Radial Basis Function-Assisted Turbo Equalization

Mong-Suan Yee, *Associate Member, IEEE*, Bee Leong Yeap, *Member, IEEE*, and Lajos Hanzo, *Senior Member, IEEE*

***Abstract*—This paper presents a turbo equalization (TEQ) scheme, which employs a radial basis function (RBF)-based equalizer instead of the conventional trellis-based equalizer of Douillard *et al.* Structural, computational complexity, and performance comparisons of the RBF-based and trellis-based TEQs are provided. The decision feedback-assisted RBF TEQ is capable of attaining a similar performance to the logarithmic maximum *a posteriori* scheme in the context of both binary phase-shift keying (BPSK) and quaternary phase-shift keying (QPSK) modulation, while achieving a factor 2.5 and 3 lower computational complexity, respectively. However, there is a 2.5-dB performance loss in the context of 16 quadrature amplitude modulation (QAM), which suffers more dramatically from the phenomenon of erroneous decision-feedback effects. A novel element of our design, in order to further reduce the computational complexity of the RBF TEQ, is that symbol equalizations are invoked at current iterations only if the decoded symbol has a high error probability. This techniques provides 37% and 54% computational complexity reduction compared to the full-complexity RBF TEQ for the BPSK RBF TEQ and 16QAM RBF TEQ, respectively, with little performance degradation, when communicating over dispersive Rayleigh fading channels.**

***Index Terms*—Decision-feedback equalizer (DFE), Jacobian logarithm, neural network, radial basis function (RBF), turbo coding, turbo equalization (TEQ).**

I. Background

PREVIOUS TURBO equalization (TEQ) research has invoked component equalizer implementations using the soft-output Viterbi algorithm (SOVA) [1], [2], the optimal maximum *a posteriori* (MAP) algorithm [2], [3], and linear filters [4]. In this paper, we employ a radial basis function (RBF)-assisted equalizer as the soft-in/soft-out (SISO) equalizer in the context of TEQ. In Sections I-A and I-B, we will introduce the RBF equalizer and the TEQ scheme, followed by the implementation details of the RBF TEQ in Section II. Section III is dedicated to the comparison of the RBF and MAP equalizers, while Section IV entails our discussions related to the complexity reduction issues of the RBF TEQ when using the Jacobian logarithm known from the field of turbo-channel coding. In Section V, the proposed scheme's performance is benchmarked against that of the optimal MAP TEQ scheme of [1]. Finally, in Section VI, we proposed a further computational complexity-reduction technique for the RBF TEQ, and in Section VII, we conclude our investigations.

Paper approved by M. Chiani, the Editor for Wireless Communication of the IEEE Communications Society. Manuscript received September 14, 2000; revised June 15, 2002. This paper was presented in part at the IEEE Vehicular Technology Conference, Tokyo, Japan, May 15–18, 2000.

The authors are with the Department of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, U.K. (e-mail:lh@ecs.soton.ac.uk).

Digital Object Identifier 10.1109/TCOMM.2003.810807

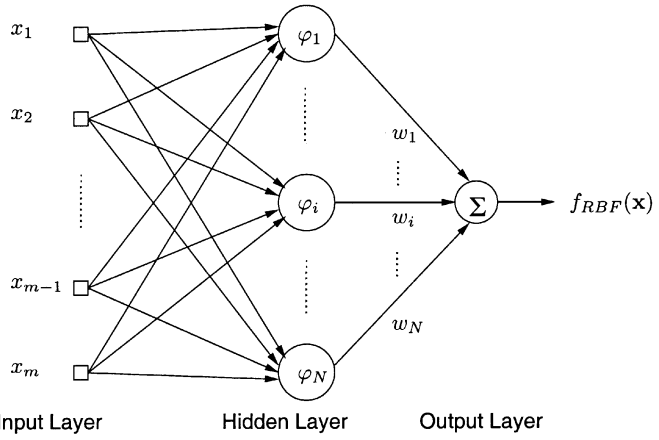


Fig. 1. Architecture of an RBF network.

A. Introduction to RBF-Based Equalization

Channel equalization can be viewed as a received phasor classification problem. However, due to channel effects, the phasors may become linearly nonseparable, and in these scenarios, the nonlinear structure of RBF-assisted equalizers can enhance the performance of conventional channel equalizers [5]. The RBF network [6] shown in Fig. 1 has an equivalent structure to the so-called optimal Bayesian equalization solution [7], and it can provide the conditional density function (CODF) of the transmitted symbols. Hence, the RBF network has been employed in various channel equalization applications [7], [8]. The overall response of the RBF network $f_{\mathrm{RBF}}(\mathbf{x})$ of Fig. 1 using Gaussian RBF φ_i can be formulated as

$$f_{\mathrm{RBF}}(\mathbf{x}) = \sum_{i=1}^{N} w_i \varphi_i(\mathbf{x})$$
$$\varphi_i(\mathbf{x}) = \exp\left(\frac{-\|\mathbf{x}-\mathbf{a}_i\|^2}{\sigma}\right) \tag{1}$$

where $\mathbf{a}_i, i = 1, \ldots, N$ represents the RBF centers, which have the same dimensionality as the input vector $\mathbf{x}$, $\|\cdot\|$ denotes the Euclidean norm, $\varphi_i(\cdot)$ is the RBF, σ is a positive constant defined as the width of the RBF, w_i is the weight of the RBF, and N is the number of hidden nodes in the RBF network [9]. The architecture of the RBF network-based equalizer designed for an $\mathcal{M}$-level modulation scheme is shown in Fig. 2. The optimal Bayesian decision solution of the mth order, $\mathcal{M}$-ary RBF equalizer receiving the signal vector $\mathbf{v}_k$ and optimized in order to achieve the minimum error probability of the detected symbol $\tilde{I}_{k-\tau}$ delayed by τ is as follows [10], [11]:

$$\tilde{I}_{k-\tau} = \mathcal{J}_i^*, \quad \text{if } f_{\mathrm{RBF}}^*(\mathbf{v}_k) = \max\left\{f_{\mathrm{RBF}}^i(\mathbf{v}_k), 1 \le i \le \mathcal{M}\right\} \tag{2}$$

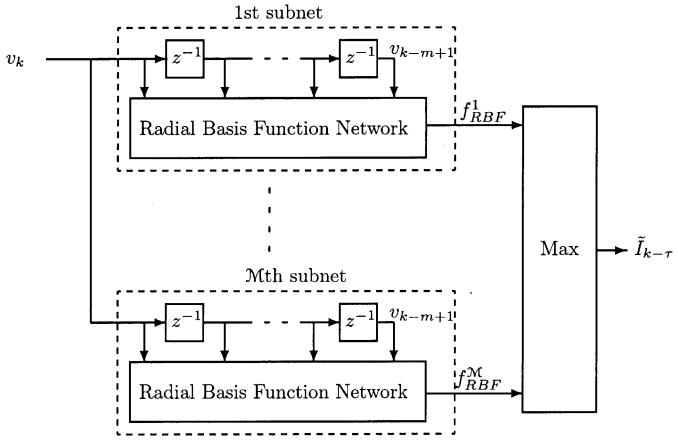


Fig. 2. RBF equalizer for an $\mathcal{M}$-ary scheme.

where $\mathcal{J}$ is the ith $\mathcal{M}$-ary symbol, and $f^i_{\mathrm{RBF}}(\mathbf{v}_k)$ is the decision variable provided by the ith sub-RBF network of Fig. 2 based on the CODF

$$f^i_{\mathrm{RBF}}(\mathbf{v}_k) = P(\mathbf{v}_k | I_{k-\tau} = \mathcal{J}_i) \cdot P(I_{k-\tau} = \mathcal{J}_i) \tag{3}$$

$$= \sum_{j=1}^{n_s^i} p_j^i p(\mathbf{v}_k - \mathbf{r}_j^i), \qquad 1 \le i \le \mathcal{M}. \tag{4}$$

The quantities p_j^i, $i = 1, \ldots, \mathcal{M}$, $j = 1, \ldots, n_s^i$ denote the *a priori* probability of occurance for each possible channel output state $\mathbf{r}_j^i$ associated with the transmitted $\mathcal{M}$-ary symbol $\mathcal{J}_i$, $i = 1, \ldots, \mathcal{M}$, and $p(\cdot)$ is the probability density function (PDF) of the additive noise imposed by the channel. The channel output state is described by the noise-free channel output vector observed in the RBF equalizer's m-dimensional input space. We will briefly describe this concept.

The symbol-spaced channel output can be defined by

$$v_k = \sum_{n=0}^{L} f_n I_{k-n} + \eta_k = \tilde{v}_k + \eta_k, \qquad -\infty \le k \le \infty \tag{5}$$

where $\{\eta_k\}$ is the additive white Gaussian noise (AWGN) sequence with variance σ_η^2, $\{f_n\}$, $n = 0, 1, \ldots, L$ is the channel impulse response (CIR), $\{I_k\}$ is the channel input sequence, and $\{\tilde{v}_k\}$ is the noise-free channel output. The channel output observed by the linear mth-order equalizer can be written in vector form as $\mathbf{v}_k = [v_k \; v_{k-1} \; \ldots \; v_{k-m+1}]^T$, and hence, we can say that the equalizer has an m-dimensional channel output observation space. For a CIR of length $L+1$, there are $n_s = \mathcal{M}^{L+m}$ possible combinations of the channel input sequence $\mathbf{I}_k = [I_k \; I_{k-1}, \ldots, I_{k-m-L+1}]^T$ that produce n_s different possible values of the noise-free channel output vector $\tilde{\mathbf{v}}_k = [\tilde{v}_k \; \tilde{v}_{k-1}, \ldots, \tilde{v}_{k-m+1}]^T$. The possible values or particular points in the observation space are referred to as the channel output states, $\mathbf{r}_j$, $j = 1, \ldots, n_s$. We denote each of the $n_s = \mathcal{M}^{L+m}$ possible combinations of the channel input sequence $\mathbf{I}_k$ of length $L+m$ symbols as $\mathbf{s}_j$, $j = 1, \ldots, n_s$, where the channel input state $\mathbf{s}_j = [s_{j1}, \ldots, s_{jp}, \ldots, s_{j(L+m)}]^T$ determines the desired channel output state $\mathbf{r}_j$, $j = 1, \ldots, n_s$ with the aid of the CIR. This is formulated as

$$\tilde{\mathbf{v}}_k = \mathbf{r}_j, \quad \text{if } \mathbf{I}_k = \mathbf{s}_j, \quad j = 1, \ldots, n_s. \tag{6}$$

