSE-MAP equalizer. In this paper we presented two novel equalization algorithms (MFS and MRS) which form independent forward and backward paths. This fact not only improves the BER performance but also makes these equalizers more robust than the DFSE-MAP equalizer. Performance of these equalizers can be optimized by using delayed APP calculation. For turbo equalization of an EDGE system, these equalizers perform much better than DFSE-MAP equalizer with the considered worst case ISI channel. With the exit-chart we have shown that MFS equalizer can achieve the ultimate output mutual information in an iterative scheme, which most of the DFSE based equalizers fail to achieve because of inherent loss in channel energy due
to hard decisions.

References


