Nonlinear Turbo Equalization Using Context Trees

Nargiz Kalantarova*, Kyeongyeon Kim†, Suleyman S. Kozat* and Andrew C. Singer†
*EEE Department, Koc University, Istanbul, Turkey
†Electrical and Computer Engineering Dept., Univ. of Illinois at Urbana-Champaign, Urbana, IL, 61801
Email: {nkalantarova,skozat}@ku.edu.tr acsinger@illinois.edu, adrianakky@gmail.com

***Abstract*—In this paper, we study adaptive nonlinear turbo equalization to model the nonlinear dependency of a linear minimum mean square error (MMSE) equalizer on soft information from the decoder. To accomplish this, we introduce piecewise linear models based on context trees that can adaptively choose both the partition regions as well as the equalizer coefficients in each region independently, with the computational complexity of a single adaptive linear equalizer. This approach is guaranteed to asymptotically achieve the performance of the best piecewise linear equalizer that can choose both its regions as well as its filter parameters based on observing the whole data sequence in advance.**

***Index Terms*—Turbo equalization, nonlinear equalization, context trees, decision feedback.**

I. Introduction

In this paper, we investigate adaptive nonlinear models for turbo equalization [1] in order to model the nonlinear dependency of a linear minimum mean square error (MSE) equalizer with priors [2] on soft information from the decoder [3]. To accomplish this we introduce piecewise linear models based on context trees [4] that partition the space of variances calculated from the soft information from the decoder. The algorithm we introduce can adaptively choose the regions of a partition as well as the linear equalizer coefficients in each region with a computational complexity of a single adaptive linear equalizer [5]. The resulting nonlinear equalizer asymptotically (and uniformly) achieves the performance of the best piecewise linear equalizer that can choose both the regions of an arbitary partition of the space spanned by the soft-information as well as the filter parameters in these regions based on the underlying signal.

The performance of equalization and decoding can be significantly improved by interaction between the equalizer and the channel decoder structure through turbo equalization [1], [6], [7], which takes advantage of the concatenated code structure of an error-correcting code (ECC) and interleaver implemented at the transmitter and the convolutional structure of the communication channel [5]. In the turbo equalization framework, linear equalizers have been investigated due to their low computational complexity as compared to trellis-based approaches [8]. Much research has focused on linear MMSE equalization [3], [9]. However, linear MMSE turbo equalizers still have higher computational complexity (and require an estimate of the channel) than adaptive linear turbo equalizers that use recursive least squares (RLS) or least mean squares (LMS), to train their coefficients [3], [5], [10]. Such computational load becomes especially important in applications with long channel impulse responses [11], [12].

In the context of linear equalization, several algorithms based on different adaptation methods have been investigated, e.g., [8], [13], [14]. Even though such adaptive linear turbo equalizers may converge to their optimal (e.g., MMSE) solution, for the stationarized problem, where the time-varying soft information is replaced by its time average [3], [15]. The performance loss of these algorithms is due to their implicit use of the log likelihood ratio (LLR) information from the decoder as stationary soft decision data [16], whereas a linear MMSE turbo equalizer considers LLR information as nonstationary *a priori* statistics over the transmitted symbols [3]. Hence, the filter coefficients of the linear MMSE turbo equalizer are time varying even for a time invariant channel since the *a priori* LLRs at the output of the decoder are time varying. However, the filter coefficients of an adaptive linear turbo equalizer, e.g., trained using the LMS update, converge to one time-invariant steady state filter [16].

In this paper, we propose a novel adaptive turbo equalization framework where the nonlinear dependence of the linear MMSE turbo equalization on the soft information from the decoder is modeled using piecewise linear models by partitioning the space of variances calculated from the soft information into disjoint regions. By this approach, we view the time-varying nature of the MMSE-optimal filter as a time-invariant, but spatially-varying filter, over space of *a priori* LLRs from the decoder. We use context trees to adaptively learn both the optimal partitioning of the space as well as the filter coefficients for each region.

Context trees and context tree weighting are extensively used in data compression [4], coding and data prediction [17]–[19]. In the context of source coding and universal probability assignment, the context tree weighting method is mainly used to calculate a weighted mixture of probabilities generated by the piecewise Markov models represented on the tree [4]. In the context of nonlinear prediction, context trees are used to represent piecewise linear models by partitioning the space of past regressors [17], [19], specifically for labeling the past observations. Note that although we use the notion of context trees for nonlinear modeling as in [4], [18]–[21], our results and usage of context trees differ from [4], [18]–[20] in a number of important ways. We emphasize that the context trees used here are specifically used to represent the nonlinear dependency of equalizer coefficients on the soft information.