The channel output state can be defined as

$$\mathbf{r}_j = \mathbf{F}\mathbf{s}_j \tag{7}$$

where $\mathbf{F}$ is the CIR matrix given in the form of

$$\mathbf{F} = \begin{bmatrix} f_0 & f_1 & \ldots & f_L & \ldots & 0 \\ 0 & f_0 & \ldots & f_{L-1} & \ldots & 0 \\ \vdots & \vdots & & & & \vdots \\ 0 & 0 & f_0 & \ldots & f_{L-1} & f_L \end{bmatrix} \tag{8}$$

with f_j, $j = 0, \ldots, L$ being the CIR taps. There are $n_s^i = n_s/\mathcal{M}$ channel output states corresponding to the transmitted symbol $I_{k-\tau} = \mathcal{J}_i$. Linking (1) with (4), we have assigned the weight of the RBF network to the *a priori* probability of occurance of the channel output state, yielding $w_j = p_j^i$ and the centers of the RBF to the channel output state vectors, formulated as $\mathbf{a}_j = \mathbf{r}_j^i$, $i = 1, \ldots, \mathcal{M}$, $j = 1, \ldots, n_s^i$. The number of hidden nodes of each sub-RBF network is given by $N = n_s^i$. In their seminal papers, Chen *et al.* [10], [12], [13] introduced decision feedback into the RBF equalizer in order to reduce its computational complexity via subset center selection. A significantly lower number of noiseless channel states are considered by the RBF decision-feedback equalizer (DFE) in computing the decision variable of (4) with the additional information of the detected symbols, which were fed back from the output of the decision device in order to assist in the RBF subcenter selection. Note that for matched filters that introduce band-edge distortion, a fractional-symbol-spaced RBF equalizer that classifies the fractionally sampled received signal corresponding to the transmitted symbol is required. A study of the fractional-symbol-spaced RBF equalizer is, however, beyond the scope of this paper. Having introduced the concept of the RBF equalizer, we will now present the proposed TEQ scheme in the next section.

B. Introduction to TEQ

Turbo coding was combined with an RBF DFE scheme in [14], where equalization and channel decoding ensued independently. However, it is possible to improve the receiver's performance, if the equalizer is fed by the channel outputs plus the soft decisions provided by the channel decoder, invoking a number of iterative processing steps, as proposed by Douillard *et al.* [1] for a convolutional coded binary phase-shift keying (BPSK) system. This scheme can be extended to $\mathcal{M}$ number of modulation levels, and the corresponding scheme is illustrated in Fig. 3.

The channel encoder is fed with independent binary data d_n and every $\mathfrak{m} = \log_2(\mathcal{M})$ number of bits of the interleaved, channel-encoded data $c_{k,l}$, $l = 1, \ldots, \mathfrak{m}$ is mapped to an $\mathcal{M}$-ary symbol I_k at signaling instant k before transmission. The subscripts n and k, l of the encoded data bits c in Fig. 3 refer to the nth data bit before interleaving, and the lth bit of the kth symbol after interleaving, respectively. In this scheme, the channel is viewed as an inner encoder of a serially concatenated encoder arrangement, since it can be modeled with the aid of a tapped delay line similar to that of a convolutional encoder [1], [15]. This leads to a structure similar to turbo coding, hence, allowing the principle of iterative decoding to be employed, where the equalizer acts as the inner decoder and the channel decoder as

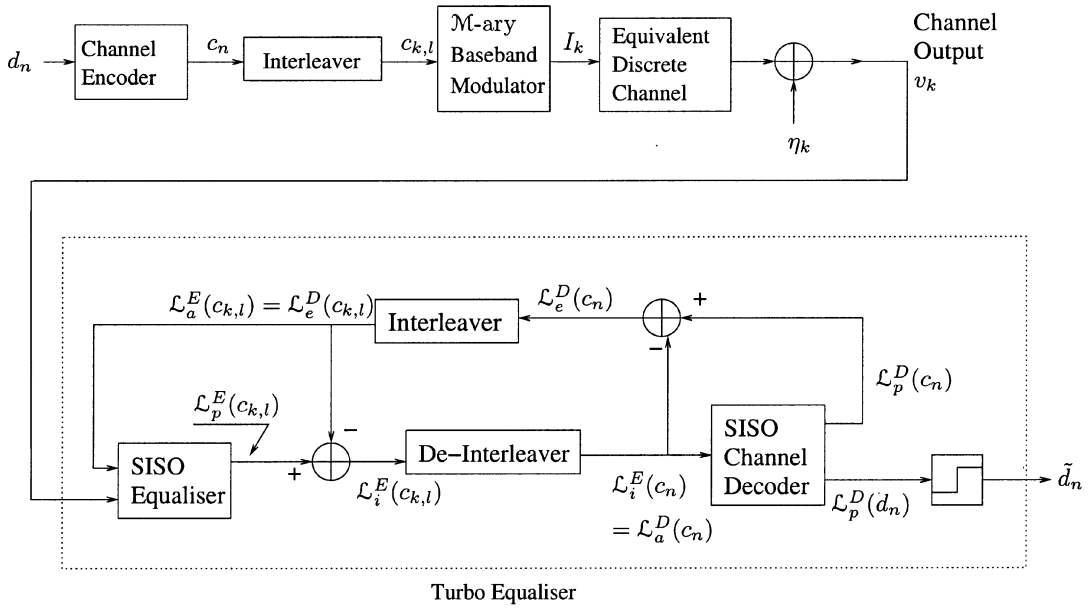


Fig. 3. Serially concatenated coded $\mathcal{M}$-ary system using the turbo equalizer, which performs the equalization, demodulation, and channel decoding iteratively.

the outer decoder. At the receiver, the equalizer and decoder employ a SISO algorithm. In previous TEQ research the SISO equalizer invoked the SOVA [1], the optimal MAP algorithm [3], and linear filters [4]. By contrast, we introduce the proposed RBF-based equalizer as the SISO equalizer in the context of TEQ. The SISO equalizer processes the *a priori* information associated with the coded bits $c_{k,l}$ transmitted over the channel and, in conjunction with the channel output values v_k, computes the *a posteriori* information concerning the coded bits. The soft values of the channel coded bits $c_{k,l}$ are typically quantified in the form of the log-likelihood ratio (LLR) defined as follows:

$$\begin{aligned}\mathcal{L}(c_{k,l}|\mathbf{v}_k) &= \ln\left[\frac{P(c_{k,l}=+1|\mathbf{v}_k)}{P(c_{k,l}=-1|\mathbf{v}_k)}\right] \\ &= L(c_{k,l}=+1|\mathbf{v}_k) - L(c_{k,l}=-1|\mathbf{v}_k), \\ &\qquad l=1,\ldots,\log_2(\mathcal{M}) \quad (9)\end{aligned}$$

where the term $L(u_k = \pm 1|\mathbf{v}_k) = ln(P(c_{k,l} = \pm 1|\mathbf{v}_k))$ is the log-likelihood of the lth data bit $c_{k,l}$ concerning the kth transmitted symbol having the value ± 1 conditioned on the received sequence $\mathbf{v}_k$.

In our description of the turbo equalizer depicted in Fig. 3, we have used the notation $\mathcal{L}^E$ and $\mathcal{L}^D$ to indicate the LLR values output by the SISO equalizer and SISO decoder, respectively. The subscripts e, i, a, and p were used to represent the extrinsic LLR, the combined channel and extrinsic LLR, the *a priori* LLR, and the *a posteriori* LLR, respectively. The following sections will discuss the associated implementation details, where the performance of the RBF TEQ scheme is compared with that of the optimal MAP TEQ scheme of [3].

II. RBF-Assisted TEQ

The RBF network-based equalizer is capable of utilizing the *a priori* information $\mathcal{L}_a^E(c_{k,l})$ provided by the channel decoder of Fig. 3, in order to improve its performance. This *a priori* information can be assigned to the weights of the RBF network [11]. Note that the weights of the equalizer, provided by the soft inputs of the equalizer, are in the form of the *a priori* probability of each possible channel output state p_j^i, as demonstrated in (4). However, the soft information provided by the decoder is in the form of the LLR of the interleaved coded bits $\mathcal{L}_e^D(c_{k,l})$, as shown in Fig. 3. Additionally, the decoder requires soft inputs in terms of the LLR of the coded bits $\mathcal{L}_a^D(c_n)$, whereas, the equalizer provides soft outputs in the form of the CODF of the transmitted symbols. Therefore, a mapping/demapping of the soft information is required between the two components. This process is highlighted below in more detail.

Referring to (7) for the specific scenario of a time-invariant CIR, and assuming that the symbols in the sequence $\mathbf{s}_j$ are statistically independent of each other, the probability of the received channel output vector $\mathbf{r}_j$ is given by

$$\begin{aligned}P(\mathbf{r}_j) &= P(\mathbf{s}_j) \\ &= P\left(s_{j1}\cap,\ldots,s_{jp}\cap,\ldots,s_{j(L+m)}\right) \\ &= \prod_{p=1}^{L+m} P(s_{jp}), \qquad j=1,\ldots,n_s^i. \quad (10)\end{aligned}$$

The transmitted symbol vector component s_{jp}, i.e., the pth symbol in the vector, is given by $\mathfrak{m} = \log_2 \mathcal{M}$ number of bits $c_{jp1}, c_{jp2}, \ldots, c_{jp\mathfrak{m}}$. Therefore, we have

$$\begin{aligned}P(s_{jp}) &= P(c_{jp1}\cap,\ldots,c_{jpl}\cap,\ldots,c_{jp\mathfrak{m}}) \\ &= \prod_{q=1}^{\mathfrak{m}} P(c_{jpl}) \\ &\quad j=1,\ldots,n_s^i, \quad p=1,\ldots,L+m, \quad l=1,\ldots,\mathfrak{m} \quad (11)\end{aligned}$$

We have to map the bits c_{jpl} representing the $\mathcal{M}$-ary symbol s_{jp} to the corresponding bit $\{c_{k,l}\}$. Note that the probability $P(\mathbf{r}_j)$ of the channel output states, and therefore, also the RBF weights defined as p_j^i in (4) are time variant, since the values of $\mathcal{L}_p(c_{k,l})$ are time variant. Based on the definition of the bit LLR of (9),

the probability of bit $c_{k,l}$ having the value of $+1$ or -1 can be obtained from the *a priori* information $\mathcal{L}_a^E(c_{k,l})$ provided by the channel decoder of Fig. 3, according to

$$P(c_{k,l}=\pm 1)=\frac{\exp\left(\frac{-\mathcal{L}_a^E(c_{k,l})}{2}\right)}{1+\exp(-\mathcal{L}_a^E(c_{k,l}))}\exp\left(\frac{1}{2}\cdot c_{k,l}\cdot\mathcal{L}_a^E(c_{k,l})\right). \tag{12}$$

Hence, referring to (10), (11), and (12), the probability $P(\mathbf{r}_j)$ of the received channel output vector can be represented in terms of the bit LLRs $\mathcal{L}_a^E(c_{jpq})$ as follows:

$$\begin{aligned}
P(\mathbf{r}_j) &= P(\mathbf{s}_j)\\
&= \prod_{p=1}^{L+m} P(s_{jp})\\
&= \prod_{p=1}^{L+m}\prod_{q=1}^{\mathtt{m}} P(c_{jpq})\\
&= \prod_{p=1}^{L+m}\prod_{q=1}^{\mathtt{m}} \frac{\exp\left(\frac{-\mathcal{L}_a^E(c_{jpq})}{2}\right)}{1+\exp\left(-\mathcal{L}_a^E(c_{jpq})\right)}\\
&\quad\cdot\exp\left(\frac{1}{2}\cdot c_{jpq}\cdot\mathcal{L}_a^E(c_{jpq})\right)\\
&= C_{\mathcal{L}_a^E(\mathbf{s}_j)}\cdot\prod_{p=1}^{L+m}\prod_{q=1}^{\mathtt{m}}\exp\left(\frac{1}{2}\cdot c_{jpq}\cdot\mathcal{L}_a^E(c_{jpq})\right)\\
&= C_{\mathcal{L}_a^E(\mathbf{s}_j)}\cdot\exp\left(\frac{1}{2}\sum_{p=1}^{L+m}\sum_{q=1}^{\mathtt{m}} c_{jpq}\cdot\mathcal{L}_a^E(c_{jpq})\right)\\
j &= 1,\ldots,n_s^i
\end{aligned} \tag{13}$$

where the constant $C_{\mathcal{L}_a^E(\mathbf{s}_j)} = \prod_{p=1}^{L+m}\prod_{q=1}^{\mathtt{m}}(\exp(-\mathcal{L}_a^E(c_{jpq})/2))/(1+\exp(-\mathcal{L}_a^E(c_{jpq})))$ is independent of the bit c_{jpq}.

Therefore, we have demonstrated how the soft output $\mathcal{L}_a^E(c_{k,l})$ of the channel decoder of Fig. 3 can be utilized by the RBF equalizer. Another way of viewing this process is that the RBF equalizer is trained by the information generated by the channel decoder. The RBF equalizer provides the *a posteriori* LLR values of the bits $c_{k,l}$ according to

$$\mathcal{L}_p^E(c_{k,l})=\ln\left(\frac{\sum_{\substack{i\\ c_{k,l}=+1}} f_{\mathrm{RBF}}^i(\mathbf{v}_k)}{\sum_{\substack{i\\ c_{k,l}=-1}} f_{\mathrm{RBF}}^i(\mathbf{v}_k)}\right) \tag{14}$$

where $f_{\mathrm{RBF}}^i(\mathbf{v}_k)$ was defined by (4), and the received sequence $\mathbf{v}_k$ is shown in Fig. 3. In the next section, we will provide a comparative study of the RBF equalizer and the conventional MAP equalizer of [16].

III. Comparison of the RBF and MAP Equalizers

The MAP equalizer provides the *a posteriori* LLR values $\mathcal{L}_p^E$ by calculating the transition probabilities through the trellis stages, based on a sequence of the received signal $\mathbf{v}_k$. We will first present a brief description of the MAP equalizer in order to show the commonalities between the MAP and RBF equalizer structures. For the MAP equalizer, the *a posteriori* LLR value $\mathcal{L}_p^E$ of the coded bit $c_{k,ll}$, given the received sequence $\mathbf{v}_k$ of Fig. 3, can be calculated according to [3]

$$\mathcal{L}_p^E(c_{k,l})=\ln\left(\frac{\sum_{(s',s)\Rightarrow c_{k,l}=+1} p(s',s,\mathbf{v}_k)}{\sum_{(s',s)\Rightarrow c_{k,l}=-1} p(s',s,\mathbf{v}_k)}\right) \tag{15}$$

where s' and s denote the states of the corresponding trellis [6] at trellis stages $k-1$ and k, respectively. The joint probability $p(s',s,\mathbf{v}_k)$ is the product of three factors [3]

$$p(s',s,\mathbf{v}_k)=\underbrace{p(s',v_{j<k})}_{\alpha_{k-1}(s')}\cdot\underbrace{P(s|s')\cdot p(v_k|s',s)}_{\gamma_k(s',s)}\cdot\underbrace{p(v_{j>k}|s)}_{\beta_k(s)} \tag{16}$$

where the term $\alpha_{k-1}(s')$ and $\beta_k(s)$ are the so-called forward- and backward-oriented transition probabilities, respectively, which can be obtained recursively, as follows [3]:

$$\alpha_k(s)=\sum_{s'}\gamma_k(s',s)\cdot\alpha_{k-1}(s') \tag{17}$$

$$\beta_{k-1}(s)=\sum_{s}\gamma_k(s',s)\cdot\beta_k(s). \tag{18}$$

Furthermore, $\gamma_k(s',s)$, $k=1,\ldots,\mathcal{F}$ represents the trellis transitions between the trellis stages $(k-1)$ and k. The trellis has to be of finite length and for the case of MAP equalization, this corresponds to the length $\mathcal{F}$ of the received sequence or the transmission burst. The branch transition probability $\gamma_k(s',s)$ can be expressed as the product of the *a priori* probability $P(s|s')=P(I_k)=\prod_{l=1}^{\mathtt{m}}P(c_{k,l})$ and the transition probability $p(v_k|s',s)$

$$\gamma_k(s',s)=\prod_{l=1}^{\mathtt{m}}P(c_{k,l})\cdot p\left(v_k|s',s\right). \tag{19}$$

The transition probability is given by

$$p(v_k|s',s)=\frac{1}{\sqrt{2\pi\sigma_\eta^2}}\exp\left(-\frac{(v_k-\tilde{v}_k)^2}{2\sigma_\eta^2}\right) \tag{20}$$

where $\tilde{v}_k$ is the noiseless channel output, and the *a priori* probability of the lth bit $c_{k,l}$ of symbol I_k, being a logical 1 or a logical 0, can be expressed in terms of its LLR values, according to (12). Since the term $1/\sqrt{2\pi\sigma_\eta^2}$ in the transition probability expression of (20) and the term $(\exp(-\mathcal{L}_a^E(c_{k,l})/2))/(1+\exp(-\mathcal{L}_a^E(c_{k,l})))$ in the *a priori* probability formula of (12) are constant over the summation in the numerator and denominator of (15), they cancel out. Hence, the transition probability is calculated according to [3]

$$\gamma_k(s',s)=w_k\cdot\gamma_k^*(s',s), \tag{21}$$

$$\gamma^*(s',s)=\exp\left(-\frac{|v_k-\tilde{v}_k|^2}{2\sigma_\eta^2}\right) \tag{22}$$

$$\begin{aligned}
w_k &= \prod_{l=1}^{\mathtt{m}}\exp\left(\frac{1}{2}\cdot c_{k,l}\cdot\mathcal{L}_a^E(c_{k,l})\right)\\
&= \exp\left(\frac{1}{2}\sum_{l=1}^{\mathtt{m}}\cdot c_{k,l}\cdot\mathcal{L}_a^E(c_{k,l})\right).
\end{aligned} \tag{23}$$

Equations (15)–(23) describe the MAP equalization process. Note the similarity of the transition probability derived for the MAP equalizer in (21) to the PDF of the RBF equalizer's ith symbol described by (4), where the terms w_k and $\gamma^*(s', s)$ are the RBF's weight and activation function, respectively, while the number of RBF nodes n_s^i is one. We also note that the computational complexity of both the MAP and the RBF equalizers can be reduced by representing the output of the equalizers in the logarithmic domain, utilizing the generalized Jacobian logarithmic relationship defined by [17]

$$\ln\left(\sum_{k=1}^{n} e^{\lambda_k}\right) = J\left(\lambda_n, J\left(\lambda_{n-1}, \ldots, J\left(\lambda_3, J(\lambda_2, \lambda_1)\right), \ldots,\right)\right) \tag{24}$$

where $J(\lambda_1, \lambda_2) = \ln(e^{\lambda_1} + e^{\lambda_2}) = \max(\lambda_1, \lambda_2) + \ln(1 + e^{-|\lambda_1 - \lambda_2|}) \approx \max(\lambda_1, \lambda_2) + \varphi_c(|\lambda_1 - \lambda_2|)$. The correction function $\varphi_c(x) = \ln(1 + e^{-x})$ has a dynamic range of $\ln(2) \geq \varphi_c(x) > 0$, and it is significant only for small values of x. Thus, $\varphi_c(x)$ can be tabulated in a lookup table in order to reduce the computational complexity [17]. The correction function $\varphi_c(\cdot)$ only depends on $|\lambda_1 - \lambda_2$, therefore, the lookup table is one dimensional, and simulation results show that only a few values have to be stored [18]. The RBF equalizer based on the Jacobian logarithm, highlighted in [14], was termed as the Jacobian RBF equalizer.

The memory of the MAP equalizer is limited by the length of the trellis, provided that decisions about the kth transmitted symbol I_k are made in possession of the information related to all the received symbols of a transmission burst. In the MAP algorithm, the recursive relationships of the forward and backward transition probabilities of (17) and (18), respectively, allow us to avoid processing the entire received sequence $\mathbf{v}_k$ every time the *a posteriori* LLR $\mathcal{L}_p^E(c_{k,l})$ is evaluated from the joint probability $p(s', s, \mathbf{v}_k)$, according to (15). This approach is different from that of the RBF-based equalizer having a feedforward order of m, where the received sequence $\mathbf{v}_k$ of m-symbols is required each time the *a posteriori* LLR $\mathcal{L}_p^E(c_{k,l})$ is evaluated using (14). However, the MAP algorithm has to process the received sequence both in a forward- and backward-oriented fashion and store both the forward and backward recursively calculated transition probabilities $\alpha_k(s)$ and $\beta_k(s)$, before the LLR values $\mathcal{L}_p^E(c_{k,l})$ can be calculated from (15). The equalizer's delay facilitates invoking information from the future samples $v_k, \ldots, v_{k-\tau+1}$ in the detection of the transmitted symbol $I_{k-\tau}$. In other words, the delayed decision of the MAP equalizer provides the necessary information concerning the future samples $v_{j>k}$, relative to the delayed kth decision, to be utilized, and the information of the future samples is generated by the backward recursion of (18).