II. System Description

In this paper, all vectors are column vectors and represented by boldface lowercase letters. Matrices are represented by boldface uppercase letters. Given a vector $\boldsymbol{x}$, $\|\boldsymbol{x}\| = \boldsymbol{x}^H\boldsymbol{x}$ is the l_2-norm, where $\boldsymbol{x}^H$ is the conjugate transpose, $\boldsymbol{x}^T$ is the ordinary transpose and $\boldsymbol{x}^*$ is the complex conjugate. For a random variable x (or a vector $\boldsymbol{x}$), $E[x] = \bar{x}$ (or $E[\boldsymbol{x}] = \bar{\boldsymbol{x}}$) is the expectation. For a vector $\boldsymbol{x}$ (or $\boldsymbol{x}(t)$), $\mathrm{diag}(\boldsymbol{x})$ (or $\mathrm{diag}(\boldsymbol{x}(t))$) is a diagonal matrix constructed from the entries of $\boldsymbol{x}$ (or $\boldsymbol{x}(t)$) and $x(i)$ (or $x(t,i)$) is the ith entry of the vector. For a square matrix $\boldsymbol{M}$, $\mathrm{tr}(\boldsymbol{M})$ is the trace. The sequences are represented using curly brackets, e.g., $\{x(t)\}$.

The block diagram of the system we consider with a linear turbo equalizer is shown in Fig. 1. The information bits $\{b(t)\}$ are first encoded using an error correcting code (ECC) and then the encoded bits are interleaved to generate $\{c(t)\}$. The interleaved code bits $\{c(t)\}$ are transmitted after symbol mapping through a baseband discrete-time channel with a finite-length impulse response $\{h(t)\}$, $t = 0, 1, \ldots, M-1$, represented by $\boldsymbol{h} \triangleq [h(M-1), \ldots, h(0)]^T$. The communication channel $\boldsymbol{h}$ is unknown. The transmitted signal $\{x(t)\}$ is assumed to be uncorrelated due to the interleaver. The received signal $y(t)$ is given by

$$y(t) \triangleq \sum_{k=0}^{M-1} h(k)x(t-k) + n(t),$$

where $\{n(t)\}$ is the additive complex white Gaussian noise with zero mean and circular symmetric variance σ_n^2. If a linear equalizer is used to combat the ISI, then the estimate of the desired data $x(t)$ using the received data $y(t)$ is given by

$$\hat{x}(t) = \boldsymbol{w}^T(t)[\boldsymbol{y}(t) - \bar{\boldsymbol{y}}(t)] + \bar{x}(t),$$

where $\boldsymbol{w}(t) = [w(t, N_2), \ldots, w(t, -N_1)]^T$ is length $N = N_1 + N_2 + 1$ linear equalizer, $\boldsymbol{y}(t) \triangleq [y(t-N_2), \ldots, y(t+N_1)]^T$. Since the received data vector $\boldsymbol{y}(t)$ is given by $\boldsymbol{y}(t) = \boldsymbol{H}\boldsymbol{x}(t) + \boldsymbol{n}(t)$, where $\boldsymbol{x}(t) \triangleq [x(t-M-N_2+1), \ldots, x(t+N_1)]^T$ and $\boldsymbol{H} \in \mathbb{C}^{N\times(N+M-1)}$ is the convolution matrix corresponding to $\boldsymbol{h} = [h(M-1), \ldots, h(0)]^T$, the estimate of $x(t)$ can be written as

$$\hat{x}(t) = \boldsymbol{w}^T(t)[\boldsymbol{y}(t) - \boldsymbol{H}\bar{\boldsymbol{x}}(t)] + \bar{x}(t), \tag{1}$$

given that the mean of the transmitted data is known.

However, in turbo equalization, instead of only using an equalizer, the equalizer and decoder are considered as the inner decoder and outer decoder, respectively, and an iterative decoding scheme is used at the receiver of Fig. 1. The equalizer computes *a posteriori* information using the received signal, transmitted signal estimate, channel convolution matrix (if known) and *a priori* probability of the transmitted data. After subtracting the *a priori* information, LLR_a^E, and de-interleaving the extrinsic information LLR_e^E, a soft input soft output (SISO) channel decoder computes the extrinsic information LLR_e^D on coded bits, which are fed back to the linear equalizer as *a priori* information LLR_a^E after interleaving.