The MAP equalizer exhibits optimum performance. However, if decision feedback is used in the RBF subset center selection, as proposed in [12], the performance of the RBF DFE TEQ in conjunction with the idealistic assumption of *correct* decision feedback is better than that of the MAP TEQ, due to the increased Euclidean distance between channel states, as will be demonstrated in Section V. However, this is not so for the more practical RBF DFE feeding back the detected symbols, which may be erroneous.

IV. Comparison of the Jacobian RBF and Log-MAP Equalizer

Building on Section III, in this section, the Jacobian logarithmic algorithm is invoked, in order to reduce the computational complexity of the MAP algorithm. We denote the forward, backward, and transition probability in the logarithmic form as follows:

$$A_k(s) = \ln\left(\alpha_k(s)\right) \tag{25}$$

$$B_k(s) = \ln\left(\beta_k(s)\right) \tag{26}$$

$$\Gamma_k(s', s) = \ln\left(\gamma_k(s', s)\right) \tag{27}$$

which we also used in Section III. Thus, we could rewrite (17) as

$$\begin{aligned} A_k(s) &= \ln\left(\sum_{s'} \gamma_k(s', s) \cdot \alpha_{k-1}(s')\right) \\ &= \ln\left(\sum_{s'} \exp\left(\Gamma_k(s', s) + A_{k-1}(s')\right)\right) \end{aligned} \tag{28}$$

and (18) as:

$$\begin{aligned} B_{k-1}(s') &= \ln\left(\sum_{s} \gamma_k(s', s) \cdot \beta_k(s)\right) \\ &= \ln\left(\sum_{s} \exp\left(\Gamma_k(s', s) + B_k(s)\right)\right). \end{aligned} \tag{29}$$

From (28) and (29), the logarithmic-domain forward and backward recursions can be evaluated, once $\Gamma_k(s', s)$ was obtained. In order to evaluate the logarithmic-domain branch metric $\Gamma_k(s', s)$, (21)–(23) and (27) are utilized to yield

$$\Gamma_k(s', s) = -\frac{|v_k - \tilde{v}_k|^2}{2\sigma_\eta^2} + \frac{1}{2}\sum_{l=1}^{\mathfrak{m}} c_{k,l} \cdot \mathcal{L}_a^E(c_{k,l}). \tag{30}$$

By transforming $\alpha_k(s)$, $\gamma_k(s', s)$, and $\beta_k(s)$ into the logarithmic domain in the context of the logarithmic maximum *a posteriori* (Log-MAP) algorithm, the expression of the LLR, namely $\mathcal{L}_p^E(c_{k,l})$ in (15) is also modified to yield (31) as shown at the bottom of the next page.

In the decoding trellis of the scheme considered there are $\mathcal{M}$ possible transitions from state s' to all possible states s, or to state s from all possible states s'. Hence, there are $\mathcal{M} - 1$ summations of the exponentials in the forward and backward recursions of (28) and (29), respectively. Using the Jacobian logarithmic relationship of (24), $\mathcal{M} - 1$ summations of the exponentials require $2(\mathcal{M} - 1)$ additions/subtractions, $(\mathcal{M}-1)$ maximum search operations, and $(\mathcal{M}-1)$ table lookup steps. Together with the $\mathcal{M}$ additions necessitated to evaluate the term $\Gamma_k(s', s) + A_{k-1}(s')$ and $\Gamma_k(s', s) + B_k(s)$ in (28) and (29), respectively, the forward and backward recursions require a total of $(6\mathcal{M} - 4)$ additions/subtractions, $2(\mathcal{M} - 1)$ maximum search operations, and $2(\mathcal{M} - 1)$ table lookup steps. Assuming that the term $(1/2)\sum_{l=1}^{\mathfrak{m}} c_{k,l} \cdot \mathcal{L}_a^E(c_{k,l})$ in (30) is a known weighting coefficient, evaluating the branch metrics given by (30) requires a total of two additions/subtractions, one multiplication, and one division.

By considering a trellis having χ number of states at each trellis stage and $\mathcal{M}$ legitimate transitions leaving each state, there are $(1/2)\mathcal{M}\chi$ number of transitions due to the bit $c_{k,l} = +1$. Each of these transitions belongs to the set $(s', s) \Rightarrow c_{k,l} = +1$. Similarly, there will be $(1/2)\mathcal{M}\chi$ number of $c_{k,l} = -1$ transitions, which belong to the set $(s', s) \Rightarrow c_{k,l} = -1$. Evaluating $A_k(s)$, $B_{k-1}(s')$, and $\Gamma_k(s', s)$ of (28), (29), and (30), respectively, at each trellis stage k associated with a total of $\mathcal{M}\chi$ transitions requires $\mathcal{M}\chi(6\mathcal{M} - 2)$ additions/subtractions, $\mathcal{M}\chi(2\mathcal{M} - 2)$ maximum search operations, $\mathcal{M}\chi(2\mathcal{M} - 2)$ table lookup steps, plus $\mathcal{M}\chi$ multiplications and $\mathcal{M}\chi$ divisions. With the terms $A_k(s)$, $B_{k-1}(s')$, and $\Gamma_k(s', s)$ of (28), (29), and (30) evaluated, computing the LLR $\mathcal{L}_p^E(c_{k,l})$ of (31) (shown at the bottom of the page) using the Jacobian logarithmic relationship of (24) for evaluating the summation terms $\ln(\sum_{(s',s)\Rightarrow c_{k,l}=+1} \exp(\cdot))$ and $\ln(\sum_{(s',s)\Rightarrow c_{k,l}=-1} \exp(\cdot))$ requires a total of $4((1/2)\mathcal{M}\chi - 1) + 2\mathcal{M}\chi + 1$ additions/subtractions, $\mathcal{M}\chi - 2$ maximum search operations, and $\mathcal{M}\chi - 2$ table lookup steps. The number of states at each trellis stage is given by $\chi = \mathcal{M}^L = n_{s,f}/\mathcal{M}$. Therefore, the total computational complexity associated with generating the *a posteriori* LLRs using the Jacobian logarithmic relationship for the Log-MAP equalizer is given in Table I.

For the Jacobian RBF equalizer, the LLR expression of (14) is rewritten in terms of the logarithmic form $\ln(f^i_{\mathrm{RBF}}(\mathbf{v}_k))$, yielding

$$\begin{aligned}\mathcal{L}_p^E(c_{k,l}) &= \ln\left(\frac{\sum_{\substack{i\\ c_{k,l}=+1}} f^i_{\mathrm{RBF}}(\mathbf{v}_k)}{\sum_{\substack{i\\ c_{k,l}=-1}} f^i_{\mathrm{RBF}}(\mathbf{v}_k)}\right)\\ &= \ln\left(\frac{\sum_{\substack{i\\ c_{k,l}=+1}} \exp\left(\ln\left(f^i_{\mathrm{RBF}}(\mathbf{v}_k)\right)\right)}{\sum_{\substack{i\\ c_{k,l}=-1}} \exp\left(\ln\left(f^i_{\mathrm{RBF}}(\mathbf{v}_k)\right)\right)}\right)\\ &= \ln\left(\sum_{\substack{i\\ c_{k,l}=+1}} \exp\left(\ln\left(f^i_{\mathrm{RBF}}(\mathbf{v}_k)\right)\right)\right)\\ &\quad - \ln\left(\sum_{\substack{i\\ c_{k,l}=-1}} \exp\left(\ln\left(f^i_{\mathrm{RBF}}(\mathbf{v}_k)\right)\right)\right),\\ &\quad l = 1, \ldots, \mathfrak{m}. \end{aligned} \tag{32}$$

TABLE I
COMPUTATIONAL COMPLEXITY OF GENERATING *A POSTERIORI* LLR $\mathcal{L}_p^E$ FOR LOG-MAP EQUALIZER AND JACOBIAN RBF EQUALIZER [14]. THE RBF EQUALIZER ORDER IS DENOTED BY m AND NUMBER OF RBF CENTERS IS n_s^i. THE NOTATION $n_{s,f} = \mathcal{M}^{L+1}$ INDICATES NUMBER OF TRELLIS STATES FOR LOG-MAP EQUALIZER AND ALSO NUMBER OF SCALAR CHANNEL STATES FOR JACOBIAN RBF EQUALIZER

	Log-MAP	Jacobian RBF
subtraction and addition	$n_{s,f}(6\mathcal{M}+2)-3$	$n_{s,f}+\mathcal{M}n_s^i(m+2)-4$
multiplication	$n_{s,f}$	$n_{s,f}$
division	$n_{s,f}$	$n_{s,f}$
max	$n_{s,f}(2\mathcal{M}-1)-2$	$\mathcal{M}n_s^i-2$
table look-up	$n_{s,f}(2\mathcal{M}-1)-2$	$\mathcal{M}n_s^i-2$

Therefore, similar to the Log-MAP equalizer, the computational complexity associated with generating the *a posteriori* LLR $\mathcal{L}_p^E$ for the Jacobian RBF equalizer is given in Table I. Fig. 4 compares the number of additions/subtractions per turbo iteration involved in evaluating the *a posteriori* LLRs $\mathcal{L}_p^E$ for the Log-MAP equalizer and Jacobian RBF equalizer according to Table I. More explicitly, the complexity is evaluated by varying the feedforward order m for different values of L, where $(L+1)$ is the CIR duration under the assumption that the feedback order is $n = L$, and the number of RBF centers is $n_s^i = \mathcal{M}^{m+L-n}/\mathcal{M}$. Since the number of multiplications and divisions involved is similar, and by comparison, the number of maximum search and table lookup stages is insignificant, the number of log-domain additions/subtractions incurred in Fig. 4 approximates the relative computational complexities involved. Fig. 4 shows significant computational complexity reduction upon using Jacobian RBF equalizers of relatively low feedforward order, especially for higher order modulation modes, such as $\mathcal{M} = 64$. The figure also shows an exponential increase of the computational complexity, as the CIR length increases. Observe in Fig. 4 that as a rule of thumb, the feedforward order of the Jacobian RBF DFE must not exceed the CIR length $(L+1)$ in order to achieve a computational complexity improvement relative to the Log-MAP equalizer, provided that we use the optimal number of RBF centers, namely, $n_s^i = \mathcal{M}^{m+L-n}/\mathcal{M}$.