As the linear equalizer, if one uses the linear MMSE equalizer for $\boldsymbol{w}(t)$, the mean and the variance of $x(t)$ are required to calculate $\boldsymbol{w}(t)$ and $\hat{x}(t)$. These quantities are

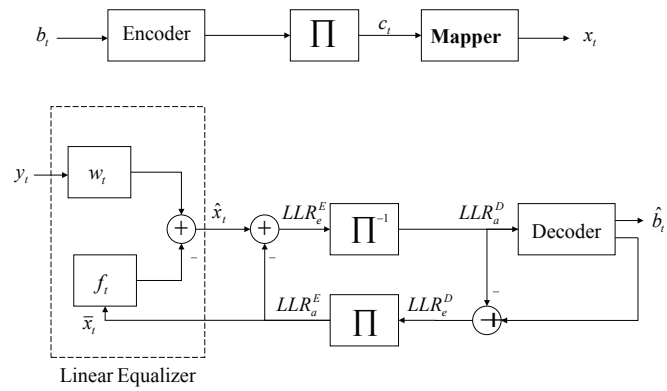

Fig. 1: Block diagram for a bit interleaved coded modulation transmitter and receiver with a linear turbo equalizer.

computed using the *a priori* information from the decoder as $\bar{x}(t) \triangleq E[x(t) : \{\mathrm{LLR}_a^E(t)\}]$ and $q(t) \triangleq E[x^2(t) : \{\mathrm{LLR}_a^E(t)\}] - \bar{x}^2(t)$. As an example, for BPSK signaling, the mean and variance are given as $\bar{x}(t) = \tanh(\mathrm{LLR}_a^E(t)/2)$ and $q(t) = 1 - |\bar{x}(t)|^2$. However, to remove dependency of $\hat{x}(t)$ to $\mathrm{LLR}_a^E(t)$ due to using $\bar{x}(t)$ and $q(t)$ in (1), one can set $\mathrm{LLR}_a^E(t) = 0$ while computing $\hat{x}(t)$, yielding $\bar{x}(t) = 0$ and $q(t) = 1$ [3]. Then, the linear MMSE equalizer is given by

$$\boldsymbol{w}(t) = [\boldsymbol{v}^H(\sigma_n^2\boldsymbol{I} + \boldsymbol{H}_r\boldsymbol{Q}(t)\boldsymbol{H}_r^H + \boldsymbol{v}\boldsymbol{v}^H)^{-1}]^T, \tag{2}$$

where $\boldsymbol{Q}(t) \in \mathbb{C}^{(N+M-2)\times(N+M-2)}$ is a diagonal matrix (due to the uncorrelatedness assumption on $x(t)$) with diagonal entries $\boldsymbol{Q}(t) = \mathrm{diag}(\boldsymbol{q}(t))$, $\boldsymbol{q}(t) \triangleq [q(t-M-N_2+1), \ldots, q(t-1), q(t+1), \ldots, q(t+N_1)]^T$, $\boldsymbol{v} \in \mathbb{C}^N$ is the $(M+N_2)$th column of $\boldsymbol{H}$, $\boldsymbol{H}_r$ is the reduced form of $\boldsymbol{H}$ where the $(M+N_2)$th column is removed. We emphasize that the linear MMSE equalizer is a nonlinear function of the vector of variances, i.e.,

$$\boldsymbol{w} = \mathcal{W}(\boldsymbol{q}) \triangleq [\boldsymbol{v}^H(\sigma_n^2\boldsymbol{I} + \boldsymbol{H}_r\,\mathrm{diag}(\boldsymbol{q})\,\boldsymbol{H}_r^H + \boldsymbol{v}\boldsymbol{v}^H)^{-1}]^T, \tag{3}$$

$\mathcal{W}(\cdot) : \mathbb{C}^{N+M-2} \rightarrow \mathbb{C}^N$ and time variation in (2) is due to the time variation in the vector of variances $\boldsymbol{q}$, i.e., $\boldsymbol{w}(t) = \mathcal{W}(\boldsymbol{q}(t))$ (assuming $\boldsymbol{h}$ is time-invariant). Hence the linear MMSE equalizer in (1) yields

$$\begin{aligned}\hat{x}(t) &= \boldsymbol{w}^T(t)[\boldsymbol{y}(t) - \boldsymbol{H}\bar{\boldsymbol{x}}(t)] \\ &= \boldsymbol{w}^T(t)\boldsymbol{y}(t) - \boldsymbol{f}^T(t)\bar{\boldsymbol{x}}(t),\end{aligned} \tag{4}$$

where $\boldsymbol{f}(t) \triangleq \boldsymbol{H}^T\boldsymbol{w}(t)$. In this sense the linear MMSE equalizer can be decomposed into a feedforward filter $\boldsymbol{w}(t)$ processing $\boldsymbol{y}(t)$ and a feedback filter $\boldsymbol{f}(t)$ processing $\bar{\boldsymbol{x}}(t)$. Both the linear MMSE feedforward and feedback filters are highly nonlinear functions of $\boldsymbol{q}$, i.e., $\boldsymbol{w} = \mathcal{W}(\boldsymbol{q})$ and $\boldsymbol{f} = \mathcal{F}(\boldsymbol{q}) \triangleq \boldsymbol{H}^T\mathcal{W}(\boldsymbol{q})$, $\mathcal{F}(\cdot) : \mathbb{C}^{N+M-2} \rightarrow \mathbb{C}^{N+M-1}$. To learn the corresponding feedforward and feedback filters that are highly nonlinear functions of $\boldsymbol{q}$, we use piecewise linear models based on vector quantization and context trees in the next section. The space spanned by $\boldsymbol{q}$ is partitioned into disjoint regions and a separate linear model is trained for each region to approximate functions $\mathcal{W}(\cdot)$ and $\mathcal{F}(\cdot)$ using piecewise linear models.

Note that if the channel is not known or estimated, one can directly train the corresponding equalizers in (4) using adaptive algorithms such as in [8], [16] without channel estimation or partitioning as done in this paper. In this case, one directly applies the adaptive algorithms to feedforward and feedback filters using the received data $\{y(t)\}$ and the mean vector $\bar{\boldsymbol{x}}(t)$ as feedback without considering the soft decisions as *a priori* probabilities. Assuming stationarity of $\bar{\boldsymbol{x}}(t)$, such adaptive feedforward and feedback filters have MMSE solutions [16]

$$\boldsymbol{w} = [\boldsymbol{v}^H(\sigma_n^2\boldsymbol{I} + \boldsymbol{H}_r E[\boldsymbol{q}(t)\boldsymbol{q}^H(t)]\boldsymbol{H}_r^H + \boldsymbol{v}\boldsymbol{v}^H)^{-1}]^T, \quad (5)$$
$$\boldsymbol{f} = \boldsymbol{H}^T\boldsymbol{w}.$$

When PSK signaling is used such that $E[|x(t)|^2 : \{\mathrm{LLR}_a^E(t)\}] = 1$, the filter coefficient vector in (5) is equal to the coefficient vector of the MMSE equalizer in [3] with time averaged soft information, i.e., time average instead of an ensemble mean. Comparing (5) and (2), we observe that using the time averaged soft information does not degrade equalizer performance in the no *a priori* information, i.e., $\boldsymbol{Q}(t) = \boldsymbol{I}$ or perfect *a priori* information, i.e., $\boldsymbol{Q}(t) = \boldsymbol{0}$, cases. In addition, the performance degradation in moderate ISI channels is often small [3] when perfect channel knowledge is used. However, the performance gap increases in ISI channels that are more difficult to equalize, even in the high SNR region [16].