$$\begin{aligned}\mathcal{L}_p^E(c_{k,l}) &= \ln\left(\frac{\sum_{(s',s)\Rightarrow c_{k,l}=+1} \alpha_{k-1}(s')\cdot\gamma_k(s',s)\cdot\beta_k(s)}{\sum_{(s',s)\Rightarrow c_{k,l}=-1} \alpha_{k-1}(s')\cdot\gamma_k(s',s)\cdot\beta_k(s)}\right)\\ &= \ln\left(\frac{\sum_{(s',s)\Rightarrow c_{k,l}=+1} \exp\left(A_{k-1}(s')+\Gamma_k(s',s)+B_k(s)\right)}{\sum_{(s',s)\Rightarrow c_{k,l}=-1} \exp\left(A_{k-1}(s')+\Gamma_k(s',s)+B_k(s)\right)}\right)\\ &= \ln\left(\sum_{(s',s)\Rightarrow c_{k,l}=+1} \exp\left(A_{k-1}(s')+\Gamma_k(s',s)+B_k(s)\right)\right)\\ &\quad - \ln\left(\sum_{(s',s)\Rightarrow c_{k,l}=-1} \exp\left(A_{k-1}(s')+\Gamma_k(s',s)+B_k(s)\right)\right), \qquad q = 1, \ldots, \mathfrak{m}\end{aligned} \tag{31}$$

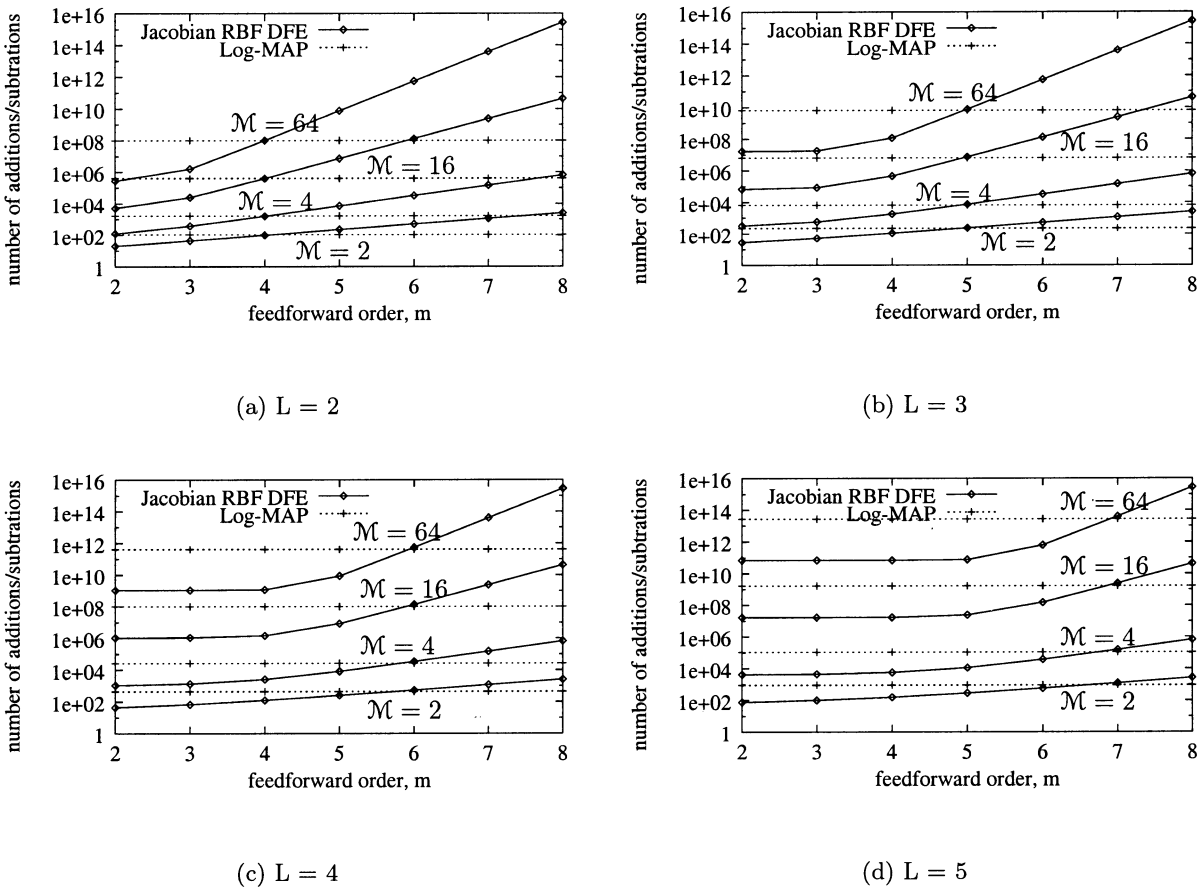


Fig. 4. Number of additions/subtractions per iteration for the Jacobian RBF DFE of varying equalizer order m and the Log-MAP equalizer for various values of L, where $L+1$ is the CIR length. The feedback order of the Jacobian RBF DFE is set to $n = L$ and the number of RBF centers is set to $n_s^i = \mathcal{M}^{m+L-n}/\mathcal{M}$.

We conclude that the Jacobian RBF DFE having a feedforward order that is equivalent to the symbol-spaced CIR length provides a computational complexity advantage, although only in terms of the additions/subtractions imposed, whereas, the number of multiplications/divisions is the same as for Log-MAP-based equalizers.

The length of the trellis determines the storage requirements of the Log-MAP equalizer, since the Log-MAP algorithm has to store both the forward- and backward-recursively calculated metrics $A_k(s)$ and $B_{k-1}(s')$ before the LLR values $\mathcal{L}_p^E(c_{k,l})$ can be calculated. For the Jacobian RBF DFE, we have to store the value of the RBF centers, and the storage requirements will depend on the CIR length $L+1$ as well as on the $\mathcal{M}$-ary modulation mode.

V. RBF TEQ Performance

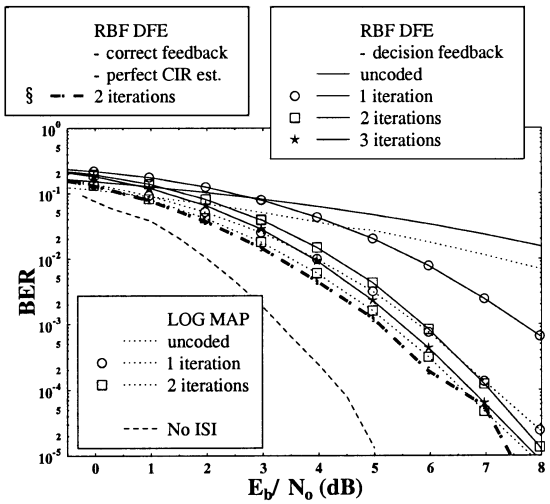
The schematic of the entire system was shown in Fig. 3, where the transmitted source bits are convolutionally encoded, channel interleaved, and mapped to an $\mathcal{M}$-ary modulated symbol. The encoder utilized is a half-rate recursive systematic convolutional (RSC) code, having a constraint length of $K = 5$ and octal generator polynomials of $G_0 = 35$ and $G_1 = 23$ as recommended in the global system for mobile communications (GSM). The transmission burst structure used in this system is the FRAMES multiple access (FMA) 1 nonspread speech burst, as specified in the Pan-European FRAMES proposal [19], which has a total duration of 72 μs, and it is constituted by a 27-symbol channel sounding or training midamble surrounded by two 72-symbol data segments. In order to decide on the amount of tolerable delay, and hence, on the depth of the channel interleaver, we considered the maximum affordable delay of a time-division multiple access/time-division duplex (TDMA/TDD)-based interactive speech/video system, which employs eight uplink and eight downlink slots. Hence, 1 transmission slot will be available after 16 TDMA slots. In our investigations, the transmission delay of the BPSK, 4QAM and 16QAM system was limited to approximately 30 ms. This corresponds to 3456 data symbols transmitted within 30 ms, and hence, 3456-bit, 6912-bit and 13 824-bit random channel interleavers were utilized for BPSK, 4QAM, and 16QAM, respectively. In this context, it is interesting to note that even though the performance is expected to degrade, when a higher order modem mode is used, the associated longer interleaver partially compensates for this performance degradation. A three-path, symbol-spaced fading channel of equal weights was utilized, where the Rayleigh fading statistics obeyed a normalized Doppler frequency of 3.3615×10^{-5}. The CIR was assumed to be burst invariant. The discrete-time channel utilized in our simulations is assumed to be a concatenation of the transmitter filter, the wireless channel, and the receiver filter.

The RBF TEQ system incorporates iterative least mean square (LMS)-based CIR estimation [20], where we first estimate the CIR with the aid of the 27-symbol training midamble of the transmission burst during the first TEQ iteration. During the subsequent TEQ iterations, all the symbols in the transmission burst, which include the estimated data symbols from the previous iterations, were utilized for training the LMS CIR estimator. Therefore, the number of effective training symbols used for CIR estimation was higher. An initial step size of 0.1 was used for the first iteration, while in the subsequent iterations, the CIR was reestimated and refined using a smaller step size of 0.01.