III. Nonlinear Turbo Equalization Using Piecewise Linear Models

We divide the space spanned by $\boldsymbol{q} \in [0,1]^{N+M-2}$ (assuming BPSK signaling for notational simplicity) into disjoint regions V_k, e.g., $[0,1]^{N+M-2} = \bigcup_{k=1}^{K} V_k$ for some K and train an independent linear equalizer in each region V_k to yield a final piecewise linear equalizer to approximate $\boldsymbol{w} = \mathcal{W}(\boldsymbol{q})$ and $\boldsymbol{f} = \mathcal{F}(\boldsymbol{q})$. The filters $\boldsymbol{w}_k(t)$ and $\boldsymbol{f}_k(t)$ are time varying since they are produced by adaptive algorithms sequentially learning the corresponding functions $\mathcal{W}(\cdot)$ and $\mathcal{F}(\cdot)$ in region V_k. Note that if K is large and the regions are dense such that $\mathcal{W}(\boldsymbol{q})$ (and $\mathcal{F}(\boldsymbol{q})$) can be considered constant in V_k, say equal to $\mathcal{W}(\boldsymbol{q}_k)$ for some $\boldsymbol{q}_k$ in region V_k, then if the adaptation method used in each region converges successfully, this yields $\boldsymbol{w}_k(t) \to \mathcal{W}(\boldsymbol{q}_k)$ and $\boldsymbol{f}_k(t) \to \mathcal{F}(\boldsymbol{q}_k)$ as $t \to \infty$. Hence, if these regions are dense and there is enough data to learn the corresponding models in each region, then this piecewise model can approximate any smoothly varying $\mathcal{W}(\boldsymbol{q})$ and $\mathcal{F}(\boldsymbol{q})$ [23].

We next introduce a binary context tree to partition $[0,1]^{N+M-2}$ into disjoint regions in order to construct a piecewise linear equalizer that can choose both the regions as well as the equalizer coefficients in these regions based on the equalization performance. On a context tree, starting from the root node, we have a left hand child and a right hand child. Each left hand child and right hand child have their own left hand and right hand children. This splitting yields a binary tree of depth D with a total of 2^D leaves at depth D. Including the leaves, the context tree has $2^{D+1}-1$ nodes. As an example, a context tree with $D = 2$ partitions $[0,1]^2$, i.e., $\boldsymbol{q} = [q(t-1), q(t+1)]^T \in [0,1]^2$, into 4 disjoint regions. Each one of these 4 disjoint regions is assigned to a leaf on

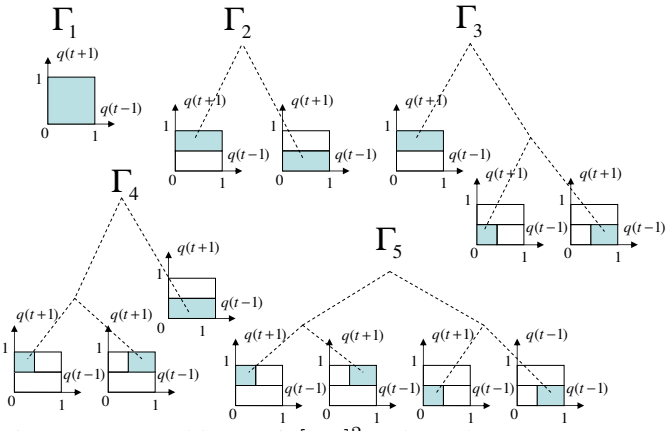

Fig. 2: All partitions of $[0,1]^2$ using binary context tree with $D = 2$. Given any partition, the union of the regions represented by the leaves of each partition is equal to $[0,1]^2$.

this binary tree. Then, recursively, each internal node on this tree represents a region, which is the union of the regions assigned to its children. For example, the left hand child node of the root node represents the region which is the union of regions assigned to the left-left hand child and the left-right hand child of the root node.

On a binary context tree of depth D, one can define a doubly exponential number, $m \approx 1.5^{2^D}$, of "complete" subtrees as in Fig. 2. A complete subtree is constructed from a subset of the nodes of the original tree, starting from the same root node. We call a subtree complete, if the union of the regions assigned to its leaves yields $[0,1]^{N+M-2}$. Note that a leaf of a subtree is either an internal node or a leaf node of the original tree. For example, for a subtree i, if the regions assigned to its leaves are labeled as $V_{1,i}, \ldots, V_{K_i,i}$ where K_i is the number of leaves of the subtree i, then $[0,1]^{N+M-2} = \bigcup_{k=1}^{K_i} V_{k,i}$. Each $V_{k,i}$ of the subtree corresponds to a node in the original tree.

With this definition, a complete subtree with the regions assigned to its leaves defines a complete "partition" of $[0,1]^{N+M-2}$. On a binary tree of depth $D = 2$, we have 5 different partitions of $[0,1]^2$ as shown in Fig. 2 labeled as $\Gamma_1, \ldots, \Gamma_5$. For each partition, since we have $\Gamma_i = \{V_{1,i}, \ldots, V_{K_i,i}\}$ and $\bigcup_{l=1}^{K_i} V_{k,i} = [0,1]^{N+M-2}$, each partition Γ_i can be used to construct a piecewise linear equalizer. Given such a context tree, one can define $m \approx (1.5)^{2^D}$ different partitions of the space spanned by $\boldsymbol{q}$ and construct a piecewise linear equalizer for each such partition. There exist a doubly exponential number of different partitionings for such piecewise linear models that one can choose from. For each such partition, one can train and use a piecewise linear model.