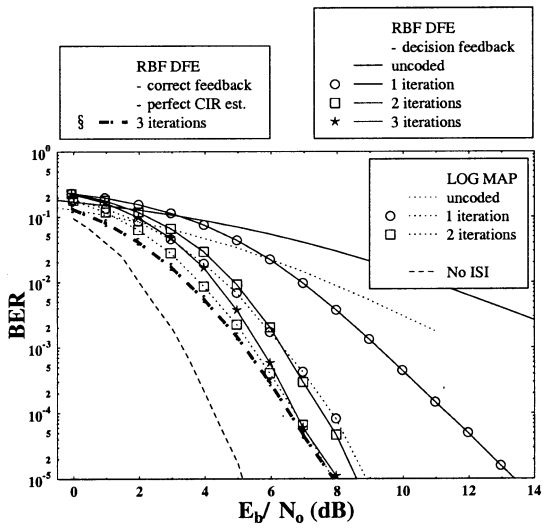
Fig. 5(a)–(c) portray the performance of the Log-MAP TEQ with advent of perfect CIR information, and that of the Jacobian RBF DFE TEQ for BPSK, 4QAM, and 16QAM, respectively. The Jacobian RBF DFE had a feedforward order of $m = 3$, feedback order of $n = 2$, and decision delay of $\tau = 2$ symbols. Fig. 5(a) and Fig. 5(b) show that the Log-MAP TEQ and the Jacobian RBF DFE TEQ converged to a similar bit-error rate (BER) performance for BPSK and 4QAM systems. However, the Log-MAP TEQ required a lower number of iterations for most cases. Specifically, two iterations were required for the Log-MAP TEQ for all the modulation schemes studied. By contrast, two, three, and eight iterations were necessary for the BPSK, 4QAM, and 16QAM Jacobian RBF DFE TEQ, respectively, in order to achieve near-perfect convergence, since the Log-MAP TEQ exhibited a better BER performance for an uncoded system than that of the Jacobian RBF DFE. The performance of the Log-MAP TEQ and that of the Jacobian RBF DFE TEQ at the final iteration was about 2 dB away from the zero-intersymbol interference (ISI) Gaussian BER curve for both BPSK and 4QAM, at a BER of 10^{-3}. For 16QAM, the effect of error propagation degraded the performance of the Jacobian RBF DFE TEQ by 7 dB at a BER of 10^{-3}, when we compared the Jacobian RBF DFE TEQ's correct feedback-based and decision-feedback-assisted performance after three iterations, as seen in Fig. 5(c). The performance of the 16QAM Jacobian RBF DFE TEQ is worse than that of the Log-MAP TEQ by only 2.5 dB at the final iteration, and is only 4.5 dB away from the zero-ISI Gaussian BER curve at a BER of 10^{-3}. We note finally that the performance can be further improved by increasing the equalizer's feedforward order at the expense of a higher computational complexity, as discussed in [11] for an uncoded system.

The iteration gain was defined as the difference between the channel signal-to-noise ratio (SNR) required in order to achieve a certain BER after one iteration, and the corresponding channel SNR required after n number of iterations. The iteration gain of the Jacobian RBF DFE TEQ at the final iteration was 1.3, 3.5, and 16 dB at a BER of 10^{-3} for the modulation modes of BPSK, 4QAM, and 16QAM, respectively. By contrast, for the Log-MAP TEQ, the corresponding iteration gains were 0.5, 0.9, and 2 dB for the modulation modes of BPSK, 4QAM, and 16QAM, respectively. The iteration gain was higher for the higher order modulation modes, since the distance between two neighboring points in the higher order constellations was lower, and hence, it was more gravely affected by ISI and noise.

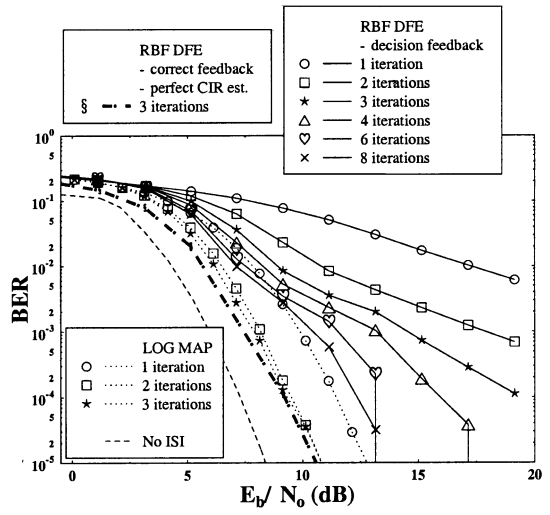
Based on the implementation complexity summarized in Table I and considering only the addition/subtraction opera-



(a)



(b)



(c)

Fig. 5. Performance of the Log-MAP TEQ and Jacobian RBF DFE TEQ over the three-path Rayleigh fading channel. Jacobian RBF DFE has a feedforward order of $m = 3$, feedback order of $n = 2$, and decision delay of $\tau = 2$ symbols. RBF feedback is decision directed.

tions, since the number of multiplications/divisions is similar, the per iteration complexity of the Jacobian RBF DFE TEQ was approximately a factor of 2.5, 4.4, and 16.3 lower than that of the Log-MAP TEQ, for BPSK, 4QAM, and 16QAM, respectively. The overall computational complexity reduction of the Jacobian RBF DFE TEQ compared to the Log-MAP TEQ using two iterations, when considering the extra iterations necessary for achieving the best performance, are 2.5, 3, and 4 for BPSK, 4QAM, and 16QAM, respectively. Overall, due to the error propagation effects, which gravely degraded the performance of the Jacobian RBF DFE TEQ in the context of 16QAM, the Jacobian RBF DFE TEQ was restricted to providing a favorable performance versus complexity tradeoff only for the lower order modulation modes, such as BPSK and 4QAM.

TEQ research has been focused on developing reduced-complexity equalizers, such as the receiver structure proposed by Glavieux *et al.* [4], where the equalizer is constituted by two linear filters. Motivated by this trend, Yeap *et al.* [21] proposed a reduced-complexity trellis-based equalizer scheme equalizing the in-phase and quadrature-phase component of the transmitted signal independently. This novel reduced-complexity equalizer is termed as the in-phase/quadrature-phase equalizer (I/Q EQ), where the associated cross coupling of the in-phase and quadrature-phase tranmsitted signal components was removed, hence, rendering the real and imaginary parts of the channel output to be dependent only on the corresponding quadrature component of the modulated signal. This reduces the number of channel states from $(\mathcal{M})^{L+1}$ to $(\sqrt{\mathcal{M}})^{L+1}$. The complexity of the RBF DFE is reduced similarly to that of the I/Q EQ by equalizing the in-phase and quadrature-phase components of the transmitted signal separately in [22]. Yee *et al.* demonstrated in [22] that the I/Q RBF-TEQ attained the same performance as the conventional trellis-based TEQ, while achieving a complexity reduction by a factor of 1.5 and 109.6 for 4QAM and 16QAM, respectively. Note that the high performance degradation of the 16QAM scheme, which was attributed to the erroneous decision-feedback effects of the RBF DFE and was characterized in Fig. 5(c), is resolved with the aid of the I/Q RBF TEQ arrangement, since the number of possible transmitted symbols to be detected is reduced from 16 to $(\sqrt{16} =)4$. In the following section, we proposed another novel method of reducing the TEQ's complexity by making use of the fact that the RBF DFE evaluates its output on a symbol-by-symbol basis.

VI. Reduced-Complexity RBF-Assisted TEQ

The Log-MAP algorithm requires forward and backward recursions through the entire sequence of symbols in the received burst, in order to evaluate the forward and backward transition probability of (17) and (18), before calculating the *a posteriori* LLR values $\mathcal{L}_p(c_{k,l})$. Effectively, the computation of the *a posteriori* LLRs $\mathcal{L}_p(c_{k,l})$ is performed on a burst-by-burst basis. The RBF-based equalizer, however, performs the evaluation of the *a posteriori* LLRs $\mathcal{L}_p(c_{k,l})$ on a symbol-by-symbol basis. Therefore, in order to reduce the associated computational complexity, the RBF-based TEQ may skip evaluating the bit LLRs of a particular symbol, which were formulated in (14), in the current iteration, when the symbol has a low error probability. Equivalently, this condition implies that all the $\mathfrak{m}$ bits in the symbol exhibit a high *a priori* LLR magnitude $|\mathcal{L}_a^E(c_{k,l})|$ after channel decoding in the previous iteration. If, however, this is not the case, the equalizer invokes a further iteration and attempts to improve the decoder's reliablility estimation of the coded bits. The output $f_{\mathrm{RBF}}^i(\mathbf{v}_k)$ of the RBF equalizer provides the LLR values of bit $c_{k,l}$ at instant k, according to (14). The bit LLRs generated by the equalizer in the current iteration obey an approximately linear relationship versus the LLRs recorded at the input of the equalizer. These equalizer input LLRs were provided by the channel decoder in the previous iteration, as is demonstrated in Figs. 6 and 7 for the BPSK and 16QAM modes, respectively, when communicating over a three-path fading channel having symbol-spaced equal CIR tap weights. The Rayleigh fading statistics obeyed a normalized Doppler frequency of 3.3615×10^{-5}. Therefore, the bit LLRs evaluated by the RBF equalizer can be *estimated* based on this near-linear relationship portrayed in Figs. 6 and 7 according to

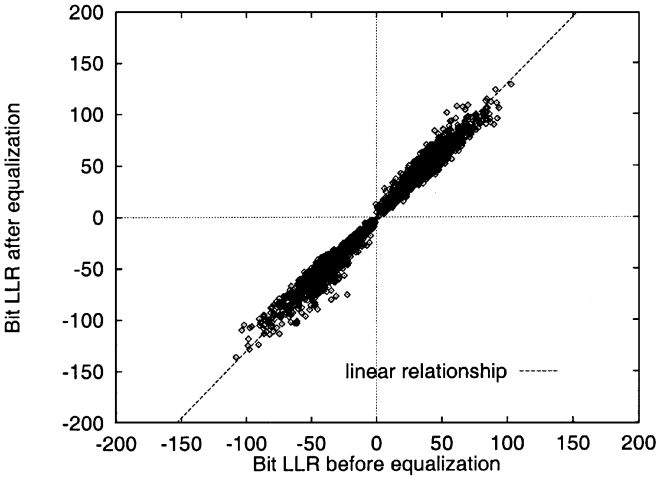
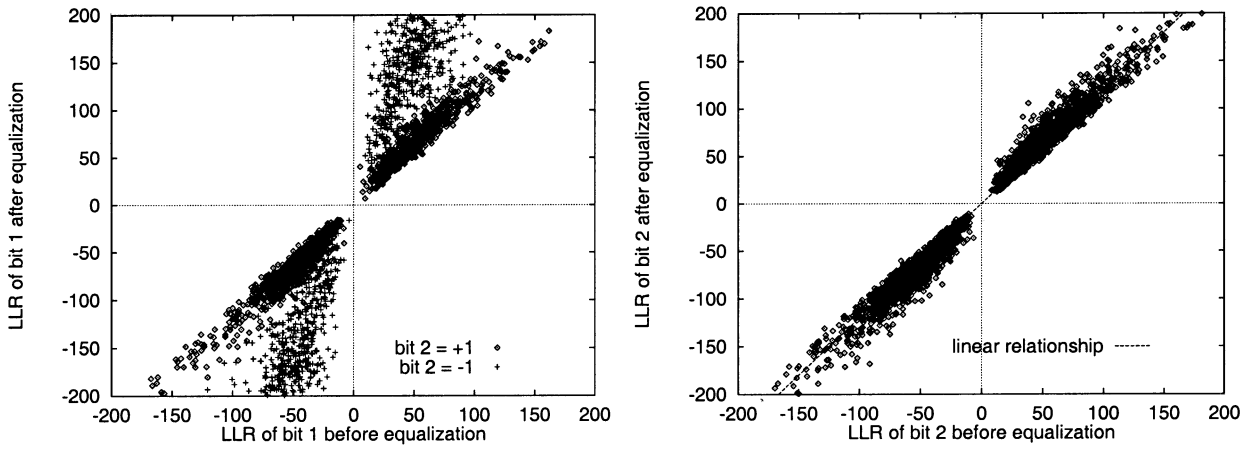