Although we can construct a doubly exponential number m of piecewise linear equalizers using the partitions defined on the context tree, note that all these piecewise linear equalizers are constructed using subsets of nodes $\rho \in \{1, \ldots, 2^{D+1}-1\}$. Hence, suppose we number each node on this context tree $\rho = 1, \ldots, 2^{D+1}-1$ and assign a linear equalizer to each node as $\hat{x}_\rho(t) = \boldsymbol{w}_\rho^T(t)\boldsymbol{y}(t) - \boldsymbol{f}_\rho^T(t)\bar{\boldsymbol{x}}(t)$. The linear models $\boldsymbol{w}_\rho$, $\boldsymbol{f}_\rho$ that are assigned to node ρ, train only on the data assigned to that node, i.e., if $\boldsymbol{q}(t)$ is in the region that is assigned to the node V_ρ then $\boldsymbol{w}_\rho$ and $\boldsymbol{f}_\rho$ are updated. Then, the piecewise linear equalizer corresponding to the partition Γ_i, say $\hat{x}_{\Gamma_i}(t)$,

is defined such that if $\boldsymbol{q}(t) \in V_{k,i}$ and ρ is the node that is assigned to $V_{k,i}$ then

$$\begin{aligned}\hat{x}_{\Gamma_i}(t) &= \hat{x}_\rho(t) \\ &= \boldsymbol{w}_\rho^T(t)\boldsymbol{y}(t) - \boldsymbol{f}_\rho^T(t)\bar{\boldsymbol{x}}(t). \end{aligned} \tag{6}$$

One of these partitions, with the given piecewise adaptive linear model $\hat{x}_{\Gamma_i}(t)$ achieves the minimal loss, e.g., the minimal accumulated squared error $\sum_{t=1}^{n}(x(t)-\hat{x}_{\Gamma_i}(t))^2$, for some n.

The context tree based equalization algorithm implicitly constructs all $\hat{x}_{\Gamma_i}(t)$, $i = 1, \ldots, m$, piecewise linear equalizers and acts as if it had actually run all these equalizers in parallel on the received data. At each time t, the final estimation $\hat{x}_{\text{ctw}}(t)$ is constructed as a weighted combination of all the outputs $\hat{x}_{\Gamma_i}(t)$ of these piecewise linear equalizers, where the combination weights are calculated proportional to the performance of each equalizer $\hat{x}_{\Gamma_i}(t)$ on the past data.The context tree algorithm keeps the track of these m piecewise linear models as well as their performance-based combination weights with computational complexity only linear in the depth of the context tree.

At each time $t > T$, $\hat{x}_{\text{ctw}}(t)$ constructs its nonlinear estimation of $x(t)$ as follows. We first find the regions (or nodes) to which $\boldsymbol{q}(t)$ belongs. Note that due to the tree structure, one needs only find the leaf node in which $\boldsymbol{q}(t)$ lies and collect all the parent nodes towards the root node. The final estimate $\hat{x}_{\text{ctw}}(t)$ is constructed as a weighted combination of the estimates generated in these nodes, i.e., $\hat{x}_\rho(t)$, $\rho \in \boldsymbol{l}$, where the weights are functions of the performance of the node estimates in previous samples. At each time t, $\hat{x}_{\text{ctw}}(t)$ requires $O(\ln(D))$ calculations to find the leaf to which $\boldsymbol{q}(t)$ belongs. Then, $D+1$ node estimations, $\hat{x}_\rho(t)$, $\rho \in \boldsymbol{l}$, are calculated and the filters at these nodes should be updated with $O(2N+M)$ computations. The final weighted combination is produced with $O(D)$ computations. Hence, at each time the context tree algorithm requires $O(D(2N+M))$ computational complexity. For this algorithm, we have the following result.