Fig. 6. LLR of the BPSK RBF turbo equalized bits before and after equalization over the three-path, symbol-spaced fading channel of equal CIR tap weights, where the Rayleigh fading statistic obeyed a normalized Doppler frequency of 3.3×10^{-5}, at an E_b/N_0 of 4 dB using BPSK.

$$\tilde{L}_p^E(c_{k,l}) = g \cdot L_a^E(c_{k,l}) \tag{33}$$

where $\tilde{L}_p^E(c_{k,l})$ is the estimated LLR of the lth bit of the transmitted symbol I_k at instant k, based on the decoder's soft output $L_a^E(c_{k,l})$ and where g is the LLR gradient. The LLR gradient g can be inferred from Figs. 6 and 7 for the BPSK and 16QAM modem modes, respectively. Note that for the 16QAM scenario, the LLR gradient of the four bits constituting a symbol, namely, that of $c_{k,1}$, $c_{k,2}$, $c_{k,3}$, and $c_{k,4}$, differs, depending on the specific bit's protection distance measured from the demodulation decision boundary of each of the received phasors in the absence of noise. Here, the bits $c_{k,1}$ and $c_{k,2}$ correspond to the in-phase component, while the bits $c_{k,3}$ and $c_{k,4}$ correspond to the quadrature-phase component. Considering only the in-phase component, Fig. 7(a) demonstrates that the LLR gradient of the most significant bit (MSB), namely, that of bit 1 or $c_{k,1}$, depends on the least significant bit (LSB), namely, on bit 2 or $c_{k,2}$. This is a consequence of the different protection distances of the symbols corresponding to $(c_{k,1}, c_{k,2}) = (\pm 1, -1)$ and



(a) Bit 1, $c_{k,1}$ (b) Bit 2, $c_{k,2}$

Fig. 7. LLR of the 16QAM RBF turbo equalized bits before and after equalization over the three-path, symbol-spaced fading channel of equal CIR tap weights, where the Rayleigh fading statistic obeyed a normalized Doppler frequency of 3.3×10^{-5}, at an E_b/N_0 of 16 dB using 16QAM.

$(c_{k,1}, c_{k,2}) = (\pm 1, +1)$. The same characteristic properties are also valid for the quadrature-phase component.

Next, we have to set the LLR magnitude threshold $|\mathcal{L}|_{\text{threshold}}$, where the estimated coded bits $c_{k,l}$ output by the decoder in the previous iteration become sufficiently reliable for refraining from further iterations. Hence, the symbols exhibiting an LLR value which is above this particular threshold are not fed back to the equalizer for further iterations, since they can be considered sufficiently reliable for subjecting them to hard decision. The LLRs passed to the decoder from the equalizer are calculated based on the linear relationship of (33) instead of the more computationally demanding (14), in order to reduce the computational complexity. We refer to this RBF-based TEQ as the reduced-complexity RBF TEQ.

The above mentioned log-likelihood gradient of (33) was found to be $g = 1.3$, according to the near-linear relationship of Fig. 6 for BPSK. For 16QAM, the LLR gradient for the LSB g_2 and g_4, corresponding to $c_{k,2}$ and $c_{k,4}$, respectively, is approximately 1.2. For the MSB, the LLR gradient $g_1^{c_{k,2}=+1}$ and $g_3^{c_{k,4}=+1}$ correspond to $c_{k,1}$ and $c_{k,3}$, respectively, when the LSB is $+1$, is equivalent to 1.2, whereas, $g_1^{c_{k,2}=-1} = g_3^{c_{k,4}=-1} = 3.2$ when the LSB is -1, according to Fig. 7. We set the LLR magnitude threshold $|\mathcal{L}|_{\text{threshold}}$ such that the symbols in the burst that were not fed back to the equalizer for further iterations became sufficiently reliable, and hence, exhibited a low probability of decoding error.

The probability of error for the detected bit $\tilde{c}_{k,l}$ can be estimated on the basis of the soft output of the turbo decoder. Referring to (9) and assuming $P(c_{k,l} = +1|\mathbf{v}_k) + P(c_{k,l} = -1|\mathbf{v}_k) = 1$, the probability of error for the detected bit is given by (34) at the bottom of the page. With the aid of the definition in (9), the probability of the bit having the value of $+1$ or -1 can be rewritten in terms of the *a posteriori* LLR of the bit, $\mathcal{L}(,|\mathbf{v})$ as follows:

$$P(c_{k,l} = +1|\mathbf{v}_k) = \frac{1}{1 + e^{-\mathcal{L}(c_{k,l}|\mathbf{v}_k)}}$$
$$P(c_{k,l} = -1|\mathbf{v}_k) = \frac{1}{1 + e^{\mathcal{L}(c_{k,l}|\mathbf{v}_k)}}. \tag{35}$$

Upon substituting (35) into (34), we redefined the probability of error of a detected bit in terms of its LLR as

$$P_{\text{error}}(c_{k,l}) = \frac{1}{1 + e^{|\mathcal{L}(c_{k,l}|\mathbf{v}_k)|}} \tag{36}$$

where $|\mathcal{L}(c_{k,l}|\mathbf{v}_k)|$ is the magnitude of $\mathcal{L}(c_{k,l}|\mathbf{v}_k)$. The threshold was set to 10 for BPSK such that the symbols that were not fed back to the equalizer exhibited a probability of error below 10^{-4}, according to (36). For 16QAM, in order to compensate for the BER degradation due to error propagation induced by incorrectly detected symbol feedback, the threshold was set to 20, which corresponds to a probability of error below 10^{-8}. Fig. 8 and Fig. 9 depict the performance of the reduced-complexity Jacobian RBF DFE TEQ relying on detected, rather than perfect, decision feedback over the three-tap equal gain, symbol-spaced Rayleigh faded CIR obeying a Doppler frequency of 3.3×10^{-5}. Referring to Fig. 8, the reduced-complexity BPSK Jacobian RBF DFE TEQ using decision feedback achieved a computational complexity reduction of approximately $((0 + 40 + 72)/3)\% = 37\%$ (0%, 40%, and 72% bits were not equalized at first, second, and third iterations, respectively) at an E_b/N_0 of 4 dB, compared to the full-complexity RBF TEQ. For 16QAM, the computational complexity reduction of the reduced-complexity Jacobian RBF DFE TEQ is approximately 54% at 10 dB according to Fig. 9.

$$P_{\text{error}}(c_{k,l}) = \begin{cases} 1 - P(c_{k,l} = +1|\mathbf{v}_k) = P(c_{k,l} = -1|\mathbf{v}_k), & \text{if } \mathcal{L}(c_{k,l}|\mathbf{v}_k) \geq 0 \\ 1 - p(c_{k,l} = -1|\mathbf{v}_k) = P(c_{k,l} = +1|\mathbf{v}_k), & \text{if } \mathcal{L}(c_{k,l}|\mathbf{v}_k) < 0. \end{cases} \tag{34}$$

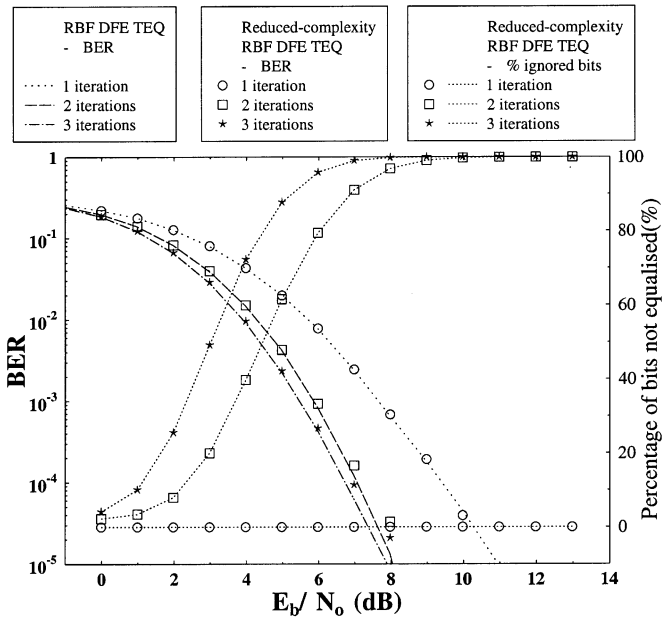


Fig. 8. BER performance and the percentage of symbols not requiring equalization by the reduced-complexity RBF TEQ using detected decision feedback over the three-tap equal-gain Rayleigh fading channel for BPSK. The LLR magnitude threshold and the LLR gradient were set to $|\mathcal{L}|_{\text{threshold}} = 10$, $g = 1.3$, respectively. The system incorporates iterative LMS-based CIR estimation.