Theorem: *Let $\{x(t)\}$, $\{n(t)\}$ and $\{y(t)\}$ represent the transmitted, noise and received signals and $\{\boldsymbol{q}(t)\}$ represents the sequence of variances constructed using the a priori probabilities for each constellation point produced by the SISO decoder. Let $\hat{x}_\rho(t)$, $\rho = 1, \ldots, 2^{D+1}-1$, are estimates of $x(t)$ produced by the equalizers assigned to each node on the context tree. The algorithm $\hat{x}_{\text{ctw}}(t)$, when applied to $\{y(t)\}$, for all n achieves*

$$\sum_{t=1}^{n}\left(x(t)-\hat{x}_{\text{ctw}}(t)\right)^2 \leq \min_{\Gamma_i}\sum_{t=1}^{n}\left[x(t)-\hat{x}_{\Gamma_i}(t)\right]^2 + 2K_i - 1, \tag{7}$$

for all i, $i = 1, \ldots, m \approx (1.5)^{2^D}$, assuming perfect feedback in decision directed mode i.e., $Q(\hat{x}(t)) = x(t)$ when $t \geq T$, where $\hat{x}_{\Gamma_i}(t)$ is the equalizer constructed as

$$\hat{x}_{\Gamma_i}(t) = \hat{x}_\rho(t),$$

ρ is the node assigned to the volume in $\Gamma_i = \{V_{1,i}, \ldots, V_{K_i,i}\}$ such that $\boldsymbol{q}(t)$ belongs and K_i is the number of regions in Γ_i. If the estimation algorithms assigned to each node are selected as adaptive linear equalizers such as an RLS update based algorithm, (7) *yields*

$$\sum_{t=1}^{n}\left[x(t)-\hat{x}_{\text{ctw}}(t)\right]^2 \leq \min_i \Bigg\{ \min_{\substack{\boldsymbol{w}_{k,i}\in\mathbb{C}^N, \boldsymbol{f}_{k,i}\in\mathbb{C}^{N+M-1} \\ k=1,\ldots,K_i}} \tag{8}$$

$$\sum_{t=1}^{n}\left[x(t)-\boldsymbol{w}_{s_i(t),i}^T\boldsymbol{y}(t)-\boldsymbol{f}_{s_i(t),i}^T\bar{\boldsymbol{x}}(t)\right]^2 + O\big((2N+M)\ln(n)\big) + 2K_i - 1 \Bigg\}. \tag{9}$$

where $s_i(t)$ is an indicator variable for Γ_i such that if $\boldsymbol{q}(t) \in V_{k,i}$, then $s_i(t) = k$.

The proof of this theorem can be obtained through extending Theorem 2 of [20] and Theorem 1 of [27].

Remark: We observe from (7) that the context tree algorithm achieves the performance of the best sequential algorithm among a doubly exponential number of possible algorithms. Note that the bound in (7) holds uniformly for all i, however the bound is the largest for the finest partition corresponding to all leaves. We observe from (9) that the context tree algorithm also achieves the performance of even the best piecewise linear model, independently optimized in each region, for all i when the node estimators in each regions are adaptive algorithms that achieve the minimum least square-error.

IV. Conclusion

In this paper, we introduced an adaptive nonlinear turbo equalization algorithm using context trees in order to model the nonlinear dependency of the linear MMSE equalizer on the soft information generated from the decoder. We use the CTW algorithm to partition the space of variances, which are time dependent and generated from the soft information, and train a separate linear model for each region. We demonstrate that the algorithm introduced asymptotically achieves the performance of the best piecewise linear model defined on this context tree with a computational complexity only of the order of an ordinary linear equalizer.

References

[1] C. Douillard, M. Jezequel, and C. Berrou, "Iterative correction of intersymbol interference: Turbo equalization," *European Transactions on Telecommunications*, vol. 6, no. 5, pp. 507–511, September-October 1995.

[2] M. Tüchler, A. Singer, and R. Koetter, "Minimum mean squared error equalization using priors," *IEEE Transactions on Signal Processing*, vol. 50, no. 3, pp. 673–683, March 2002.

[3] M. Tuchler, R. Koetter, and A. C. Singer, "Turbo equalization: Principles and new results," *IEEE Transactions on Communications*, vol. 50, no. 5, pp. 754–767, May 2002.

[4] F. M. J. Willems, "Coding for a binary independent piecewise-identically-distributed source," *IEEE Trans. on Info. Theory*, vol. 42, pp. 2210–2217, 1996.

[5] J. Proakis, *Digital Communications*, New York: McGraw-Hill, 1995.