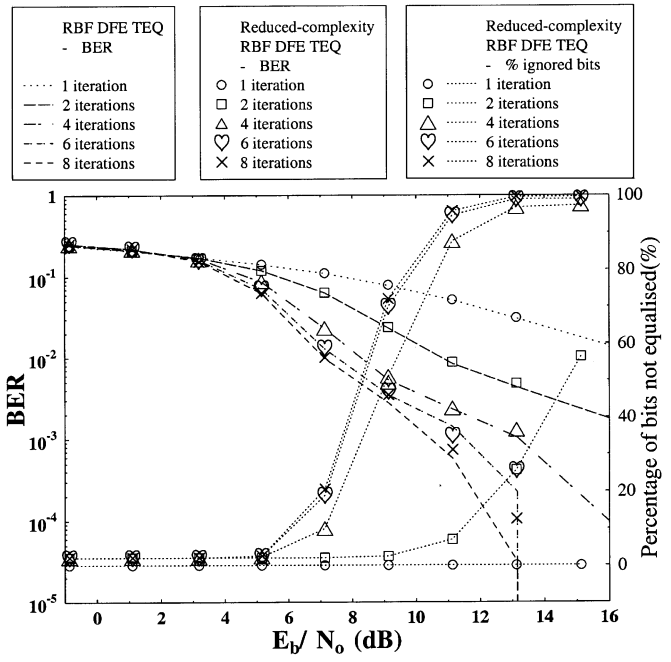


Fig. 9. BER performance and the percentage of symbols not requiring equalization by the reduced-complexity RBF TEQ using detected decision feedback over the three-tap equal-gain Rayleigh fading channel for 16QAM. LLR magnitude threshold and LLR gradients were set to $|\mathcal{L}|_{\text{threshold}} = 20$, $g_1^{c_{k,2}=+1} = g_3^{c_{k,4}=+1} = 1.2$, $g_1^{c_{k,2}=-1} = g_3^{c_{k,4}=-1} = 3.2$, and $g_2 = g_4 = 1.2$, respectively. The system incorporates iterative LMS-based CIR estimation.

The reduced-complexity RBF DFE TEQ implementation can be used instead of the RBF DFE TEQ in order to provide substantial computational reductions without degrading the BER performance. Since the reliability of the symbols in the decoded burst is provided by the channel decoder in the previous iteration, we were capable of designing a system where the percentage of bits not equalized in the decoded burst was set according to our design criteria for every iteration, such that each burst exhibited a predetermined fixed computational complexity reduction for the sake of practical, constant-complexity implementations.

VII. Conclusion

In this paper, the Jacobian RBF DFE TEQ has been proposed and analyzed comparatively in conjunction with the well-known Log-MAP TEQ [3], [17]. The associated performances and complexities have been compared in the context of BPSK, 4QAM, and 16QAM. The computational complexity of the Jacobian RBF DFE TEQ is dependent on the number of RBF centers, the CIR length, and modulation mode. The associated per-iteration-based implementation complexity of the Jacobian RBF DFE TEQ ($m = 3$, $n = 2$, $\tau = 2$) compared to that of the Log-MAP TEQ was approximately a factor 2.5, 4.4, and 16.3 lower in the context of BPSK, 4QAM, and 16QAM, respectively, when communicating over the three-path, equal-weight, symbol-spaced burst-invariant Rayleigh fading channel environment considered. This computational complexity advantage is only attained, however, in terms of the additions/subtractions, whereas the number of multiplications/divisions is the same as for Log-MAP equalizers. The performance degradation compared to the conventional Log-MAP TEQ [3] was negligible for BPSK and 4QAM, but was approximately 2.5 dB for 16QAM at BER of 10^{-3}. The significant performance degradation of the 16QAM scheme is due to the error propagation effect of the DFE, which becomes more grave in conjunction with higher order constellations. Our proposed reduced-complexity Jacobian RBF DFE TEQ was shown to provide an equivalent BER performance to that of the RBF DFE TEQ at a reduced computational load for both BPSK and 16QAM schemes. The reduced-complexity Jacobian RBF DFE TEQ using detected decision feedback provided approximately 37% (at E_b/N_0 of 4 dB) and 54% (at E_b/N_0 of 10 dB) computational reduction for BPSK and 16QAM, respectively, over dispersive Rayleigh channels.

References

[1] C. Douillard, M. Jézéquel, and C. Berrou, "Iterative correction of intersymbol interference: turbo equalization," *Eur. Trans. Telecommun.*, vol. 6, pp. 507–511, Sept./Oct. 1995.

[2] L. Hanzo, T. H. Liew, and B. L. Yeap, *Turbo Coding, Turbo Equalization and Space-Time Coding*. New York: Wiley/IEEE, 2002.

[3] G. Bauch, H. Khorram, and J. Hagenauer, "Iterative equalization and decoding in mobile communications systems," in *Proc. European Personal Mobile Communications Conf.*, Bonn, Germany, Sept. 30–Oct. 2, 1997, pp. 301–312.

[4] 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.

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

[6] L. Hanzo, C. H. Wong, and M. S. Yee, *Adaptive Wireless Transceiver*. New York: Wiley/IEEE, 2002.

[7] S. Chen, B. Mulgrew, and P. M. Grant, "A clustering technique for digital communications channel equalization using radial basis function networks," *IEEE Trans. Neural Networks*, vol. 4, pp. 570–579, July 1993.

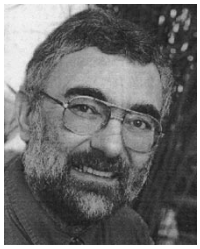
[8] M. S. Yee, B. L. Yeap, and L. Hanzo, "Radial basis function-assisted turbo equalization," in *Proc. IEEE Vehicular Technology Conf.*, Tokyo, Japan, May 15–18, 2000, pp. 640–644.
[9] S. Haykin, *Neural Networks: A Comprehensive Foundation*. New York: Macmillan, 1994.
[10] S. Chen, S. McLaughlin, and B. Mulgrew, "Complex-valued radial basis function network—part II: Application to digital communications channel equalization," *EURASIP Signal Processing*, vol. 36, pp. 175–188, Mar. 1994.
[11] M. S. Yee and L. Hanzo, "Multi-level radial basis function network-based equalisers for Rayleigh channels," in *Proc. IEEE Vehicular Technology Conf.*, Houston, TX, May 16–19, 1999, pp. 707–711.
[12] S. Chen, B. Mulgrew, and S. McLaughlin, "Adaptive Bayesian equalizer with decision feedback," *IEEE Trans. Signal Processing*, vol. 41, pp. 2918–2927, Sept. 1993.
[13] S. Chen, S. McLaughlin, B. Mulgrew, and P. M. Grant, "Adaptive Bayesian decision feedback equalizer for dispersive mobile radio channels," *IEEE Trans. Commun.*, vol. 43, pp. 1937–1945, May 1995.
[14] M. S. Yee, T. H. Liew, and L. Hanzo, "Block turbo coded burst-by-burst adaptive radial basis function decision feedback equalizer assisted modems," in *Proc. IEEE Vehicular Technology Conf.*, vol. 3, Amsterdam, The Netherlands, Sept. 1999, pp. 1600–1604.
[15] A. Knickenberg, B. L. Yeap, J. Hamorsky, M. Breiling, and L. Hanzo, "Non-iterative joint channel equalization and channel decoding," in *Proc. GLOBECOM'99*, Rio de Janeiro, Brazil, Dec. 1999, pp. 442–446.
[16] K. Abend, T. J. Harley, Jr., B. D. Fritchman, and C. Gumacos, "On optimum receivers for channels having memory," *IEEE Trans. Inform. Theory*, vol. IT-14, pp. 818–819, Nov. 1968.
[17] P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and suboptimal MAP decoding algorithms operation in the log domain," in *Proc. IEEE Int. Conf. Communications*, vol. 2, Seattle, WA, June 1995, pp. 1009–1013.
[18] P. Robertson, P. Hoeher, and E. Villebrun, "Optimal and suboptimal maximum *a posteriori* algorithms suitable for turbo decoding," *Eur. Trans. Telecommun.*, vol. 8, pp. 119–125, Mar./Apr. 1997.
[19] A. Klein, R. Pirhonen, J. Sköld, and R. Suoranta, "FRAMES multiple access mode 1—wideband TDMA with and without spreading," in *Proc. PIMRC'97*, Sept. 1997, pp. 37–41.
[20] C. H. Wong, B. L. Yeap, and L. Hanzo, "Wideband burst-by-burst adaptive modulation with turbo equalization and iterative channel estimation," in *Proc. IEEE VTC 2000*, Tokyo, Japan, 2000, pp. 2044–2048.
[21] B. L. Yeap, C. H. Wong, and L. Hanzo, "Reduced complexity in-phase/quadrature-phase turbo equalization using iterative channel estimation," in *Proc. IEEE Int. Conf. Communications*, Helsinki, Finland, June 2001, pp. 393–397.
[22] M. S. Yee, B. L. Yeap, and L. Hanzo, "Reduced complexity in-phase/quadrature-phase turbo equalization using radial basis functions," in *Proc. IEEE VTC 2001*, Rhodes, Greece, May 6–9, 2001, pp. 1395–1399.

Mong-Suan Yee (S'97–A'00) received the B.Eng. degree in electronics engineering and the Ph.D. degree from the University of Southampton, Southampton, U.K.

She is continuing her research as a Postdoctoral Research Fellow with the University of Southampton, Southampton, U.K. Recently, she joined the Toshiba Research Laboratories, Bristol, U.K. Her current research interests are in the field of neural network-based algorithms and their application to various wireless communications problems. She has coauthored numerous papers and a monograph in this field.

Bee Leong Yeap (S'97–M'00) graduated from the University of Southampton, Southampton, U.K., with a first-class honors degree in 1996 in electronics engineering, and in 2000, he received the Ph.D. degree from the same university.

He is continuing his research as a Postdoctoral Research Fellow with the University of Southampton, Southampton, U.K. His research interests include turbo coding, turbo equalization, adaptive modulation, and space–time coding. He coauthored numerous papers and a research monograph.



Lajos Hanzo (M'91–SM'92) received the master degree in electronics in 1976 and the doctorate degree in 1983, both from the Technical University of Budapest, Hungary.

During his 26-year career in telecommunications, he has held various research and academic posts in Hungary, Germany, and the U.K. Since 1986, he has been with the Department of Electronics and Computer Science, University of Southampton, Southampton, U.K., where he holds the Chair in Telecommunications. He has coauthored ten IEEE Press/Wiley books on mobile radio communications, published in excess of 400 research papers, organized and chaired conference sessions, presented overview lectures, and has been awarded a number of distinctions. Currently, he is managing an academic research team, working on a range of research projects in the field of wireless multimedia communications sponsored by industry, the Engineering and Physical Sciences Research Council (EPSRC) U.K., the European IST Programme, and the Mobile Virtual Center of Excellence (VCE), U.K. He is an enthusiastic supporter of industrial and academic liaison and he offers a range of industrial courses.

Dr. Hanzo is an IEEE Distinguished Lecturer.