[6] J. Hagenauer, "The turbo principle: Tutorial introduction and state of the art," *IEEE Transactions on Communications*, vol. 44, pp. 1261–1271, Oct. 1996.

[7] C. Berrou and A. Blavieux, "Near optimum error correcting coding and decoding: Turbo codes," in *Allerton Conf.*, 1982, pp. 945–954.

[8] A. Glavieux, C. Laot, and J. Labat, "Turbo equalization over a frequency selective channel," in *Proc. Int. Symp. Turbo Codes, Brest, France*, 1997, pp. 96–102.

[9] S. Song, A.C. Singer, and K.-M. Sung, "Soft input channel estimation for turbo equalization," *IEEE Transactions on Signal Processing*, vol. 52, no. 10, pp. 2885 – 2894, October 2004.

[10] A. H. Sayed, *Fundamentals of Adaptive Filtering*, John Wiley and Sons, 2003.

[11] A.C. Singer, J.K. Nelson, and S.S. Kozat, "Signal processing for underwater acoustic communications," *IEEE Communications Magazine*, vol. 47, no. 1, pp. 90–96, January 2009.

[12] J.F. Sifferlen, H.C. Song, W.S. Hodgkiss, W.A. Kuperman, and M. Stevenson, "An iterative equalization and decoding approach for underwater acoustic communication," *IEEE J. Oceanic Engr.*, pp. 182–197, 2008.

[13] C. Laot, R. Le Bidan, and D D. Leroux, "Low-complexity MMSE turbo equalization: a possible solution for EDGE," *IEEE Transactions on Wireless Communications*, vol. 4, no. 3, pp. 965 – 974, May 2005.

[14] J.-W. Choi, R. Drost, A. Singer, and J. Preisig, "Iterative multichannel equalization and decoding for high frequency underwater acoustic communications," in *IEEE Sensor Array and Multichannel Sig. Proc. Workshop*, 2008, pp. 127–130.

[15] M. Tuchler, *Iterative Equalization Using Priors*, Ph.D. Thesis, University of Illinois at Urbana-Champaign, Urbana, 2000.

[16] R. Bidan, *Turbo-equalization for bandwith-efficient digital communications over frequency-selective channels*, Ph.D. Thesis, Institut TELECOM/TELECOM Bretagne, 2003.

[17] S. S Kozat and A. C. Singer, "Universal switching linear least squares prediction," *IEEE Transactions on Signal Processing*, vol. 56, pp. 189–204, Jan. 2008.

[18] D. P. Helmbold and R. E. Schapire, "Predicting nearly as well as the best pruning of a decision tree," *Machine Learning*, vol. 27, no. 1, pp. 51–68, 1997.

[19] E. Takimoto, A. Maruoka, and V. Vovk, "Predicting nearly as well as the best pruning of a decision tree through dyanamic programming scheme," *Theoretical Computer Science*, vol. 261, pp. 179–209, 2001.

[20] S. S Kozat, A. C. Singer, and G. Zeitler, "Universal piecewise linear prediction via context trees," *IEEE Transactions on Signal Processing*, vol. 55, pp. 3730–3745, 2007.

[21] Y. Yilmaz and S. S. Kozat, "Competitive randomized nonlinear prediction under additive noise," *IEEE Signal Processing Letters*, vol. 17, no. 4, pp. 335–339, 2010.

[22] S. Lee, A. C. Singer, and N. R. Shanbhag, "Linear turbo equalization analysis via BER transfer and EXIT charts," *IEEE Transactions on Signal Processing*, vol. 53, pp. 2883–2897, Aug. 2005.

[23] O. J. J. Michel, A. O. Hero, and A. E. Badel, "Tree structured nonlinear signal modeling and prediction," *IEEE Transactions on Signal Processing*, pp. 3027–3041, 1999.

[24] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*, The Springer International Series in Engineering and Computer Science, 1992.

[25] S. Haykin, *Adaptive Filter Theory*, Prentice Hall, 1996.

[26] K. Kyeongyeon, J. W. Choi, A. C. Singer, and K. Kim, "A new adaptive turbo equalizer with soft information classification," in *ICASSP, Dallas, TX*, 2010, pp. 3206–3209.

[27] A. C. Singer and M. Feder, "Universal linear prediction by model order weighting," *IEEE Transactions on Sig. Proc.*, vol. 47, no. 10, 1999.